

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA



KHOA ĐIỆN – ĐIỆN TỬ



BÁO CÁO BÀI TẬP LỚN

Thiết Kế Hệ Thống Nhúng

**Hệ thống bãi đỗ xe thông minh sử dụng
Bluetooth và Cảm biến hồng ngoại**

Giảng viên hướng dẫn: Thầy Nguyễn Trung Hiếu

<i>Họ và tên</i>	<i>MSSV</i>	<i>% tham gia</i>
Tổng Phước Thịnh	2010657	34 %
Trương Trần Tuấn Kiệt	2010366	33 %
Đình Bá Duy	2010179	33 %

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

MỤC LỤC

<u>Chương 0.</u> Lời cảm ơn -----	Trang 1
<u>Chương 1.</u> Giới thiệu đề tài -----	Trang 2
1.1 Đặt vấn đề-----	Trang 2
1.2 Nội dung nghiên cứu -----	Trang 2
<u>Chương 2.</u> Lý thuyết -----	Trang 3
1.1 Vi điều khiển STM32F103 -----	Trang 3
1.2 Module Bluetooth HC06-----	Trang 3
1.3 IR Proximity-----	Trang 5
1.4 Servo-----	Trang 5
1.5 LCD I ² C-----	Trang 7
<u>Chương 3.</u> Thiết kế phần cứng-----	Trang 10
<u>Chương 4.</u> Thiết kế phần mềm-----	Trang 11
<u>Chương 5.</u> Kết luận và hướng phát triển-----	Trang 17

Chương 0. Lời cảm ơn

Chúng em xin gửi lời cảm ơn chân thành đến Khoa..., Trường... đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành đề tài nghiên cứu này. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy/cô... đã dày công truyền đạt kiến thức và hướng dẫn chúng em trong quá trình làm bài.

Em đã cố gắng vận dụng những kiến thức đã học được trong học kỳ qua để hoàn thành bài tiểu luận. Nhưng do kiến thức hạn chế và không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất kính mong sự góp ý của quý thầy cô để bài tiểu luận của em được hoàn thiện hơn.

Một lần nữa, em xin trân trọng cảm ơn sự quan tâm giúp đỡ của các thầy cô đã giúp đỡ em trong quá trình thực hiện bài tiểu luận này.

Xin trân trọng cảm ơn!

Chương 1. Giới thiệu đề tài

1.1 Đặt vấn đề:

Từ lâu, cùng với sự phát triển của các đô thị thì nhu cầu đỗ xe cũng tăng theo. Hiện nay, đã có nhiều hệ thống bãi giữ xe giúp cho con người có được sự thuận tiện và giảm chi phí vận hành cho bộ phận quản lý.

Yêu cầu của bãi đỗ xe thông minh là có thể tối ưu chi phí quản lý, yếu tố thẩm mỹ và hơn hết là sự an toàn tài sản của người đỗ.

1.2 Nội dung nghiên cứu:

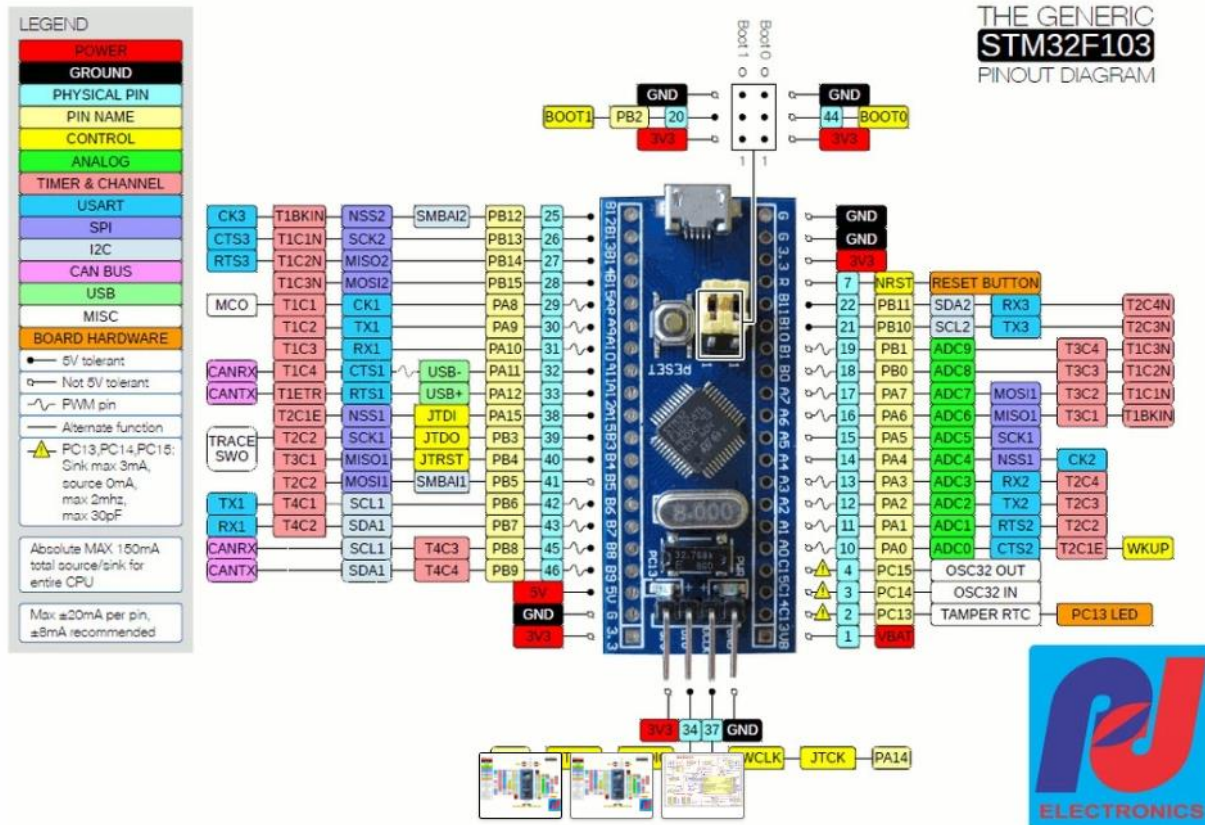
Với yêu cầu đó thì nhóm chúng em trình bày đề tài bãi giữ xe thông minh nhằm giải quyết những vấn đề trên. Chúng em tập trung vào nghiên cứu những vấn đề sau:

