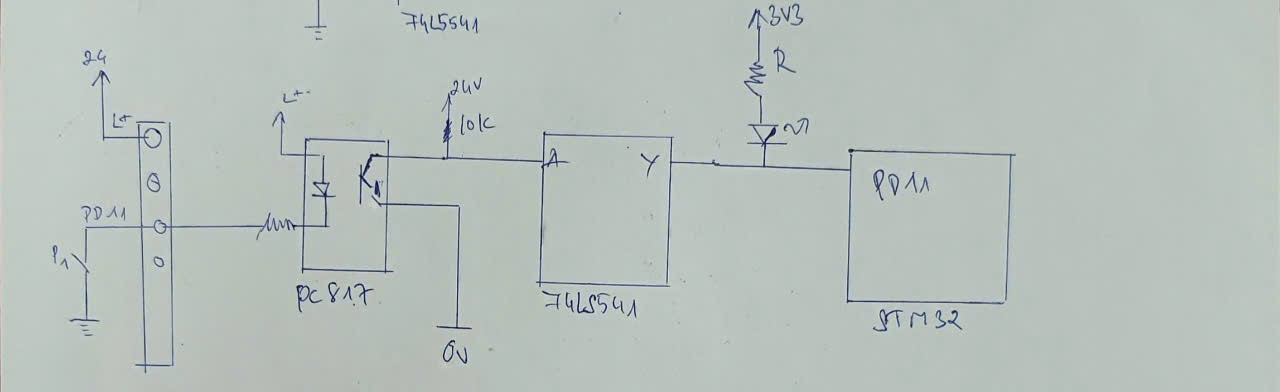
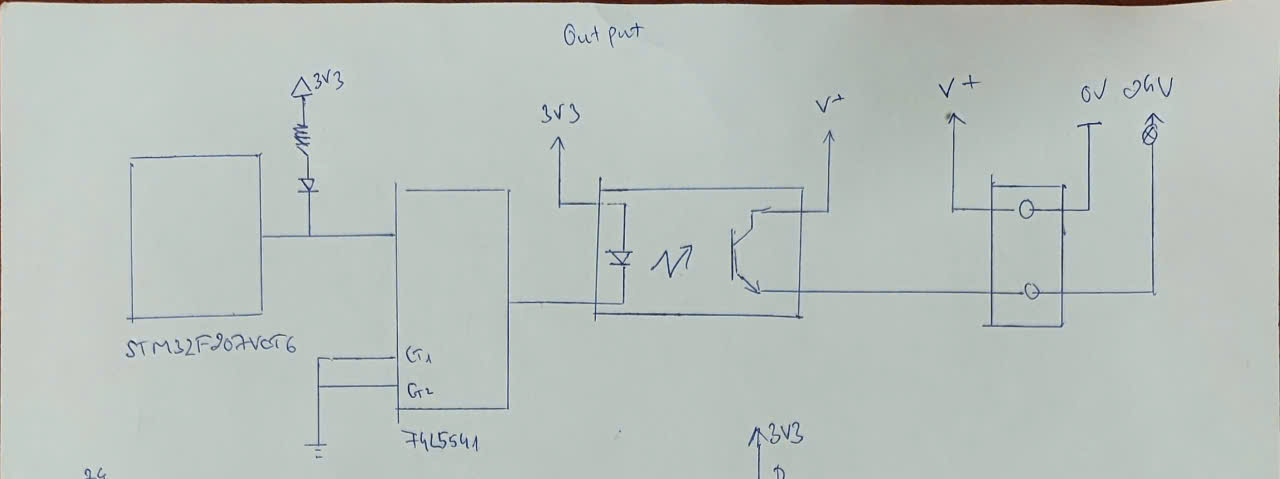
|  |  |
| --- | --- |
| TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TPHCM  **KHOA CƠ KHÍ CHẾ TẠO MÁY**  **BỘ MÔN CƠ ĐIỆN TỬ**  **Họ và tên:** Võ Hồng Quân  **Mã số SV:** 22134012 | **BÁO CÁO**  **THỰC TẬP LẬP TRÌNH VI XỬ LÝ**  Ngày:  Tuần học: 01  Nội dung: **GPIO và State Machine** |

