LAB: Final Project - SMART HOME and Auto Parking Car

Date: 2022-12-20

Author: 21800447 Jeahyun Oh

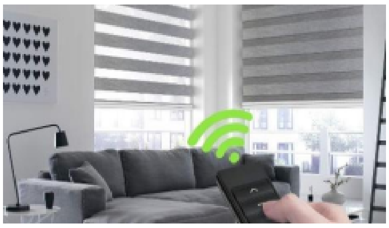
Github: https://github.com/Ohjeahyun1/EC-jeahyun-447.git

Demo Video: https://youtu.be/uKYzLiGAYlo

SMART HOME

Introduction

In this project, design an embedded system to realize a simple smart home security system with the following design criteria.



Requirement

Hardware

- MCU
  - NUCLEO-F411RE
- Analog Sensor
  - Light intensity sensor(MSE004LSM) x1
  - Ultrasonic distance sensor(HC-SR04) x1
  - Temperature sensor(LM35DZ) x1
- Digital Sensor
  - PIR motion sensor (HD-SEN0018) x1

    - Sound sensor(SZH-EK033) x1
- Actuator
    - LED x2
    - RC Servo Motor (SG90) x3
    - DC Motor x1
    - Digital Buzzer (ELB030300) x2
    - I2C LCD (SZH-EK101) x1
- Sensor
    - Button (B1)
- Communication
    - Bluetooth Module (HC-06) x1
    - PLX-DAQ
- Other
    - breadboard x2
    - Array resistor (330 Ohm)

Software

- Keil uVision, CMSIS, EC_HAL library

Problem 1: Create Flow Chat

Create Flow Chat

The important parts of the functions of the smart home are the security mode and automatic management functions.

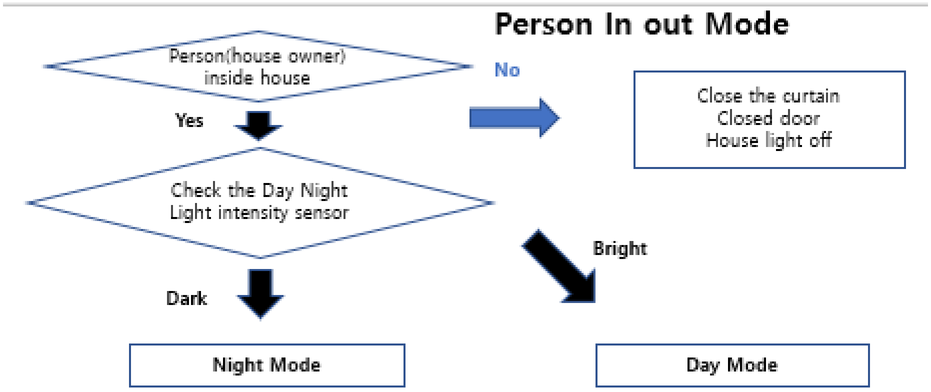
The system was operated by dividing when the owner of the house was present and not present. In addition, the day and night were automatically divided so that the system could operate accordingly.

It also added a function that automatically updates the user through the computer so that the user can know the situation of the house.

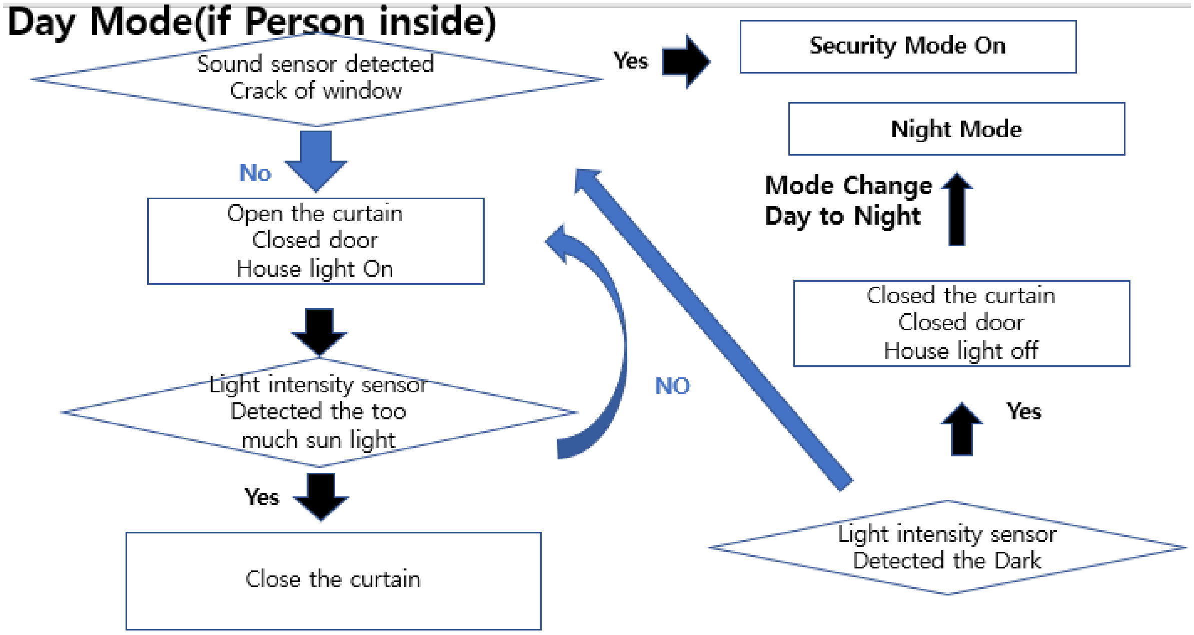
The figure below is a flowchart of the function of the smart home.

Flow Chart

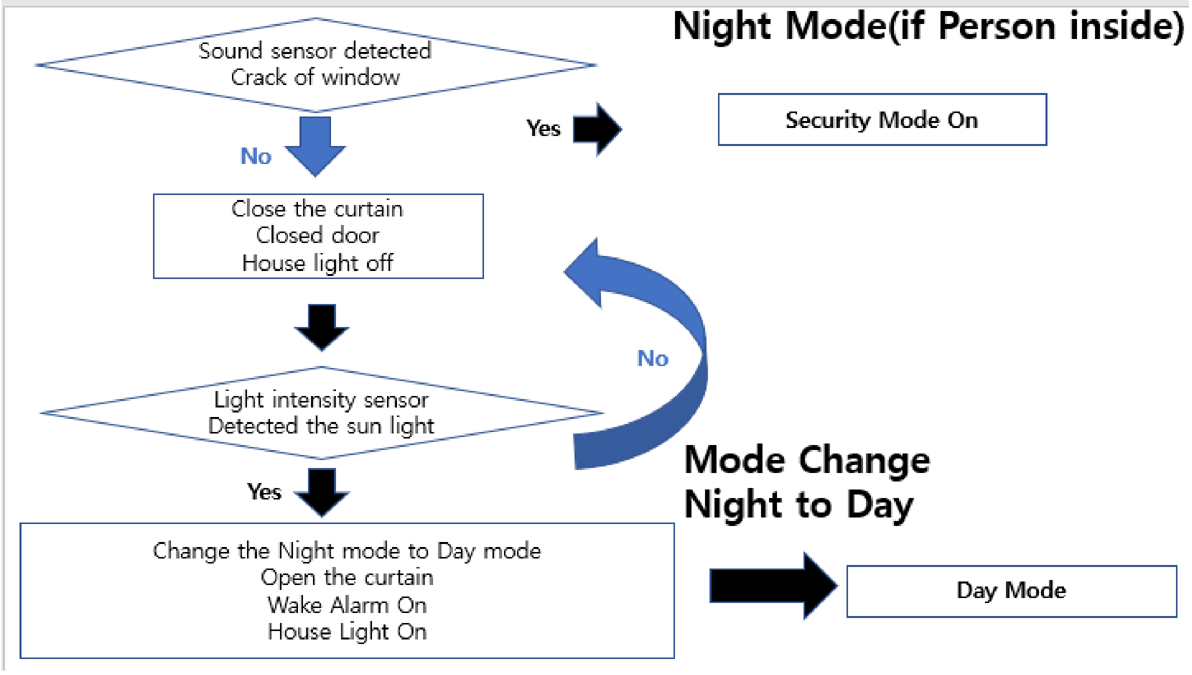
Owner In Out check



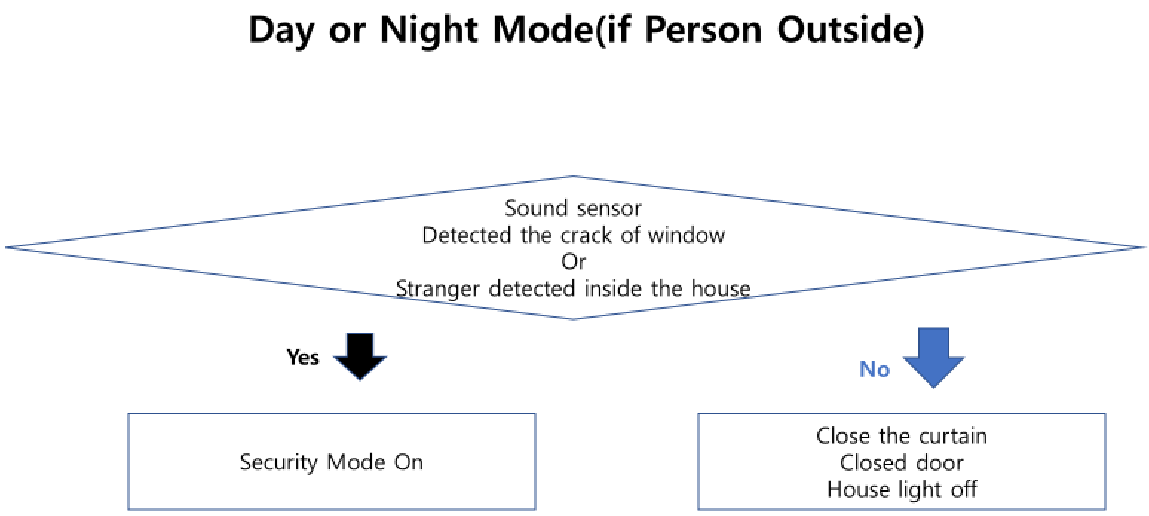
Day Mode (Owner inside)


Day Mode(if Person inside)
Sound sensor detected Crack of window
Yes
Security Mode On
No
Open the curtain Closed door House light On
Light intensity sensor Detected the too much sun light
Yes
Close the curtain
NO
Night Mode
Mode Change Day to Night
Closed the curtain Closed door House light off
Yes
Light intensity sensor Detected the Dark