- Kit Bluepill STM32F103C8T6
- Truyền nhận tín hiệu qua Bluetooth
- Cảm biến vật cản hồng ngoại IR – Proximity
- Động cơ Servo

Chương 2. Lý thuyết

2.1 Vi điều khiển Bluepill STM32F103C8T6

Sơ đồ chân:

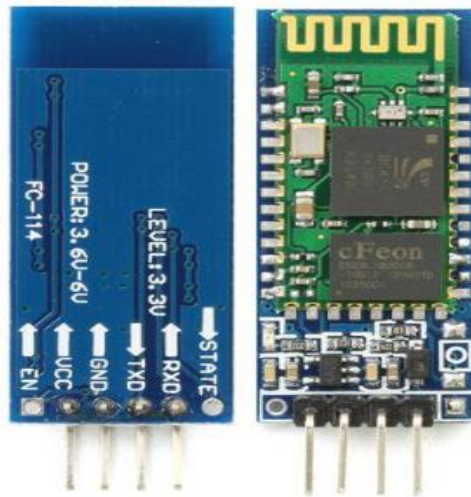


Thông tin module:

- Tên : STM32F103C8T6 - ARM 32 CPU Cortex-M3.
- Điện áp hoạt động : 2V-3.6V DC
- Chế độ Debug: SWD.
- Tần số làm việc : 72MHz.
- Bộ nhớ flash : 64K
- SRAM : 20K
- Tích hợp USB để cấp nguồn và giao tiếp.
- Hỗ trợ các chuẩn giao tiếp : CAN, I2C, SPI UART/USART, USB.

2.2 Module giao tiếp Bluetooth HC-06

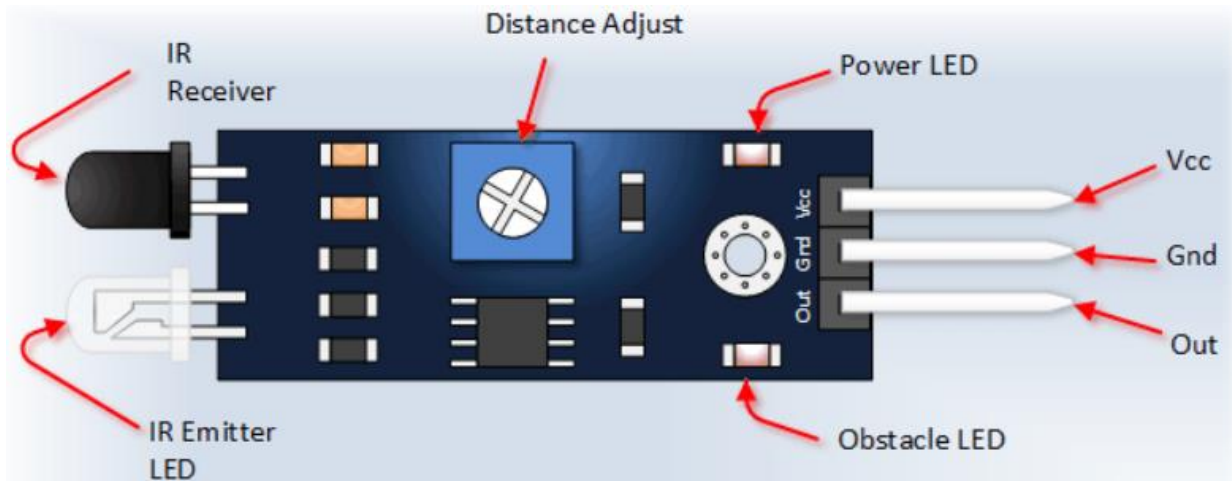
Đây là module sử dụng tập lệnh AT (at commands). Được để ở cấu hình mặc định giúp truyền nhận tín hiệu qua UART với baudrate mặc định: 9600.



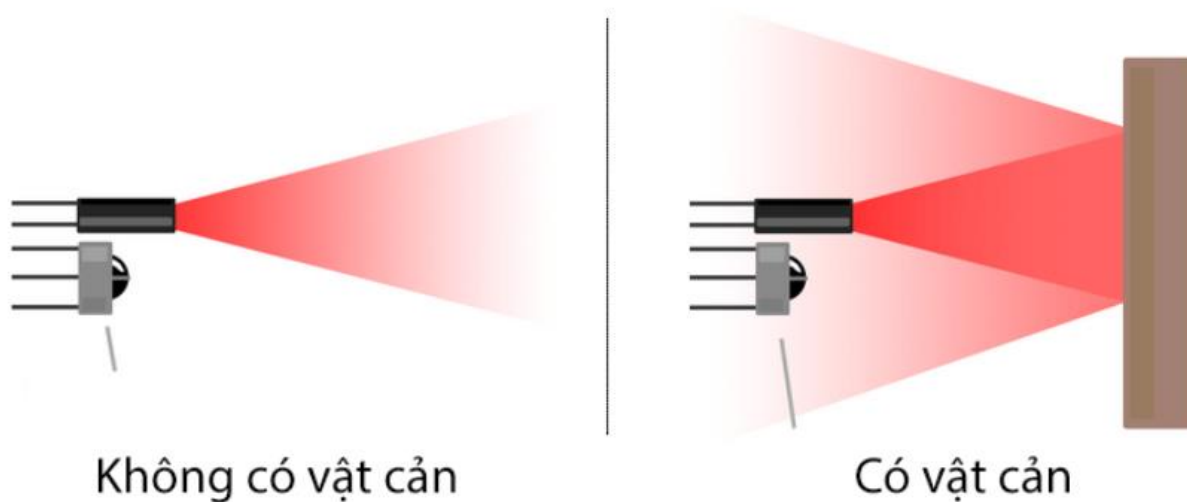
THÔNG SỐ KỸ THUẬT MODULE THU PHÁT BLUETOOTH HC-06:

- Điện áp hoạt động: 3.3 ~ 5VDC.
- Điện áp giao tiếp: TTL tương thích 3.3VDC và 5VDC.
- Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Dải tần sóng hoạt động: Bluetooth 2.4GHz
- Sử dụng CSR mainstream bluetooth chip, bluetooth V2.0 protocol standards.
- Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA
- Kích thước của module chính: 28 mm x 15 mm x 2.35 mm
- Thiết lập mặc định:
Baud rate: 9600, N, 8, 1.
Pairing code: 1234.

2.3 Cảm biến hồng ngoại IR Proximity



Có 2 loại cảm biến hồng ngoại là *cảm biến hồng ngoại chủ động* và *cảm biến hồng ngoại bị động*. Trong bài này sẽ sử dụng cảm biến chủ động.



Cảm biến hồng ngoại chủ động: Gồm 2 thành phần: nguồn phát hồng ngoại sử dụng một bóng đèn LED để phát ra tia sóng hồng ngoại và cảm biến hồng ngoại. Khi gặp vật thể, tia hồng ngoại bật ngược lại và đi vào cảm biến hồng ngoại.

2.4 Động cơ servo

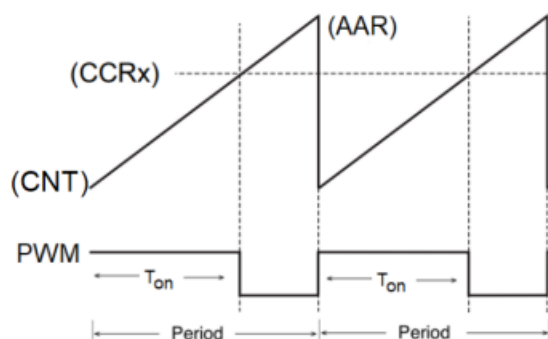
Có 2 loại động cơ Servo là : AC và DC. Trong đề tài, nhóm em sẽ sử dụng Servo DC để phù hợp với mô hình.

Chế độ hoạt động servo được hình thành bởi những hệ thống hồi tiếp vòng kín. Động cơ servo nhận một tín hiệu xung điện (PWM) từ bộ điều khiển để hoạt động và được kiểm soát bằng bộ mã hóa (encoder).