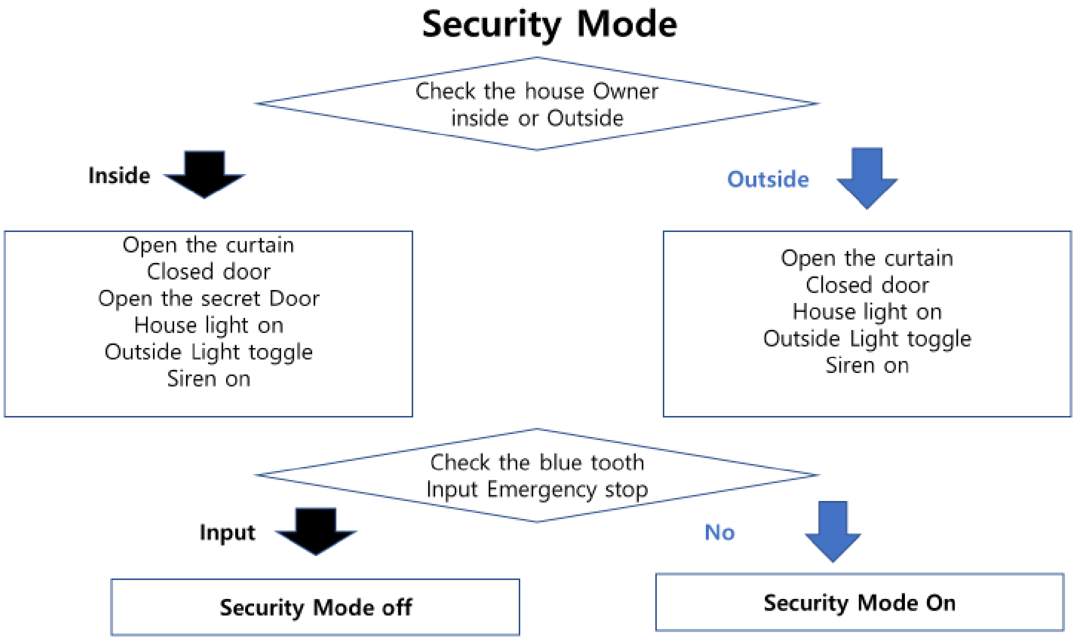
Night Mode (Owner inside)


Night Mode(if Person inside)
Sound sensor detected Crack of window
Yes
Security Mode On
No
Close the curtain Closed door House light off
No
Light intensity sensor Detected the sun light
Yes
Change the Night mode to Day mode Open the curtain Wake Alarm On House Light On
Mode Change Night to Day
Day Mode

Owner Outside

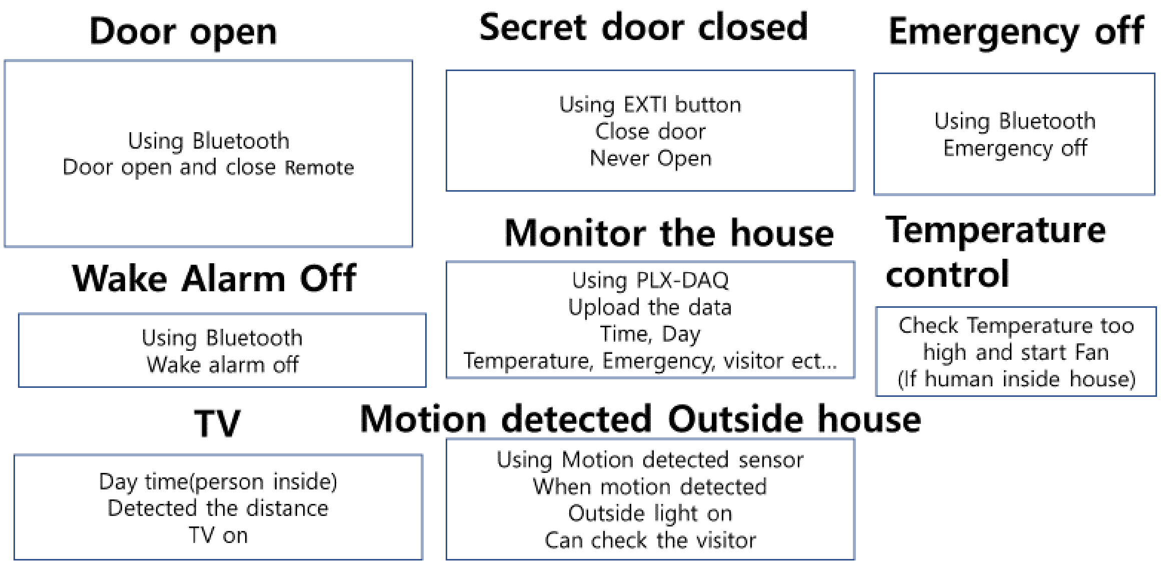

Day or Night Mode(if Person Outside)
Sound sensor
Detected the crack of window
Or
Stranger detected inside the house
Yes
No
Security Mode On
Close the curtain
Closed door
House light off

Security Mode



Additional Functions

Additional Functions



Door open

Using Bluetooth
Door open and close Remote

Secret door closed

Using EXTI button
Close door
Never Open

Emergency off

Using Bluetooth
Emergency off

Monitor the house

Using PLX-DAQ
Upload the data
Time, Day
Temperature, Emergency, visitor ect...

Temperature control

Check Temperature too
high and start Fan
(If human inside house)

Wake Alarm Off

Using Bluetooth
Wake alarm off

TV

Day time(person inside)
Detected the distance
TV on

Motion detected Outside house

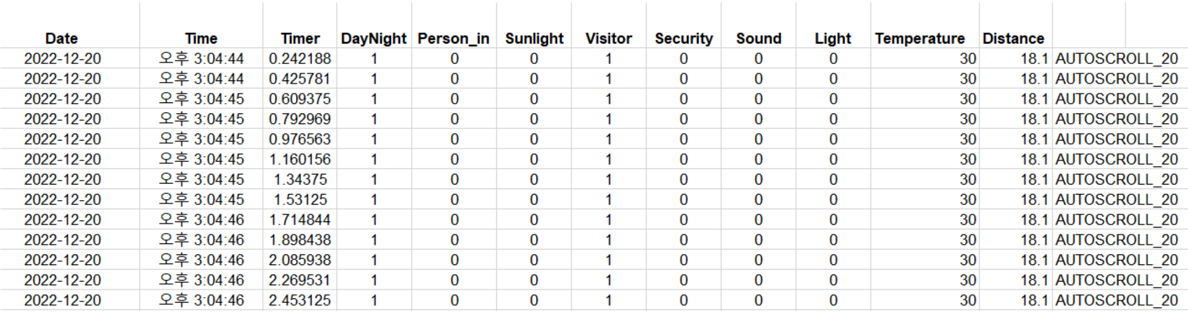
Using Motion detected sensor
When motion detected
Outside light on
Can check the visitor

Problem 2: Make Code

Make a smart home program with flow chart.

- Check the Day Night with Light intensity sensor
- Detected the crack of window with sound sensor
- Door and curtain control using RC servo Motor
- Monitor the house using PLX-DAQ
- Remote controller using Bluetooth
- Temperature detection through temperature sensor and Motor operation
- Wake up and security alarms
- Visitor detection and light-on at the entrance through motion sensor
- etc....

PLX-DAQ

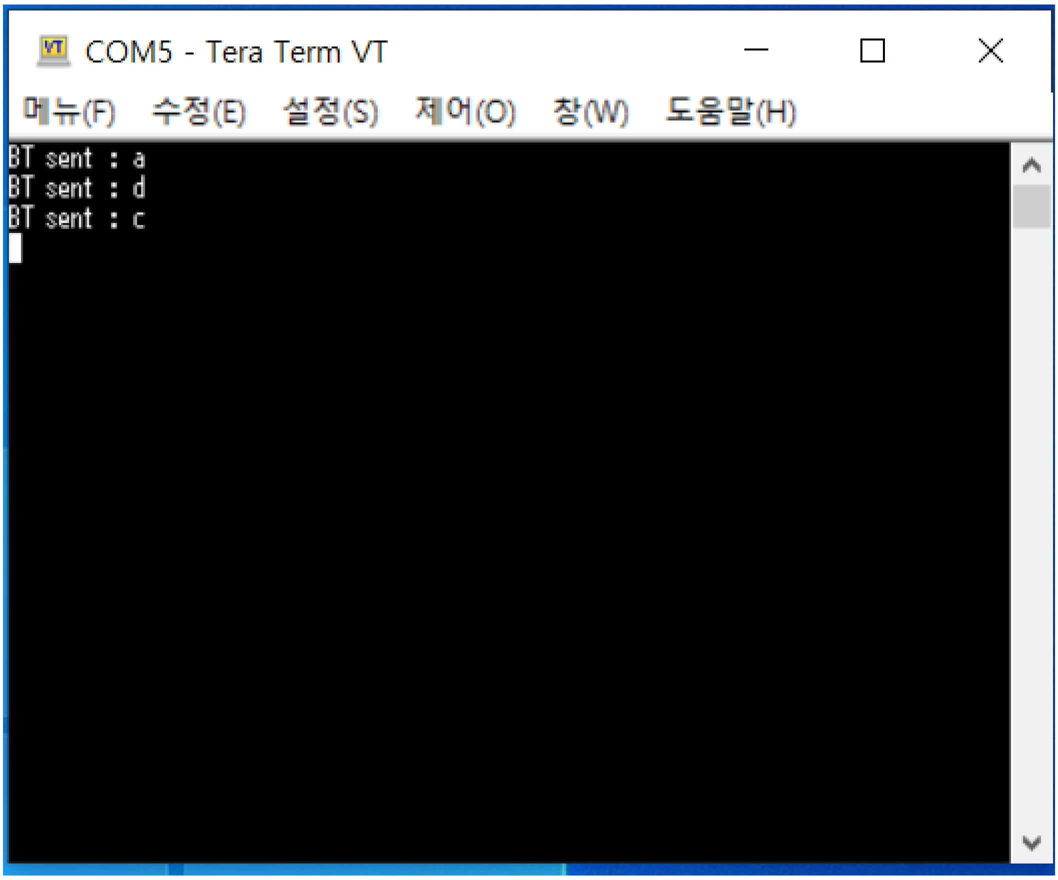


Date	Time	Timer	DayNight	Person_in	Sunlight	Visitor	Security	Sound	Light	Temperature	Distance	
2022-12-20	오후 3:04:44	0.242188	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:44	0.425781	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	0.609375	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	0.792969	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	0.976563	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	1.160156	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	1.34375	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:45	1.53125	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:46	1.714844	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:46	1.898438	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:46	2.085938	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:46	2.269531	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20
2022-12-20	오후 3:04:46	2.453125	1	0	0	1	0	0	0	30	18.1	AUTOSCROLL_20

Transferring data from MCU to PC so that owner could know information in the house from anywhere and anytime.

Automatically update date,time, day and night check, the owner inside or outside, sunlight, visitor, security mode, window sound, light, temperature, Distance check to identify intruders.