Nguyên lí tạo PWM:

- Khi thanh ghi CNT bắt đầu đếm thì ngõ ra PWM sẽ ở mức 1.
- Khi CNT đến bằng giá trị trong thanh ghi CCRx thì ngõ ra PWM sẽ bị reset về 0.
- Khi CNT tăng đến bằng giá trị ARR thì CNT sẽ bị reset về 0, và ngõ ra PWM sẽ đặt lên 1.
- Tiếp tục chu trình như vậy sẽ tạo ra dạng sóng PWM.



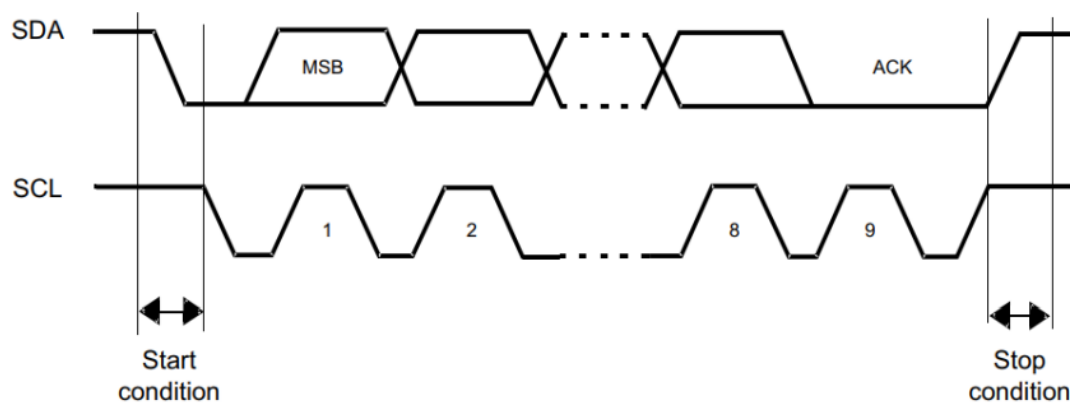
Công thức tính tần số PWM:
$$f_{\text{pwm}} = \frac{F_{\text{Timer}}}{(PSC+1)(ARR+1)}$$

2.5 Màn hình LCD I²C

I²C (Inter – Integrated Circuit) là 1 giao thức giao tiếp nối tiếp đồng bộ được phát triển bởi Philips Semiconductors, sử dụng để truyền nhận dữ liệu giữa các IC với nhau chỉ sử dụng hai đường truyền tín hiệu. GIAO TIẾP I²C sử dụng 2 đường truyền tín hiệu:

SCL - Serial Clock Line : Tạo xung nhịp đồng hồ do Master phát đi.

SDA – Serial Data Line : Đường truyền nhận dữ liệu.



Quá trình truyền dữ liệu đến thiết bị I²C:

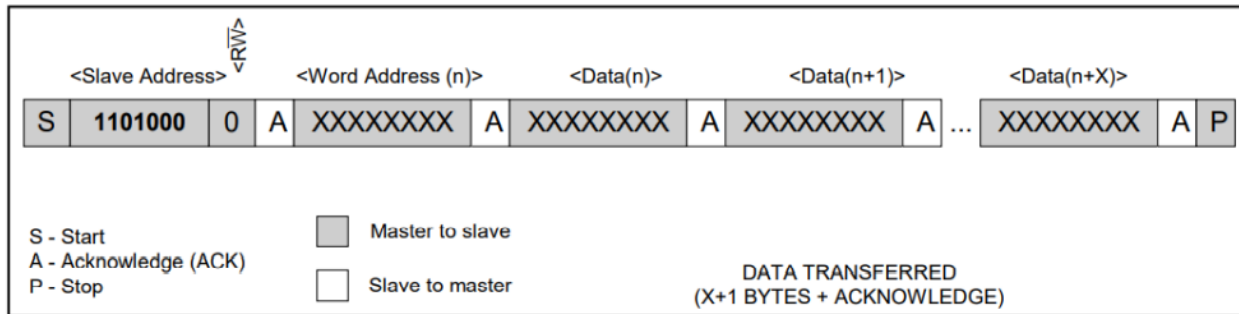
Bước 1: Master gửi tín hiệu start I²C.

Bước 2: Master gửi địa chỉ của thiết bị I²C (7 bit) kèm bit Write (bit 0).

Bước 3: Master gửi địa chỉ của thanh ghi dữ liệu của Slave muốn ghi giá trị.

Bước 4: Master gửi giá trị mà muốn ghi vào thanh ghi ở bước 3.

Bước 5: Master tạo tín hiệu stop.



Quá trình nhận dữ liệu của thiết bị I2C:

Bước 1: Master gửi tín hiệu start I2C

Bước 2: Master gửi địa chỉ của thiết bị I2C (7 bit) kèm bit Write (bit 0).