Blue tooth



```
BT sent : a
BT sent : d
BT sent : c
```

Remote control of open and close the door, turn of alarm, turn off security mode, etc. from the PC to the MCU using Blue Tooth

Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_Final_Smarthome`

- The project name is "**LAB_Final_Smarthome".**
- Create a new source file named as "**LAB_Final_Smarthome.c"**

> You MUST write your name on the source file inside the comment section.

2. Include your updated library in `\repos\EC\lib\` to your project.

- **ecGPIO.h, ecGPIO.c**
- **ecRCC.h, ecRCC.c**
- **ecEXTI.h, ecEXTI.c**
- **ecTIM.h**, **ecTIM.c**
- **ecPWM.h ecPWM.c**
- **ecADC.h ecADC.c**
- **ecSysTick.h ecSysTick.c**
- **ecUART.h ecUART.c**

1. Use flow charts to implement code
2. Create a smart house model and test it to see if it works.
3. When you make a door and curtain, you use hinges to make them.
4. Put distance sensor, temperature sensor, LED, and motor, etc, in the house and put Light intensity sensor, motion sensor, and security alarm buzzer outside the house.

Configuration

USART

Type	Port - Pin	Configuration
System Clock		PLL 84MHz
USART2: PC – MCU PLX_DAQ		No Parity, 8-bit Data, 1-bit Stop bit, 9600 baud-rate
USART1: PC – MCU Blue tooth	TXD: PA9 RXD: PA10	No Parity, 8-bit Data, 1-bit Stop bit, 9600 baud-rate

ADC

TIM2	ADC1_CH8 (1st channel) ADC1_CH9 (2nd channel)	PB_0(Temperature Sensor) PB_1 (Light intensity sensor)
Up-Counter, Counter CLK 1kHz OC1M(Output Compare 1 Mode) : PWM mode 1 Master Mode Selection: (TRGO) OC1REF	ADC Clock Prescaler /8 12-bit resolution, right alignment Single Conversion mode Scan mode: Two channels in regular group External Trigger (Timer2 TRGO) @1kHz Trigger Detection on Rising Edge	Analog Mode No Pull-up Pull-down

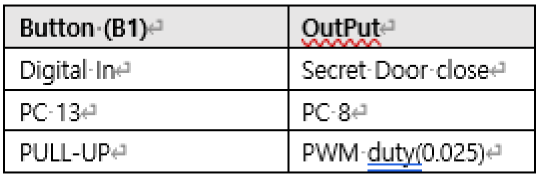
Ultrasonic Distance Sensor

System Clock	PWM	Input Capture
PLL(84MHz)	PA6(TIM3_CH1)	PB7(TIM4_CH2)
	AF, Push-Pull, No Pull-up Pull-down, Fast	AF, No Pull-up Pull-down
	PWM period: 50msec Pulse width: 10usec	Counter Clock: 0.1MHz (10us) TI2 -> IC2 (rising edge) TI2 -> IC1 (falling edge)

RC sevor Motor

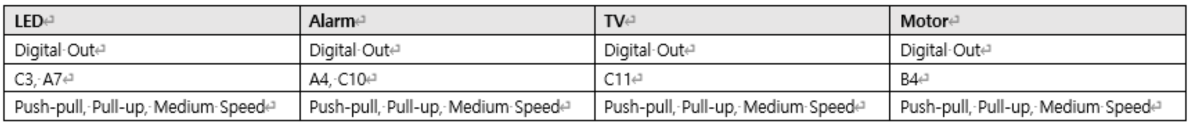
PWM(Door)	PWM(Curtain)	PWM(Secret Door)
PB5(TIM3_CH2)	PC8(TIM3_CH3)	PC8(TIM3_CH4)
AF, Push-Pull, No Pull-up Pull-down, Fast	AF, Push-Pull, No Pull-up Pull-down, Fast	AF, Push-Pull, No Pull-up Pull-down, Fast
PWM period: 20msec	PWM period: 20msec	PWM period: 20msec
duty ratio: 0.5~2.5msec	duty ratio: 0.5~2.5msec	duty ratio: 0.5~2.5msec

EXTI



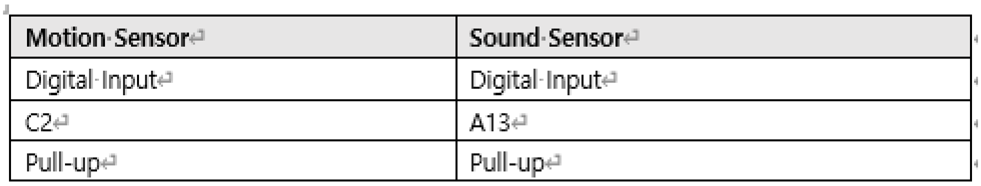
Button (B1)	OutPut
Digital In	Secret Door close
PC 13	PC 8
PULL-UP	PWM duty(0.025)

Digital Out



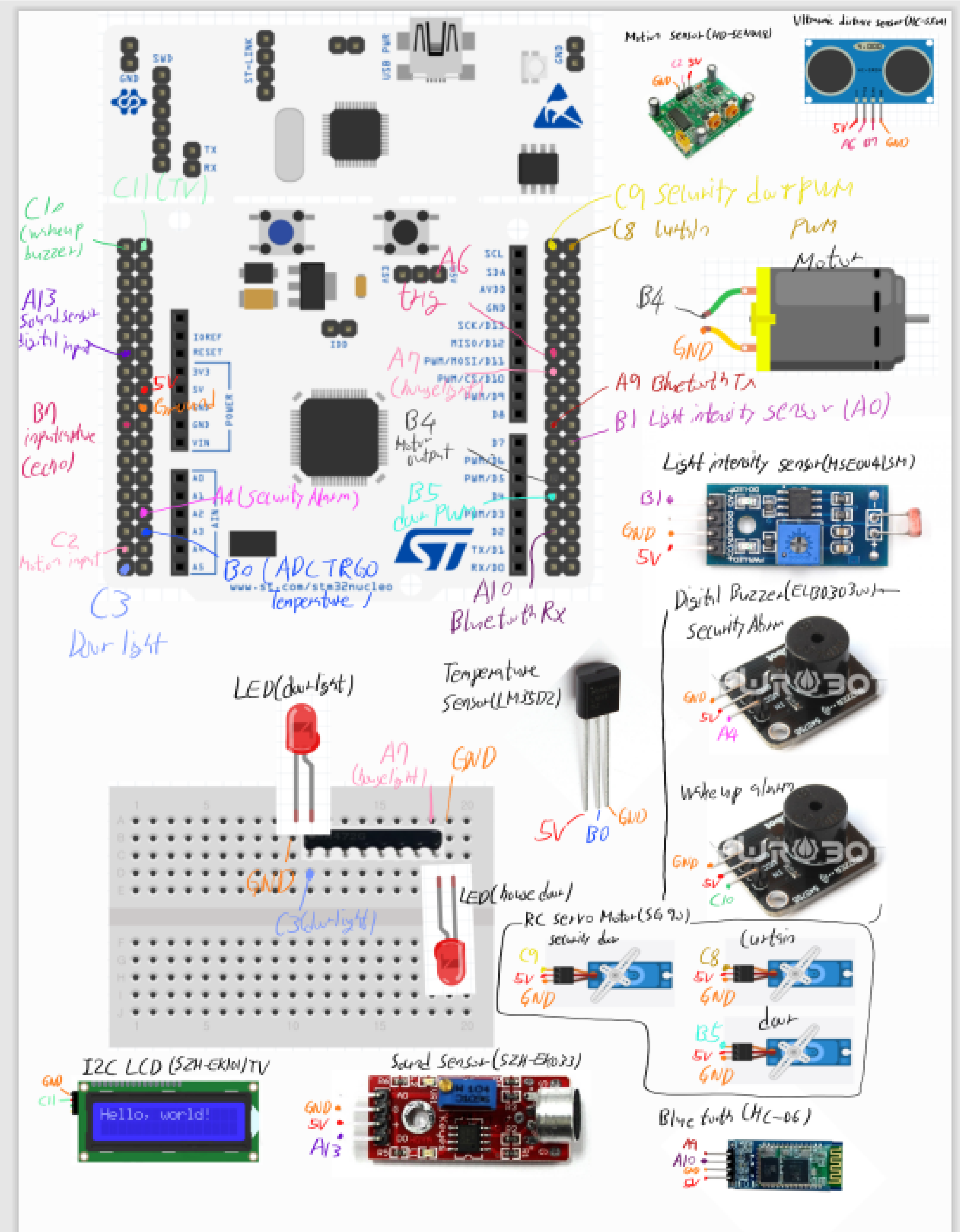
LED	Alarm	TV	Motor
Digital Out	Digital Out	Digital Out	Digital Out
C3, A7	A4, C10	C11	B4
Push-pull, Pull-up, Medium Speed	Push-pull, Pull-up, Medium Speed	Push-pull, Pull-up, Medium Speed	Push-pull, Pull-up, Medium Speed

Digital Input



Motion Sensor	Sound Sensor
Digital Input	Digital Input
C2	A13
Pull-up	Pull-up

Circuit Diagram



Code

Your code goes here: https://github.com/Ohjeahyun1/EC-jeahyun-447/blob/8c945375c7e959bab20aede7e55385b616ff5112/lab/LAB_Final_Project_main_1.c

https://github.com/Ohjeahyun1/EC-jeahyun-447/tree/main/include

Explain your source code with necessary comments.

Description with Code

- Code Initialization

```
// Initialization
void setup(void)
{
    RCC_PLL_init();                                    // 84Mhz clock
    SysTick_init();                                    // SysTick init
    USART_init(USART2, 9600);                          // USART 2 init (PLX-DAQ)
    USART_begin(USART1, GPIOA, 9, GPIOA,10, 9600);    // USART 1 Blue Tooth
PA10 - RXD , PA9 - TXD

    /*
    // Doorbell switch
    GPIO_init(GPIOC, BUTTON_PIN, INPUT);  // calls RCC_GPIOC_enable() and button
pin mode -> input
    GPIO_pupdr(GPIOC, BUTTON_PIN, EC_PU); // GPIOC button pin pupdr -> pull up
  EXTI_init(GPIOC,BUTTON_PIN,FALL,0);   //EXTI button PIN -> trigger
type(falling),propriority(0)
  */

    // ADC setting
  ADC_init(GPIOB, 0, TRGO);                            //ch8 (Temperature)
    ADC_init(GPIOB, 1, TRGO);                          //ch9 (Light intensity
sensor)

    // ADC channel sequence setting
    ADC_sequence(2, seqCHn);

    // ADON, SW Trigger enable
    ADC_start();



    //EXTI init
    GPIO_init(GPIOC, BUTTON_PIN, INPUT);               // calls
RCC_GPIOC_enable() and button pin mode -> input
    GPIO_pupdr(GPIOC, BUTTON_PIN, EC_PU);              // GPIOC button pin pupdr
-> pull up
  EXTI_init(GPIOC,BUTTON_PIN,FALL,0);
                                              // EXTI button PIN -> trigger
type(falling),propriority(0)

    //Motor init
    GPIO_all_init(GPIOB, 4, OUTPUT,EC_PU,PP,SMED);    // motor
    //Alarm init
    GPIO_all_init(GPIOA,4,OUTPUT,EC_PU,PP,SMED);      // security alarm

    GPIO_all_init(GPIOC,10,OUTPUT,EC_PU,PP,SMED);     // wake up alarm
    // TV init
    GPIO_all_init(GPIOC,11,OUTPUT,EC_PU,PP,SMED);     // TV
    //LEDs init
    GPIO_all_init(GPIOA,7,OUTPUT,EC_PU,PP,SMED);      // house light
    GPIO_all_init(GPIOC,3,OUTPUT,EC_PU,PP,SMED);      // door light
  // Motion detected input
    GPIO_init(GPIOC, 2, INPUT);                        // calls
RCC_GPIOC_enable() and button pin mode -> input
    GPIO_pupdr(GPIOC, 2, EC_PU);                       // GPIOC button pin pupdr
-> pull up
    // Sound detected digital input
    GPIO_init(GPIOA, 13, INPUT);                       // calls
RCC_GPIOC_enable() and button pin mode -> input
    GPIO_pupdr(GPIOA, 13, EC_PU);                      // GPIOC button pin pupdr
-> pull up
```

```
    // PWM configuration ---------------------------------------------
    PWM_t trig;                                        // PWM(TIM3_CH1) for trig
    PWM_init(&trig,GPIOA,6,UP,SFAST,PP,EC_NOPUPD,1);   // PA_6: Ultrasonic trig
pulse
    PWM_period_us(&trig, 50000);                       // PWM of 50ms period.
Use period_us()
    PWM_pulsewidth_us(&trig, 10);                      // PWM pulsewidth 10us


    // Input Capture configuration(house) ---------------------------
    IC_t echo_h;                                       // Input Capture for echo
    ICAP_init(&echo_h,GPIOB,7,EC_NOPUPD);              // P7 as input caputre
    ICAP_counter_us(&echo_h, 10);                      // ICAP counter step time
as 10us
    ICAP_setup(&echo_h, 2, IC_RISE);                   // TIM4_CH2 as IC2 ,
rising edge detec
    ICAP_setup(&echo_h,1,IC_FALL);                     // TIM4_CH1 as IC1 ,
falling edge detec


  // PWM RC servo Motor
    PWM_init(&Door,GPIOB,5,UP,SFAST,PP,EC_PU,1);       // TIM3_CH2(PB5) UP
clock,FAST,1ms clock
  PWM_period_ms(&Door,20);                             // set PWM period 20ms
  PWM_init(&Curtain,GPIOC,8,UP,SFAST,PP,EC_PU,1);      // TIM3_CH3(PC8) UP
clock,FAST,1ms clock
  PWM_period_ms(&Curtain,20);                          // set PWM period 20ms
    PWM_init(&Door_s,GPIOC,9,UP,SFAST,PP,EC_PU,1);     // TIM3_CH4(PC9) UP
clock,FAST,1ms clock
  PWM_period_ms(&Door_s,20);                           // set PWM period 20ms

}
```

- EXTI

When security mode is activated, if the owner is inside, the secret door opens, so that the secret door can be closed after escaping the house.

```
// Secrte Door close when owner escape
void EXTI15_10_IRQHandler(void) {
    if (is_pending_EXTI(BUTTON_PIN)){                  //when button pressed
     PWM_duty(&Door_s,0.025);                          // if owner outside close
the secret door
        in = 0;                                          // owner outside
     clear_pending_EXTI(BUTTON_PIN);                   // cleared by writing '1'
    }
}
```

- Time interrupt

Measure the distance of the house to determine if thieves have entered, etc

```
// Detect the distance using Ultrasonic distance sensor
void TIM4_IRQHandler(void){
    if(is_UIF(TIM4)){                                                  //
Update interrupt
```

```
        ovf_cnt++;
               // overflow count
        clear_UIF(TIM4);
// clear update interrupt flag
    }
  if(is_CCIF(TIM4, 2)){
   // TIM4_Ch2 (IC2) Capture Flag. Rising Edge Detect
        time1_h = TIM4->CCR2;
         // Capture TimeStart
        clear_CCIF(TIM4, 2);
    }else if(is_CCIF(TIM4, 1)){
     // TIM4_Ch1 (IC1) Capture Flag. Falling Edge Detect
        time2_h = TIM4->CCR1;
           // Capture TimeEnd
        if((time2_h-time1_h)<(TIM4->ARR+1)&(ovf_cnt==1)) ovf_cnt=0;     // if
(time2-time1)< ARR+1 make over count 0
        timeInterval_h = ((time2_h-time1_h)+(TIM4->ARR+1)*ovf_cnt)/100; // Total
time of echo pulse (10us * counter pulses -> [msec] unit)
        ovf_cnt = 0;                                                      //
overflow reset
        clear_CCIF(TIM4, 1);
       // clear capture/compare interrupt flag
    }
}
```

- ADC

Converting Analog values of Temperature and Light intensity sensor to Digital values

```
// ADC Temperature and Light detect
void ADC_IRQHandler(void){
    if((is_ADC_OVR())){
        clear_ADC_OVR();
    }

    if(is_ADC_EOC()){                //after finishing sequence
            if (flag==0){                // Temperature sensor
                Tem = ADC_read()/10.0;
                if(Tem > 60) flag ^= 1;    // for the flag error
            }
            else if (flag==1){           // Light intensity sensor
                Light_detect = ADC_read();
            }
        flag =! flag;
    }
}
```

- USART 1

Send commands using Blue tooth from PC to MCU

```
// USART1 BlueTooth PC -> MCU
void USART1_IRQHandler(){       //USART1 INT
    if(is_USART_RXNE(USART1)){
        mcu2Data = USART_getc(USART1);                       // PC -> MCU
        USART_write(USART1,(uint8_t*) "BT sent : ", 10);
        USART_write(USART1, &mcu2Data, 1);
        USART_write(USART1, "\r\n", 2);
            bReceive = 1;

    }
}
```

Main Code

- USART 2 setting

PLX-DAQ setting

```
int main(void) {
    // Initialiization --------------------------------------------------------
    setup();
//  printf("Hello Nucleo\r\n");

    //USART2 excel_DAQ initialize
    USART_write(USART2,(unsigned char*) "CLEARSHEET\r\n",12);
    USART_write(USART2,(unsigned char*)
"LABEL,Date,Time,Timer,DayNight,Person_in,Sunlight,Visitor,Security,Sound,Light,
Temperature,Distance\r\n",105);
```

- USART 2 output

Transmit date,time,day and night, sunlight, etc. from MCU to PC via PLX-DAQ

```
//DayNight,Person_in,Sunlight,Visitor,Security,Sound,Light,Temperature,Distance
     //USART2 Trasnmit sensor value to server
        sprintf(buf1, "%d", daynight);                   // Day Night
        sprintf(buf2, "%d", in);                         // Owner inside outside
        sprintf(buf3, "%f", Light_detect);               // Sunlight
        if (motion == 4)      sprintf(buf4, "%d", 1);      // Motion detect
outside
        else sprintf(buf4, "%d", motion);
    sprintf(buf5, "%d", security);                   // Security Mode on off
            if (sound > 0)     sprintf(buf6, "%d", 0);     // Sound sensor
        else sprintf(buf6, "%d", 1);
    sprintf(buf7, "%d", Light_h);                    // House Light
    sprintf(buf8, "%f", Tem);                          // Temperature
    sprintf(buf9, "%f", distance_h);                     // Distance of house
    //
        USART_write(USART2,(unsigned char*) "DATA,DATE,TIME,TIMER,",21);    //
transmit char to USART6
        USART_write(USART2,&buf1,4);
        USART_write(USART2,(unsigned char*) ",",1);                       //
transmit char to USART6
        USART_write(USART2,&buf2,4);
        USART_write(USART2,(unsigned char*) ",",1);                       //
transmit char to USART6
        USART_write(USART2,&buf3,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf4,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf5,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf6,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf7,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf8,4);
        USART_write(USART2,(unsigned char*) ",",1);                      //
transmit char to USART6
        USART_write(USART2,&buf9,4);
        USART_write(USART2,(unsigned char*) ",AUTOSCROLL_20\r\n",16);         //
transmit char to USART6
```

- USART 1 (Blue Tooth)

Send Commands that to open a door, etc from PC to MCU

```
// Inifinite Loop ----------------------------------------------------------
    while(1){
        if (bReceive == 1){
            if (mcu2Data == 'd' ){             //d input
            door = 1;                          //door open
          in = 1;                            // person inside
            }else if (mcu2Data == 'e' ){       //e input
            security = 1;                      //security on
            }else if (mcu2Data == 'x' ){       //x input
            security = 0;                      //security off
            }else if (mcu2Data == 'o' ){       //o input
            in = 0;                            //person outside
            }else if (mcu2Data == 'a' ){       //a input
            alarm_w = LOW;                     //alarm off
            }else if (mcu2Data == 'c' ){       //c input
            door = 0;                          //door close
            }
            bReceive = 0;
    }
```

- Motion detect

If the motion sensor detects a person, the motion flag on

```
// motion detected Light outside on
     motion = GPIO_read(GPIOC,2);

    //for increasing the hime of the motion light
     if(motion == 4) {
         motion_o = 10; //motion detected
         }
```

- Owner in out and Day night

As shown in the flow chart, the light, curtains, and TV at home are automatically adjusted differently depending on the conditions of (day and night) and (when the landlord is in the house and not in the house).

```
//owner inout check
    if(in  == 1){                                      // person inside
    if(daynight == 1){                                 // day mode
            Light_h = HIGH;                                // Light on
        // Depending on distance TV on off
        if (distance_h  < 6.0 ) TV = HIGH;               // TV On
        else TV = LOW;                                   // TV off
        // Light Too strong
        if(Light_detect < 500) {
            light_co++;
            if(light_co > 3) PWM_duty(&Curtain,0.025);     // too many sunlight
closed curtain
        }else if(Light_detect > 2000){                   // Day -> Night
            PWM_duty(&Curtain,0.025);                      // closed curtain
            Light_h = LOW;                                 // house light off
            alarm_w = LOW;                                 // wake up Alarm off
          daynight = 0;                                  // night mode
        }else {
            PWM_duty(&Curtain,0.075);                      // opened curtain
            light_co = 0;
        }
    }else{                                             // night mode
      Light_h = LOW;                                   // House Light off
        PWM_duty(&Curtain,0.025);                        // closed curtain
        PWM_duty(&Door,0.025);                           // closed the door
        if(Light_detect < 800) {                         // Night -> Day
         PWM_duty(&Curtain,0.075);                       // open the curtain
         Light_h = HIGH;                                 // house light off
       alarm_w = HIGH;                                 // Wake up Alarm on
       daynight = 1;                                   // Day mode
        }

    }
}else{                                               // No person inside
    Light_h = LOW;                                     // Light off
  PWM_duty(&Door,0.025);                             // closed the door
    PWM_duty(&Curtain,0.025);                          // closed curtain
    if (distance_h  < 6.0 && distance_h > 2.0)  person_count++;            //
check stranger in the house
  else person_count = 0;
    if (person_count > 7) security = 1;                // Because of the
error(Distance sensor)
}
```