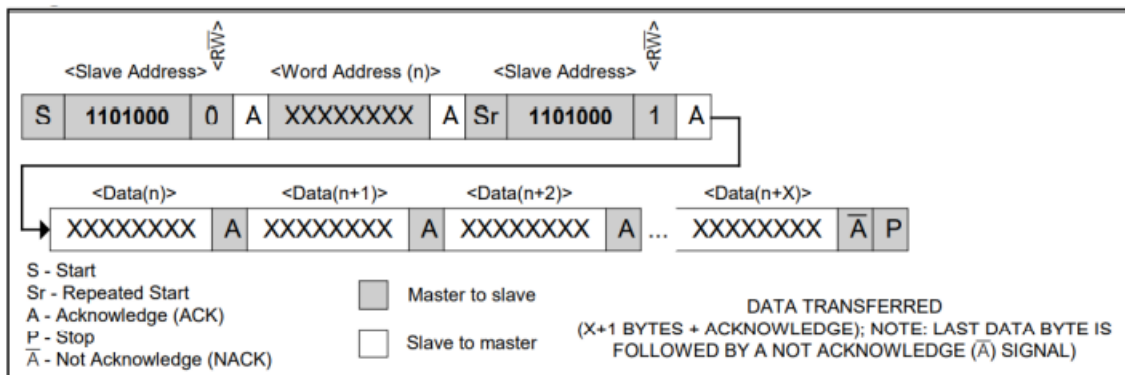
Bước 3: Master gửi địa chỉ thanh ghi của Slave mà muốn đọc dữ liệu.

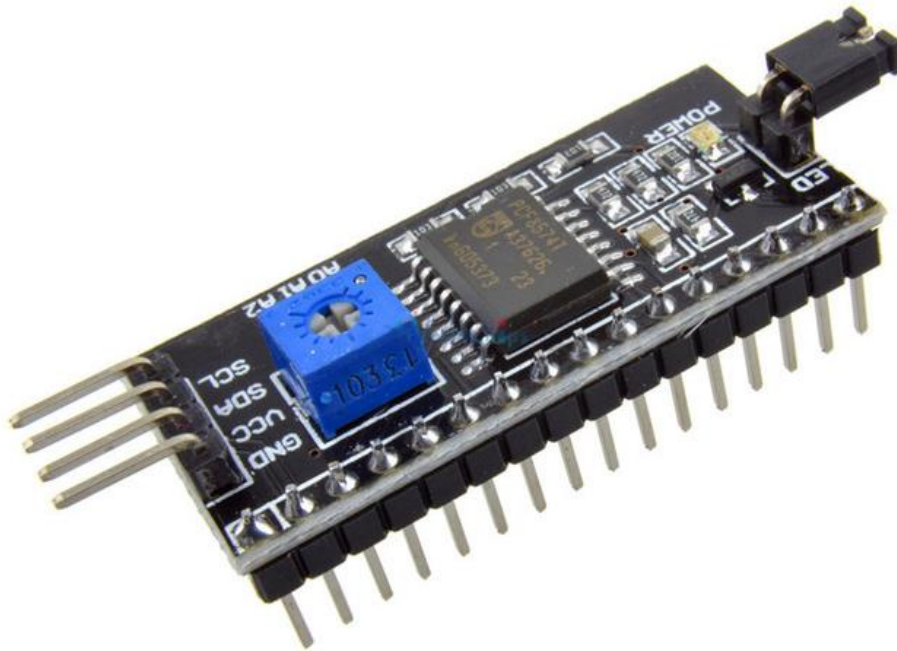
Bước 4: Master gửi tín hiệu Repeated Start.

Bước 5: Master gửi địa chỉ của thiết bị I2C (7 bit) kèm bit Read (bit 1);

Bước 6: Master đọc dữ liệu chứa trong thanh ghi ở bước 3 từ Slave gửi về.

Bước 7: Master tạo tín hiệu Stop.

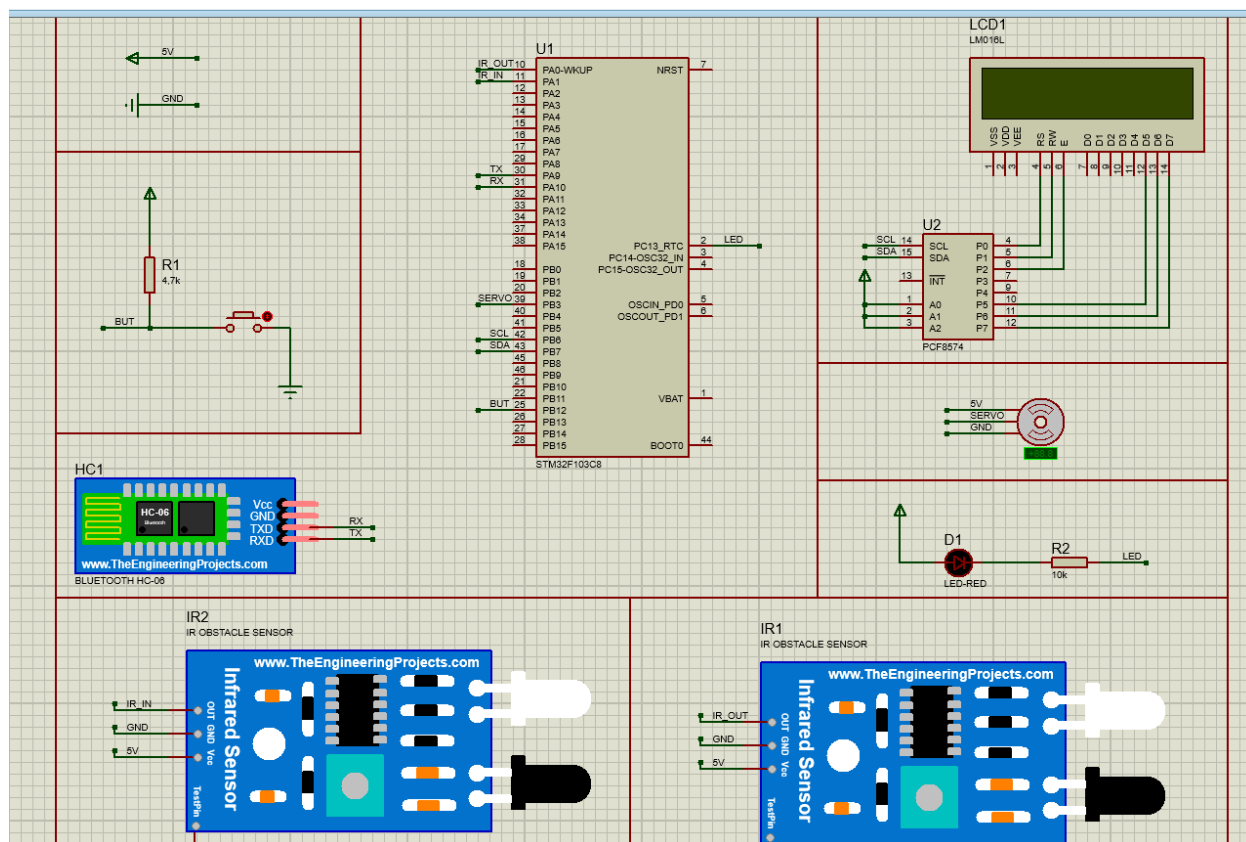




Module I2C LCD 16x2

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16x2 (RS, EN, D7, D6, D5 và D4) thì module IC2 thì chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

Chương 3. Thiết kế phần cứng



Khởi Nguồn 5V: được lấy từ mạch nạp ST- Link.

Khởi nút nhấn: Dùng để chỉnh chế độ sử dụng Cảm biến hoặc giao tiếp Bluetooth.

Khởi LED indicator: Chọn Mode.

LED sáng: giao tiếp Bluetooth.

LED tắt: sử dụng Cảm biến.

HC-06: Giao tiếp Bluetooth giữa điện thoại và hệ thống qua UART kết nối với chân PA9 và PA10 của Vi điều khiển.

Khởi cảm biến hồng ngoại: Gồm 2 cảm biến được gắn bên trong và ngoài bãi xe. Giá trị được trả về là 1 nếu phát hiện vật cản (xe), là 0 nếu không phát hiện vật cản.

Servo: Được dùng để điều chỉnh thanh chắn trước bãi xe.

Màn hình LCD: Hiển thị thông điệp giúp người lái xe dễ dàng vào bãi.

Chương 4. Thiết kế phần mềm

- Khai báo Thư viện và LCD:

```
/* Includes -----  
#include "main.h"  
  
/* Private includes -----  
/* USER CODE BEGIN Includes */  
#include "LiquidCrystal_I2C.h"  
#include "string.h"  
LiquidCrystal_I2C lcd;  
/* USER CODE END Includes */
```

- Khai báo các giao tiếp ngoại vi và Timer:

```
/* Private variables -----*/  
I2C_HandleTypeDef hi2c1;  
  
TIM_HandleTypeDef htim2;  
  
UART_HandleTypeDef huart1;
```

- Khai báo biến:

```
/* USER CODE BEGIN 0 */  
static uint8_t carSlot= 3;      //Cho trong trong bãi  
  
const uint16_t angle;  
uint8_t Rxdata;  
uint8_t Rxbuff[100];  
uint8_t flagOn=0, flagOff=0;  
//uint8_t index = 0;  
uint8_t btn_sta;  
  
uint8_t mode=0;
```

Chú ý: Đối với biến carSlot là biến *static* là biến sẽ được khởi tạo 1 lần duy nhất từ khi khởi động chương trình để tránh xảy ra hiện tượng sai số lượng xe trong bãi.