- Door open and close

Control door according to door flag

```
 if(door == 0)      PWM_duty(&Door,0.025);         // closed the door
        else if(door ==1)       PWM_duty(&Door,0.075);       // open the door
```

- Security Mode

Security mode is activated when a person enters in the absence of a owner or when a sound sensor detects a window breaking sound.

```
 //Detect the crack of windown
  sound = GPIO_read(GPIOA,13);
 // Sound detected security mode on
    if(sound == 0) security++;

// Security Mode
  if(security > 0) {                                 // Security On
        PWM_duty(&Curtain,0.075);                        // open curtain
        PWM_duty(&Door,0.025);                           // closed the door
        alarm_s = HIGH;                                  // Security Alarm on
        Light_o ^=1;                                     // Light outside toggle
      Light_h = 1;                                     // Light inside On
        if (in == 1)        PWM_duty(&Door_s,0.075);         // if person inside
open the secret door
      else  PWM_duty(&Door_s,0.025);                   // if person outside
close the secret door
    }else{                                             // Security Off
        PWM_duty(&Door_s,0.025);                         // close the secret
door
        alarm_s = LOW;                                   // Security Alarm off
    if (motion_o > 1) Light_o = HIGH;                  // motion detected Light
outside On
    else Light_o = LOW;                                  // Light outside On
    }
```

- Temperature control

Motor operation if the temperature value received by the temperature sensor is too high when owner inside the house and security mode off.

```
// If Tempeature too high Motor start
        if(Tem > 30) {
            if(security == 0){                             // Security Mode off
            if(in == 1)         GPIO_write(GPIOB,4,HIGH);      // person inside
motor start
          else            GPIO_write(GPIOB,4,LOW);       // person outside motor
stop
            }
        }else GPIO_write(GPIOB,4,LOW);                   // Tem < 30 motor stop
```

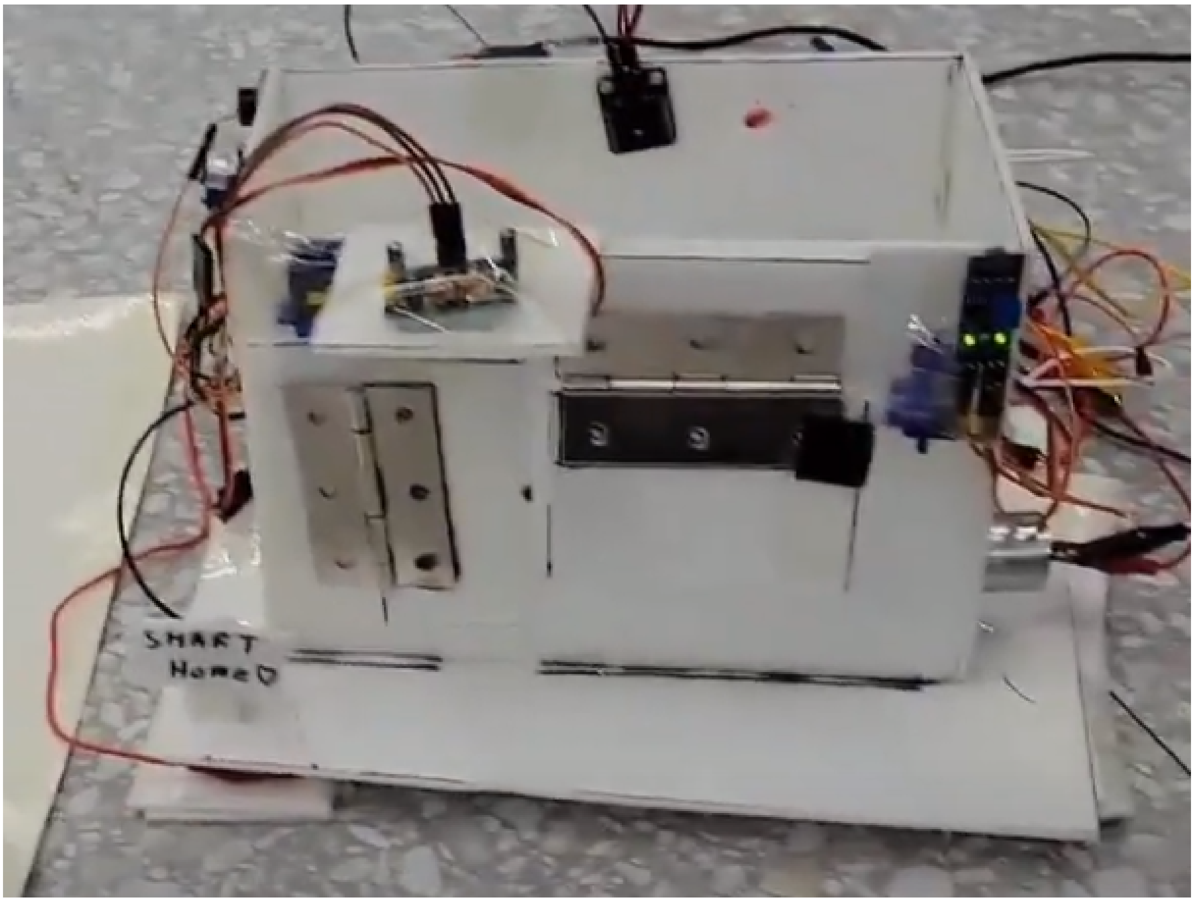
- GPIO_write

LEDS, alarms and TV output depending on the situation

```
// LEDs
    GPIO_write(GPIOA,7,Light_h);                      // Light inside house
    GPIO_write(GPIOC,3,Light_o);                      // Light outside house
    // Alarm
    GPIO_write(GPIOA,4,alarm_s);                      // security alarm
    GPIO_write(GPIOC,10,alarm_w);                     // wake up alarm
    // TV
    GPIO_write(GPIOC,11,TV);                          // TV
```

Results

Experiment images and results



By implementing the functions shown in the flow chart, it was possible to realize a house that automatically lived a pleasant and safe life under various conditions.

Add demo video link: https://youtu.be/uKYzLiGAYlo

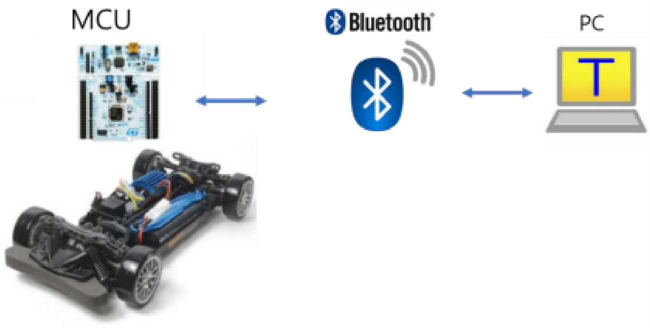
Complement

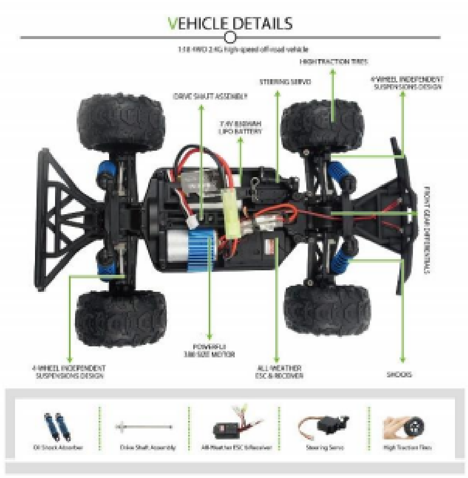
1. If the door was made to open if the correct password was entered using a keypad, it would have been similar to the actual house.
2. Adding a security camera would have made it more secure.
3. It would be a better smart home if I2C LCD was not simply used as a TV, but I2C was used to display temperatures.
4. It would be a better smart home if using buttons and EXTI to add functions like doorbells.

Auto Parking Car

Introduction

In this project, design an simple program to control an RC car steering and speed by sending the command message from PC via bluetooth. And, also have auto parking function.





Requirement

Hardware

- MCU
  - NUCLEO-F411RE
- Actuator:
  - DC motor x2
- Analog Sensor
  - Ultrasonic distance sensor(HC-SR04) x1
  - IR Reflective Sensor (TCRT 5000) x2
- Communication
  - Bluetooth Module (HC-06) x1
- Others
  - DC motor driver(L9110s)
  - breadboard
  - RC car
  - Battery

Software

- Keil uVision, CMSIS, EC_HAL library

Problem 1: Create Flow Chat

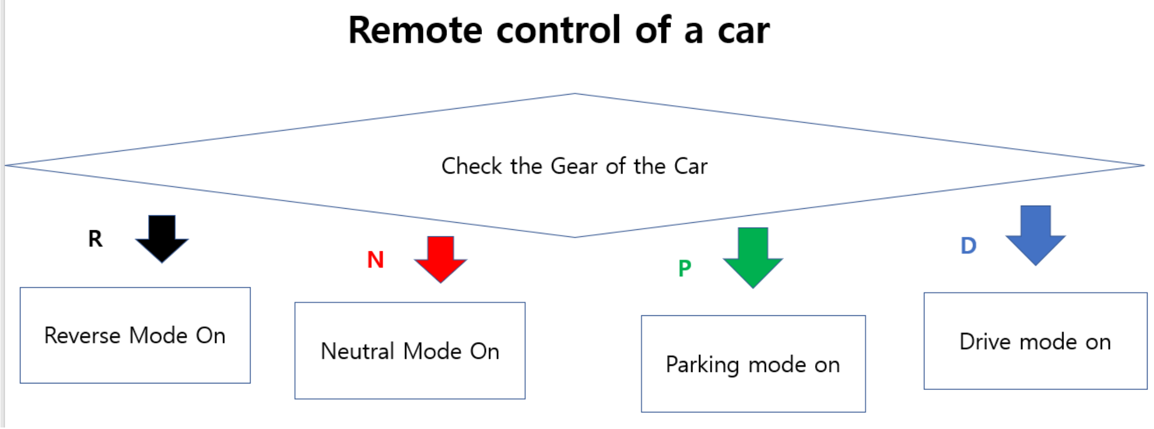
Create Flow Chat

The most important features of a car with automatic parking are remote control and automatic parking.