- Chương trình LCD:

```
//=====LCD=====
void displayGreeting()
{
    lcd_set_cursor(&lcd,0, 4);
    lcd_printf(&lcd,"WELCOME!!!");
    lcd_set_cursor(&lcd,1, 2);
    if (carSlot >0)
        lcd_printf(&lcd,"Slot left: %d",carSlot-1);
    else if( carSlot==0)
        lcd_printf(&lcd,"Slot left: %d",carSlot);
    if(carSlot== 0)
        carSlot--0;
    else if( carSlot>0)
        carSlot--1;
}

void displayGoodbye()
{
    lcd_set_cursor(&lcd,0, 4);
    lcd_printf(&lcd,"GOODBYE!!!");
    lcd_set_cursor(&lcd,1, 2);
    lcd_printf(&lcd,"See you again.");
    if(carSlot==3)
        carSlot+=0;
    else if( carSlot<3)
        carSlot+=1;
}

void warningFull()
{
    lcd_set_cursor(&lcd,0, 4);
    lcd_printf(&lcd,"SORRY!!!");
    lcd_set_cursor(&lcd,1, 2);
    lcd_printf(&lcd,"Full Slot. :(");
}
```

- Chương trình Servo:

```
//=====Servo=====
long map(long x, long in_min, long in_max, long out_min, long out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

void sv_write(uint8_t angle)    //Angle: 0->180
{
    uint16_t ccr= map(angle, 0,180, 400,2500);
    __HAL_TIM_SetCompare(&htim2, TIM_CHANNEL_2, ccr);
}

//=====
```

- Chương trình HC – 06:

```
//=====BLUETOOTH=====
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if (huart->Instance == huart1.Instance)
    {
        //  Rxbuff[index] =Rxdata;
        //  index++;
        if(Rxdata == 'a')
        {
            flagOn=1;
        }
        else if(Rxdata == 'b')
        {
            flagOff=1;
        }
        HAL_UART_Receive_IT(&huart1,&Rxdata,1);
    }
}
//=====
```

- Chương trình Nút nhấn và chống rung:

```
btn_sta= HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12);
if(btn_sta== 0)
{
    HAL_Delay(20);
    if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12) ==0)
        mode ++;
    HAL_GPIO_TogglePin(GPIOC,GPIO_PIN_13);
    while (HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12) ==0);
}
```

- Vòng lặp while(1) của chương trình chính:

```

191  /* USER CODE BEGIN WHILE */
192  while (1)
193  {
194      btn_sta= HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12);
195      if(btn_sta== 0)
196      {
197          HAL_Delay(20);
198          if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12) ==0)
199              mode ++;
200          HAL_GPIO_TogglePin(GPIOC,GPIO_PIN_13);
201          while (HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_12) ==0);
202      }
203
204      if(mode %2 ==0)
205      {
206          HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
207          if(carSlot == 3)          //KHONG CO XE TRONG BAI
208          {
209              if ( HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) == 0 )          //IR_Ngoai
210              {
211                  displayGreeting();
212                  sv_write(90);
213                  HAL_Delay(2500);
214                  lcd_clear_display(&lcd);
215                  sv_write(0);
216              }
217
218              else if ( HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) ==0) //IR_Trong
219              {
220                  displayGoodbye();
221                  sv_write(90);
222                  HAL_Delay(2500);
223                  lcd_clear_display(&lcd);
224                  sv_write(0);
225              }
226          }
227          else if (carSlot < 3 && carSlot >0)          //CO XE TRONG BAI
228          {
229              if ( HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) == 0 )          //IR_Ngoai
230              {
231                  displayGreeting();
232                  sv_write(90);
233                  HAL_Delay(2500);
234                  lcd_clear_display(&lcd);
235                  sv_write(0);
236              }
237              else if ( HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_1) == 0 ) //IR_Trong
238              {
239                  displayGoodbye();
240                  sv_write(90);
241                  HAL_Delay(2500);
242                  lcd_clear_display(&lcd);
243                  sv_write(0);
244              }
245          }