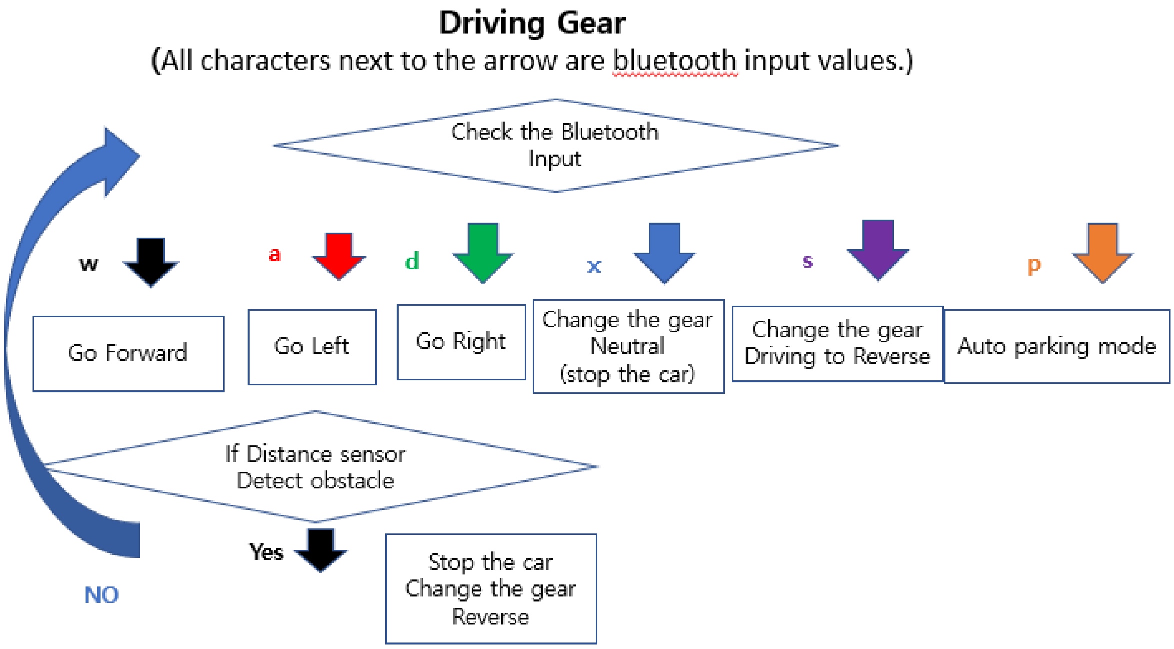
Through Bluetooth communication, the car's gear can be changed, and it can be driven in various directions such as left turn and right turn straight. It also implemented an automatic stop function when detecting obstacles.
Line tracing is used to automatically park.

The figure below is a flowchart of the function of the Auto parking Car.

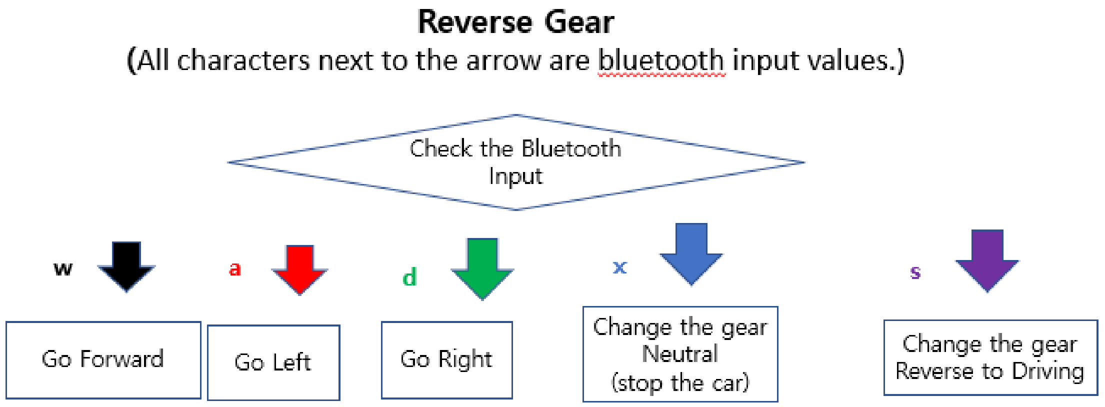
Remote control



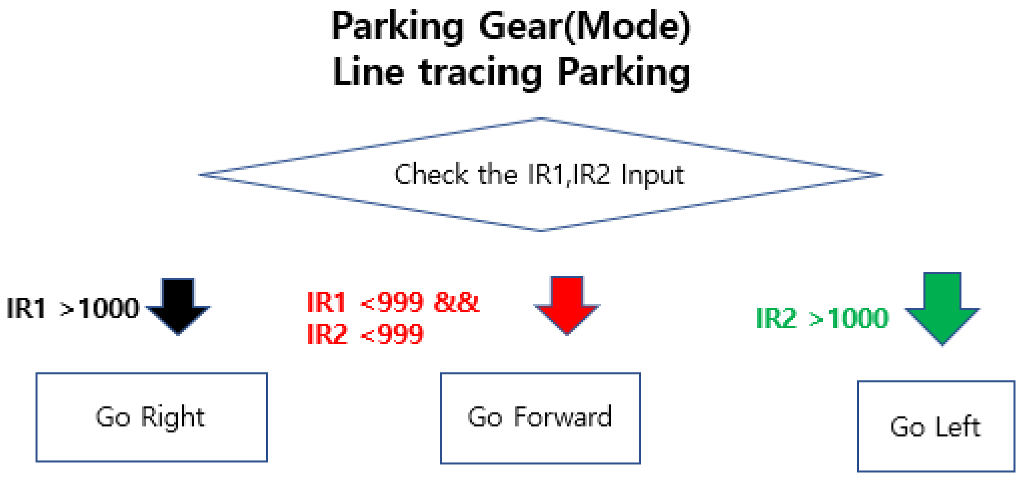
Driving Gear


Driving Gear
(All characters next to the arrow are bluetooth input values.)
Check the Bluetooth Input
w
a
d
x
s
p
Go Forward
Go Left
Go Right
Change the gear Neutral (stop the car)
Change the gear Driving to Reverse
Auto parking mode
If Distance sensor Detect obstacle
Yes
Stop the car Change the gear Reverse
NO

Reverse Gear


Reverse Gear
(All characters next to the arrow are bluetooth input values.)
Check the Bluetooth Input
w
a
d
x
s
Go Forward
Go Left
Go Right
Change the gear Neutral (stop the car)
Change the gear Reverse to Driving

Parking Mode



Problem 2: Make Code

Make a Auto parking program with flow chat.

- Send commands from PC to MCU via Bluetooth
- Change the speed and direction of the motor according to the command and gear
- Line tracing via IR sensor when park command is received
- Stop when detecting obstacles in a Diving Gear

Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_Final_Autoparkingcar`

- The project name is "**LAB_Final_Autoparkingcar".**
- Create a new source file named as "**LAB_Final_Autoparkingcar.c**"

> You MUST write your name on the source file inside the comment section.

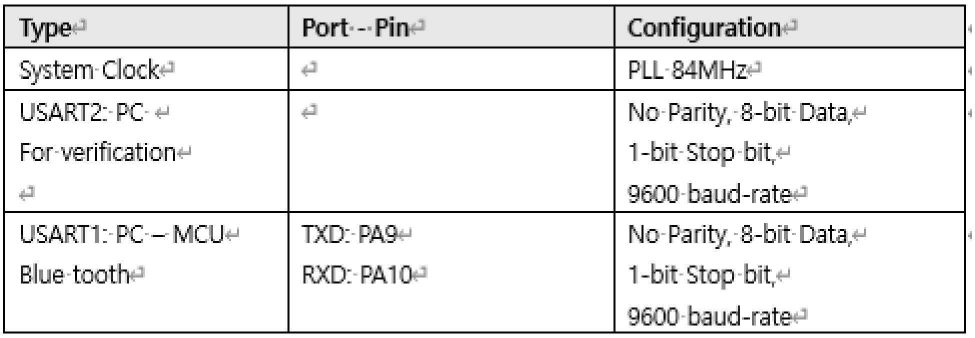
2. Include your updated library in `\repos\EC\lib\` to your project.

- **ecGPIO.h, ecGPIO.c**
- **ecRCC.h, ecRCC.c**
- **ecEXTI.h, ecEXTI.c**
- **ecTIM.h**, **ecTIM.c**
- **ecPWM.h ecPWM.c**
- **ecADC.h ecADC.c**
- **ecSysTick.h ecSysTick.c**
- **ecUART.h ecUART.c**

1. Connect the motor and the motor driver
2. Send commands from PC to MCU via Bluetooth.
3. Use flow charts to implement code.
4. Create RC cars (e.g. front of distance sensor).
5. Make a track and check if line tracing works well.

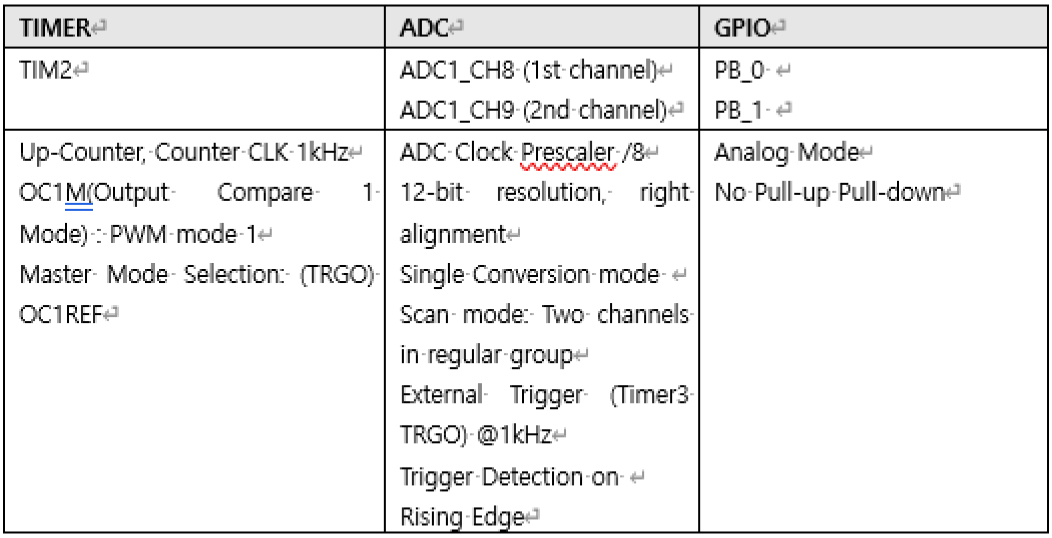
Configuration

USART



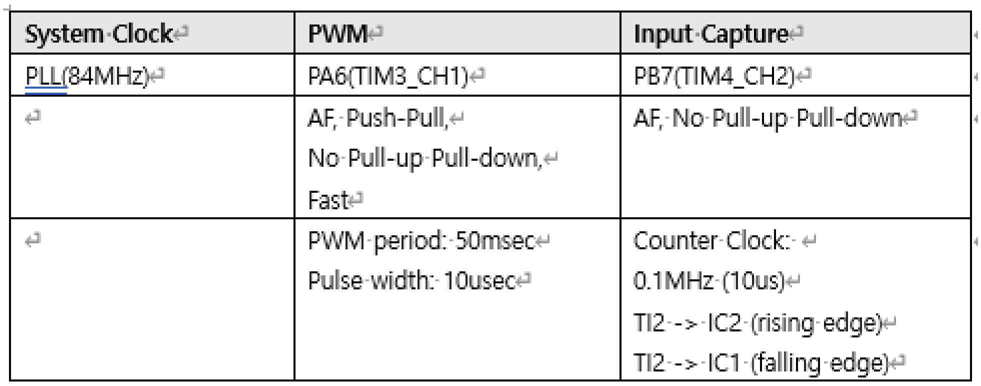
Type	Port - Pin	Configuration
System Clock		PLL 84MHz
USART2: PC For verification		No Parity, 8-bit Data, 1-bit Stop bit, 9600 baud-rate
USART1: PC – MCU Blue tooth	TXD: PA9 RXD: PA10	No Parity, 8-bit Data, 1-bit Stop bit, 9600 baud-rate

ADC



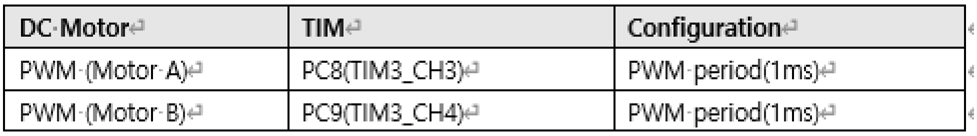
TIMER	ADC	GPIO
TIM2	ADC1_CH8 (1st channel) ADC1_CH9 (2nd channel)	PB_0 PB_1
Up-Counter, Counter CLK 1kHz OC1M(Output Compare 1 Mode) : PWM mode 1 Master Mode Selection: (TRGO) OC1REF	ADC Clock Prescaler /8 12-bit resolution, right alignment Single Conversion mode Scan mode: Two channels in regular group External Trigger (Timer3 TRGO) @1kHz Trigger Detection on Rising Edge	Analog Mode No Pull-up Pull-down

Ultrasonic Distance Sensor



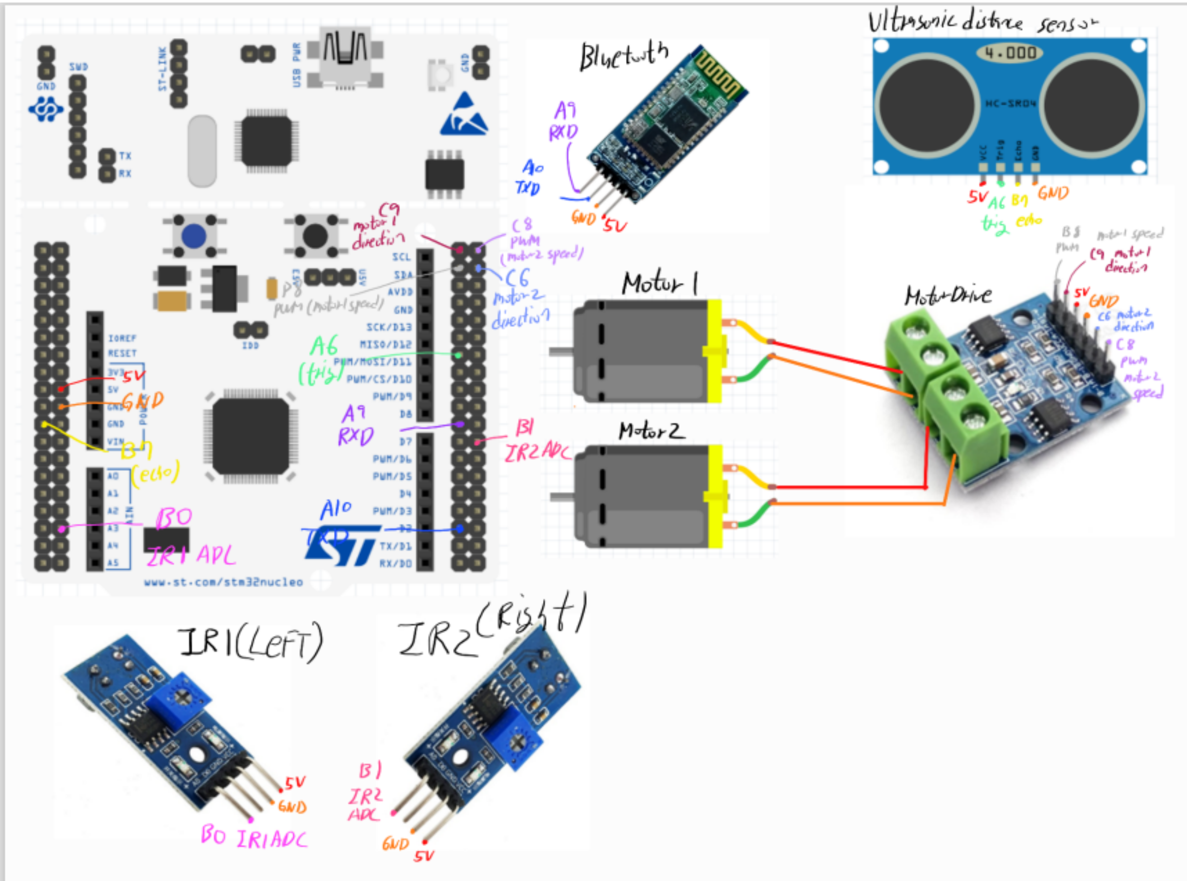
System Clock	PWM	Input Capture
PLL(84MHz)	PA6(TIM3_CH1)	PB7(TIM4_CH2)
	AF, Push-Pull, No Pull-up Pull-down, Fast	AF, No Pull-up Pull-down
	PWM period: 50msec Pulse width: 10usec	Counter Clock: 0.1MHz (10us) TI2 -> IC2 (rising edge) TI2 -> IC1 (falling edge)

Motor



DC Motor	TIM	Configuration
PWM (Motor A)	PC8(TIM3_CH3)	PWM period(1ms)
PWM (Motor B)	PC9(TIM3_CH4)	PWM period(1ms)

Circuit Diagram



Code

Your code goes here: https://github.com/Ohjeahyun1/EC-jeahyun-447/blob/8c945375c7e959bab20aede7e55385b616ff5112/lab/LAB_Final_Project_main_2.c

https://github.com/Ohjeahyun1/EC-jeahyun-447/tree/main/include

Explain your source code with necessary comments.

Description with Code

- Code Initialization

```
// Initialiization
void setup(void)
{
    RCC_PLL_init();                                                    // 84Mhz
clock
    // USART congfiguration
    USART_init(USART2, 9600);                                          // USART2
For veritication
    USART_begin(USART1, GPIOA, 9, GPIOA,10, 9600);                     // USART 1
Blue Tooth PA9 - RXD , PA10 - TXD
    // ADC setting
  ADC_init(GPIOB, 0, TRGO);                                          //ch8 IR1
LEFT
    ADC_init(GPIOB, 1, TRGO);                                          //ch9 IR2
RIGHT

    // ADC channel sequence setting
    ADC_sequence(2, seqCHn);
    // ADON, SW Trigger enable
    ADC_start();

        // secrect Door on off switch
    GPIO_init(GPIOC, BUTTON_PIN, INPUT);                              // calls
RCC_GPIOC_enable() and button pin mode -> input
    GPIO_pupdr(GPIOC, BUTTON_PIN, EC_PU);                             // GPIOC
button pin pupdr -> pull up
  EXTI_init(GPIOC,BUTTON_PIN,FALL,0);                               //EXTI
button PIN -> trigger type(falling),propriority(0)


    // PWM configuration --------------------------------------------------------
-------------
    PWM_t trig;
           // PWM(TIM3_CH1) for trig
    PWM_init(&trig,GPIOA,6,UP,SFAST,PP,EC_NOPUPD,1);
// PA_6: Ultrasonic trig pulse
    PWM_period_us(&trig, 50000);                                        // PWM
of 50ms period. Use period_us()
    PWM_pulsewidth_us(&trig, 10);                                       // PWM
pulse width of 10us


    //Motor speed PWM init
    PWM_init(&dcPwm[A], dcPwmPin[A].port, dcPwmPin[A].pin,UP,SFAST,PP,EC_PU,1);
    PWM_init(&dcPwm[B], dcPwmPin[B].port, dcPwmPin[B].pin,UP,SFAST,PP,EC_PU,1);

    PWM_period_ms(&dcPwm[A], 1);                                      // Motor
period 1ms
    PWM_period_ms(&dcPwm[B], 1);                                      // Motor
period 1ms

    //Motor direction init
    for (int i = 0; i < 2; i++){
        GPIO_init(dcDirPin[i].port, dcDirPin[i].pin, OUTPUT);
        GPIO_pupdr(dcDirPin[i].port, dcDirPin[i].pin, EC_PD);
        GPIO_otype(dcDirPin[i].port, dcDirPin[i].pin, PP);
        GPIO_ospeed(dcDirPin[i].port, dcDirPin[i].pin, SHIGH);
    }

    GPIO_write(dcDirPin[A].port, dcDirPin[A].pin, mode);
    GPIO_write(dcDirPin[B].port, dcDirPin[B].pin, mode);


    // Input Capture configuration ----------------------------------------------
-------------------------

    IC_t echo;
           // Input Capture for echo
    ICAP_init(&echo,GPIOB,7 ,EC_NOPUPD);                                  // PB7
as input caputre
    ICAP_counter_us(&echo, 10);                                         // ICAP
counter step time as 10us
    ICAP_setup(&echo, 2, IC_RISE);                                      //
TIM4_CH2 as IC2 , rising edge detec
```

```
    ICAP_setup(&echo,1,IC_FALL);                                      //
TIM4_CH1 as IC1 , falling edge detec

}
```