```

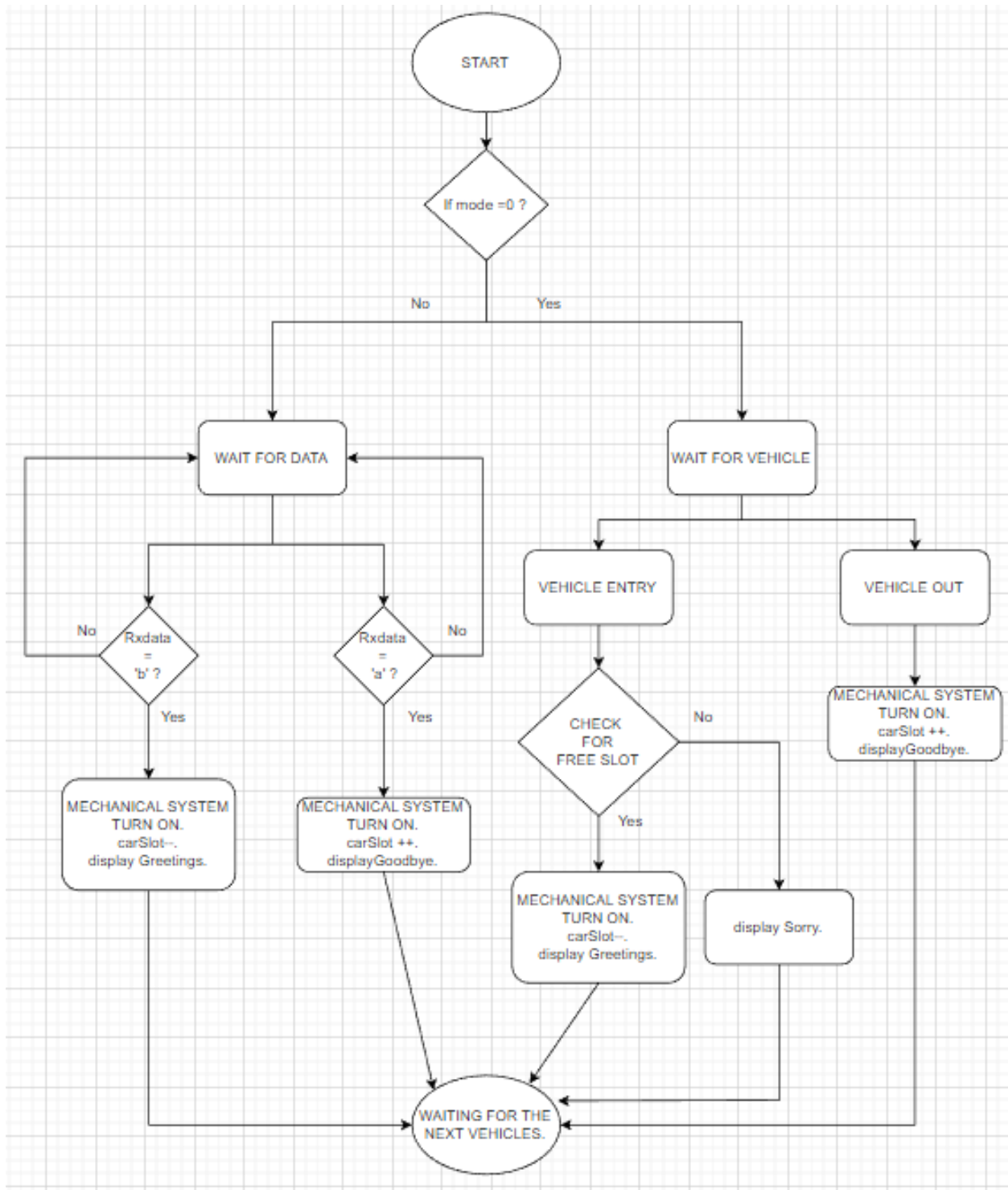


```

246     else if ( carSlot == 0 ) //FULL
247     {
248         if ( HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0) == 0 )    //IR_Ngoai
249         {
250             warningFull();
251             HAL_Delay(2500);
252             lcd_clear_display(&lcd);
253         }
254         else if ( HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_1) == 0 ) //IR_Trong
255         {
256             displayGoodbye();
257             sv_write(90);
258             HAL_Delay(2500);
259             lcd_clear_display(&lcd);
260             sv_write(0);
261         }
262     }
263     HAL_Delay(1000);
264 }
265
266     else if (mode %2 != 0)
267     {
268         HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
269         if(flagOn)
270         {
271             displayGreeting();
272             sv_write(90);
273             HAL_Delay(2500);
274             lcd_clear_display(&lcd);
275             sv_write(0);
276             flagOn =0;
277         }
278
279         else if (mode %2 != 0)
280         {
281             HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
282             if(flagOn)
283             {
284                 displayGreeting();
285                 sv_write(90);
286                 HAL_Delay(2500);
287                 lcd_clear_display(&lcd);
288                 sv_write(0);
289                 flagOn =0;
290             }
291
292             if(flagOff)
293             {
294                 displayGoodbye();
295                 sv_write(90);
296                 HAL_Delay(2500);
297                 lcd_clear_display(&lcd);
298                 sv_write(0);
299                 flagOff =0;
300             }
301         }
302     }
303
304     /* USER CODE END WHILE */

```

Lưu đồ thuật toán – Block Diagram:



Chương 5. Kết luận và hướng phát triển

Đứng trước xu hướng phát triển của công nghệ thông tin như vũ bão, việc hệ thống hoá việc quản lý là vô cùng quan trọng và cấp thiết. Nó sẽ giúp cho các tổ chức nhà nước giải quyết các công việc được nhanh chóng và hiệu quả. Ứng dụng cơ sở dữ liệu đã giải quyết được vấn đề đó. Với đề tài: Bãi giữ xe thông minh sử dụng Bluetooth và cảm biến của nhóm chúng em kì vọng sẽ giải quyết được vấn đề về chi phí quản lý, an ninh ở các khu đô thị, đặc biệt là nơi đỗ xe, góp phần vào sự phát triển của xã hội.