- Time interrupt

To check the obstacles in front of the car

```
// Detect the obstacle using Ultrasonic distance sensor
void TIM4_IRQHandler(void){
    if(is_UIF(TIM4)){                                                   //
Update interrupt
        ovf_cnt++;
               // overflow count
        clear_UIF(TIM4);
// clear update interrupt flag
    }
    if(is_CCIF(TIM4, 2)){
   // TIM4_Ch2 (IC2) Capture Flag. Rising Edge Detect
        time1 = TIM4->CCR2;
       // Capture TimeStart
        clear_CCIF(TIM4, 2);                                              //
clear capture/compare interrupt flag
    }
    else if(is_CCIF(TIM4, 1)){
   // TIM4_Ch1 (IC1) Capture Flag. Falling Edge Detect
        time2 = TIM4->CCR1;
         // Capture TimeEnd
        if((time2-time1)<(TIM4->ARR+1)&(ovf_cnt==1)) ovf_cnt=0;         // if
(time2-time1)< ARR+1 make over count 0
        timeInterval = ((time2-time1)+(TIM4->ARR+1)*ovf_cnt)/100;           //
Total time of echo pulse (10us * counter pulses -> [msec] unit)
        ovf_cnt = 0;                                                      //
overflow reset
        clear_CCIF(TIM4, 1);
       // clear capture/compare interrupt flag
    }
}
```

- ADC

Change the value of the IR sensors from analog to digital for line tracing

```
// ADC IR sensor
void ADC_IRQHandler(void){
    if((is_ADC_OVR())){
        clear_ADC_OVR();
    }

    if(is_ADC_EOC()){       //after finishing sequence
            if (flag==0){
                IR1 = ADC_read();
            }
            else if (flag==1){
                IR2 = ADC_read();
            }
        flag =! flag;
    }
}
```

- EXTI

Auto parking mode On

```
// Button pressed park mode On
void EXTI15_10_IRQHandler(void) {
    if (is_pending_EXTI(BUTTON_PIN)){     // when button pressed
   park = 1;                           // Park mode On
     clear_pending_EXTI(BUTTON_PIN);     // cleared by writing '1'
    }

}
```

- USART 1 (BlueTooth)

Using Blue Tooth to send command from PC to MCU

```
// Button pressed park mode On
void EXTI15_10_IRQHandler(void) {
    if (is_pending_EXTI(BUTTON_PIN)){     // when button pressed
   park = 1;                           // Park mode On
     clear_pending_EXTI(BUTTON_PIN);     // cleared by writing '1'
    }

}
```

Main Code

- USART 1 (Blue Tooth)

Send Commands that to change gear, direction of the car,etc from PC to MCU

```
//USART 1 send command PC to MCU
        if (bReceive == 1){
            // Forward command (Gear D) Backward command (Gear R)
            if (mcu2Data == 'w' ){       // w input
                if(mode == 1){             // Gear Diving
                Right =0.38;
                Left =0.3;
                }else{                     // Gear Reverse
                Right = 0.7;
                Left = 0.8;
                }
            }else if(mcu2Data == 'd' ){  //d input
      // Right command (Gear D,R)
                if(mode == 1)   {          // Gear Diving
                        Right = 0.38;
                        Left = 1.0;
                    }
                else {                     // Gear Reverse
                    Right = 1.0;
                    Left = 0.5;
                }
            }else if(mcu2Data == 'a' ){  //a input
            // LEFT command (Gear D,R)
                if(mode == 1)               {    // Gear Diving
                    Right = 1.0;
                    Left = 0.2;
                }
                else {                     // Gear Reverse
                    Right = 0.3;
                    Left = 1.0;
                }
            }else if(mcu2Data == 's' ){  // S input
                // Change Gear command (Gear D <-> R)
                mode ^= 1;                 // Gear change
                // Change Motor direction
    GPIO_write(dcDirPin[A].port, dcDirPin[A].pin, mode);
    GPIO_write(dcDirPin[B].port, dcDirPin[B].pin, mode);
                // Change Gear and Car stop
                if(mode == 1){             // Gear Diving
                Right = 1.0;
                Left = 1.0;
                }else {                    // Gear Reverse
                Right = 0.0;
                Left = 0.0;
                }
            } else if(mcu2Data == 'p'){  // P input
                // Auto Paking Gear(Mode) command
                park = 1;
            }else if(mcu2Data == 'x') {  // X input
                // Car stop command (Gear D,R) and Auto parking stop
                park = 0;
                if(mode == 1){             // Gear Diving
                Right = 1.0;
                Left = 1.0;
                }else {                    // Gear Reverse
                Right = 0.0;
                Left = 0.0;
                }
            }

            bReceive = 0;

        }
```

- Auto parking mode

Auto parking using Line Tracing with IR sensors

```
if (IR1 > 1000) DIR_flag = RIGHT;                      //when IR1 sensor detected the line out RIGHT FLAG
        else if (IR2 > 1000) DIR_flag = LEFT;                  //when IR2 sensor detected the line out LEFT FLAG
        else if (IR1 < 999 && IR2 <999) DIR_flag = FORWARD;    //FORWARD FLAG

        // Auto parking mode
        if(park == 1){
            if (DIR_flag == FORWARD ){                           //FORWARD input
```

```
                Right =0.52;                                       //0.62
                Left =0.5;                                         //0.6
            }else if(DIR_flag == LEFT ){                         //LEFT input
                        Right = 0.4;                                   // 0.4
                        Left = 0.8;                                    // 1.0
            }else if(DIR_flag == RIGHT ){                        //RIGHT input
                    Right = 0.8;                                     // 1.0
                    Left = 0.5;                                      // 0.5
            }
        }
```

- Obstacle detect

Stop when sensors are used to detect obstacles in front of the vehicle

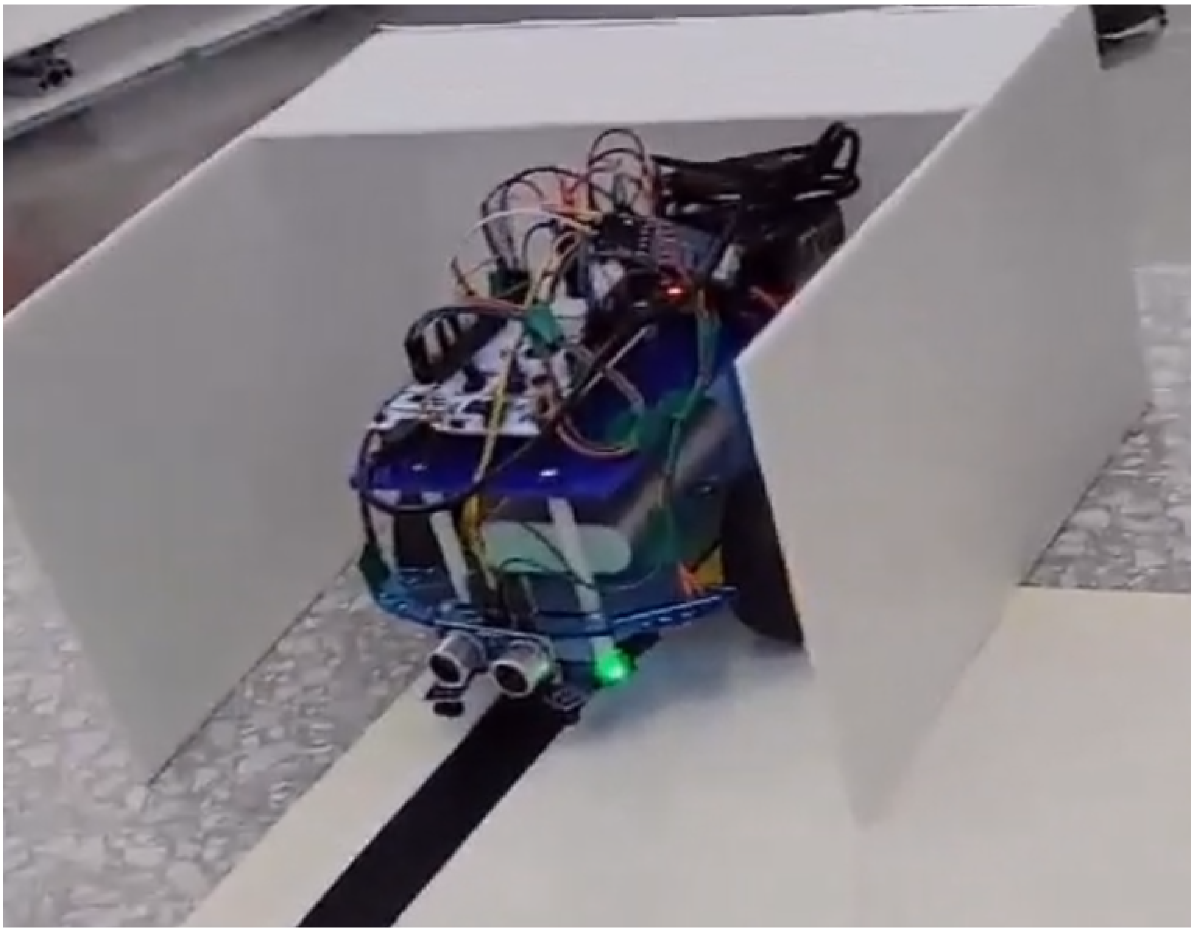
```
distance = (float) timeInterval * 340.0 / 2.0 / 10.0;  // [mm] -> [cm]


        if(distance  < 6.0) {                                      //if obstacle detected
            k++;                                                 // obstacle flag
        }else if(distance > 10.0){                             //when obstacle run out flag clear
            k = 0;
        }
        if(k == 1){                                            // obstacle flag on
            if(mode == 1 && park == 0){                          // Car stop when Gear D,P(Auto parking mode)
                Right = 1.0;
                Left = 1.0;
                }
        }
```

- Motor speed change

```
// change the Motor PWM with command
    PWM_duty(&dcPwm[A], Right);
    PWM_duty(&dcPwm[B], Left);
```

Results

Experiment images and results



When the command was given from the PC to the MCU, the car was able to go straight, turn right,left, stop, change gears and etc.

In addition, when the automatic parking command was issued, automatic parking could be performed using line tracing.

Add demo video link: https://youtu.be/uKYzLiGAYlo

Complement

1. The original plan was to use zigbee communication between the car and the house to stop using a distance sensor in the parking lot when parking, but this plan was not used because it thought a remote monitoring system of a smart home was necessary. However, it may have been possible if other elements were used.
2. There was a problem that the output of the motor was not always uniform, so that it could not be driven well during actual operation. If I buy another DC motor or use a stepper motor, this part will be solved. Another way would have been to add a code to improve or down the motor's output.
3. Line tracing was also not always fully run. The first reason is the power problem of the motor, and even though the command was given to go forward, the motor did not work. The second is the problem of the line, but I think there was a problem that the IR sensor did not work well because the line was made too thick. The first problem will be solved with two solutions. The second problem could be solved if Line was made through more trial and error.
4. It would have been better to install a black box like a real car or use a rear camera or distance sensor to use it in parking or general driving.

Reference

Complete list of all references used (github, blog, paper, etc)