1. **Khảo sát sơ đồ nguyên lý:**

Sinh viên vẽ sơ đồ nguyên lý khối **Digital Input** và giải thích



Sinh viên vẽ sơ đồ nguyên lý khối **Digital Output** và giải thích



1. **Bảng kết nối input – output**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên phần cứng | Vị trí chân vi điều khiển tương ứng | Mức tín hiệu ở trạng thái tích cực |
| 1 | Đèn D1 | PA8 | T Thấp |
| 2 | Đèn D2 | PC9 | Thấp |
| 3 | Nút nhấn P1 | PE13 | t Thấp |
| 4 | Nút nhấn P2 | PE12 | Thấp |
| 5 | Nút nhấn P3 | PE10 | Thấp |
| 6 | Nút nhấn P4 | PE8 | T Thấp |
| 7 | Công tắt gạt vị trí 1 | PE7 |  |
| 8 | Công tắt gạt vị trí 2 | PB2 |  |
| 9 | Công tắt gạt vị trí 3 | PC4 |  |

1. **Lập trình**
   1. **Cài đặt phần mềm và thư viện:**

* Để lập trình STM32 thì người sử dụng cần cài đặt các phần mềm sau:

|  |  |  |
| --- | --- | --- |
| STT | Phần mềm | Chức năng |
| 1 | STM32 CubeMX | Tạo dự án lập trình bằng giao diện GUI  Thư viện HAL |
| 2 | Keil C for ARM | Môi trường soạn thảo mã nguồn chương trình và biên dịch C cho vi điều khiển ARM. |
| 3 | STM32 ST-LINK Utility | Tải chương trình đã biên dịch (.hex) từ máy tính xuống vi điều khiển |

* 1. **Tạo thư viện myGPIO**
* Viết thư. viện myGPIO.h và myGPIO c với các hàm và macro sau đây:
  + Định nghĩa các trạng thái đèn D\_ON, D\_OFF
  + Định nghĩa các trạng thái nút nhấn BUTTON\_ACTIVE, BUTTON\_INACTIVE
  + Định nghĩa các trạng thái công tắc gạt SW1\_ACTIVE, SW2\_ACTIVE, SW3\_ACTIVE
  + Hàm void **Light\_cmd**(uint8\_t **d**, uint8\_t **status**) để bật tắt đèn D1, D2:
    - Light\_cmd(D1, D\_ON) : Bật đèn D1 sáng
    - Light\_cmd(D1, D\_OFF) : Tắt đèn D1
    - Light\_cmd(D2, D\_ON) : Bật đèn D2 sáng
    - Light\_cmd(D2, D\_OFF) : Tắt đèn D2
  + Hàm GPIO\_PinState **PushButton\_read**(uint8\_t **p**) để đọc trạng thái của nút nhấn p. Ví dụ:
    - **PushButton\_read** (P1) : Đọc về trạng thái của nút nhấn P1. Nếu đang được nhấn thì trả về giá trị BUTTON\_ACTIVE; nếu chưa được nhấn thì trả về giá trị BUTTON\_INACTIVE.
  + Hàm uint8\_t **Switch\_read**(void) để đọc vị trí của switch. Ví dụ:
    - **Switch\_read** (void) : Đọc về trạng thái của Switch. Nếu SW đang gạt về vị trí 1 (bên trái) thì hàm trả về SW1\_ACTIVE. Nếu SW đang gạt về vị trí 2 (giữa) thì hàm trả về SW2\_ACTIVE. Nếu SW đang gạt về vị trí 3 (bên phải) thì hàm trả về SW3\_ACTIVE.

Mã nguồn chương trình (chỉ copy nội dung đã thêm hoặc chỉnh sửa từ code dự án)

* myGPIO.h:

#ifndef MYGPIO\_H

#define MYGPIO\_H

#define P\_INACTIVE GPIO\_PIN\_SET

#define D1 1

#define D2 2

#define SW\_ON GPIO\_PIN\_SET

#define SW\_INACTIVE GPIO\_PIN\_RESET

#define P1 1

#define P2 2

#define P3 3

#define P4 4

#define SW1\_ACTIVE 1

#define SW2\_ACTIVE 2

#define SW3\_ACTIVE 3

extern GPIO\_PinState P1\_STATUS;

extern GPIO\_PinState P2\_STATUS;

extern GPIO\_PinState P3\_STATUS;

extern GPIO\_PinState P4\_STATUS;

void Light\_cmd(uint8\_t d, uint8\_t status);

uint8\_t PushButton\_read(uint8\_t p);

uint8\_t Switch\_read(void);

void Baitap3\_3(uint8\_t P1\_STATUS, uint8\_t P2\_STATUS);

#endif

* myGPIO.c:

#include "main.h"

#include "myGPIO.h"

GPIO\_PinState P1\_STATUS;

GPIO\_PinState P2\_STATUS;

GPIO\_PinState P3\_STATUS;

GPIO\_PinState P4\_State;

void Light\_cmd(uint8\_t d, uint8\_t status){

switch(d){

case D1:

HAL\_GPIO\_WritePin(D1\_GPIO\_Port, D1\_Pin, status);

break;

case D2:

HAL\_GPIO\_WritePin(D2\_GPIO\_Port, D2\_Pin, status);

break;

default:

return;

}

}

uint8\_t PushButton\_read(uint8\_t p) {

switch (p) {

case P1:

return HAL\_GPIO\_ReadPin(P1\_GPIO\_Port, P1\_Pin);

case P2:

return HAL\_GPIO\_ReadPin(P2\_GPIO\_Port, P2\_Pin);

case P3:

return HAL\_GPIO\_ReadPin(P3\_GPIO\_Port, P3\_Pin);

case P4:

return HAL\_GPIO\_ReadPin(P4\_GPIO\_Port, P4\_Pin);

default:

return P\_INACTIVE;

}

}

uint8\_t Switch\_read(void) {

if (HAL\_GPIO\_ReadPin(SW1\_1\_GPIO\_Port, SW1\_1\_Pin) == SW\_ON) {

return SW1\_ACTIVE;

} else if (HAL\_GPIO\_ReadPin(SW1\_2\_GPIO\_Port, SW1\_2\_Pin) == SW\_ON) {

return SW2\_ACTIVE;

} else if (HAL\_GPIO\_ReadPin(SW1\_3\_GPIO\_Port, SW1\_3\_Pin) == SW\_ON) {

return SW3\_ACTIVE;

} else {

return SW\_INACTIVE;

}

}

* 1. **Lập trình ứng dụng**
* Khi mới mở điện: Đèn D1, D2 tắt;
* Mỗi khi nhấn nút **P1** đèn D1 đảo trạng thái (vừa nhấn vào đã tác động):
  + Trạng thái 1: Đèn sáng; Trạng thái 2: Đèn tắt. Ví dụ: Nhấn P1 lần 1 thì đèn D1 sáng; nhấn P1 lần 2 thì đèn D1 tắt; nhấn P1 lần 3 thì đèn D1 sáng.v.v…
* Mỗi khi nhấn nút **P2** thì đèn D2 đảo trạng thái (nhấn vào thả tay ra mới tác động):
  + Trạng thái 1: Đèn chớp tắt chu kỳ 2s; Trạng thái 2: Đèn tắt.

Mã nguồn chương trình (chỉ copy nội dung đã thêm hoặc chỉnh sửa từ code dự án)

uint8\_t stateP1 = 0;

uint8\_t stateP2 = 0;

int DelayCnt = 0;

void Baitap3\_3(uint8\_t P1\_STATUS, uint8\_t P2\_STATUS) {

// P1 state

if (stateP1 == 0) {

Light\_cmd(D1, D\_OFF);

stateP1 = 1;

} else if (stateP1 == 1) {

Light\_cmd(D1, D\_OFF);

if (P1\_STATUS == P\_ACTIVE) {

stateP1 = 2;

}

} else if (stateP1 == 2) {

Light\_cmd(D1, D\_ON);

if (P1\_STATUS == P\_INACTIVE) {

stateP1 = 3;

}

} else if (stateP1 == 3) {

Light\_cmd(D1, D\_ON);

if (P1\_STATUS == P\_ACTIVE) {

stateP1 = 4;

}

} else if (stateP1 == 4) {

Light\_cmd(D1, D\_OFF);

if (P1\_STATUS == P\_INACTIVE) {

stateP1 = 0;

}

}

// P2 state

uint16\_t valueCnt = 0;

if (stateP2 == 0) {

Light\_cmd(D2, D\_OFF);

if (P2\_STATUS == P\_ACTIVE) {

stateP2 = 1;

}

} else if (stateP2 == 1) {

Light\_cmd(D2, D\_OFF);

if (P2\_STATUS == P\_INACTIVE) {

stateP2 = 2;

}

// Blink

} else if (stateP2 == 2) {

if (DelayCnt < 1000) {

Light\_cmd(D2, D\_ON);

HAL\_Delay(20);

DelayCnt+=20;

}

else if (DelayCnt < 2000) {

Light\_cmd(D2, D\_OFF);

HAL\_Delay(20);

DelayCnt+=20;

}

else {

DelayCnt = 0;

}

if (P2\_STATUS == P\_ACTIVE) {

stateP2 = 3;

}

} else if (stateP2 == 3) {

if (DelayCnt < 1000) {

Light\_cmd(D2, D\_ON);

HAL\_Delay(20);

DelayCnt+=20;

}

else if (DelayCnt < 2000) {

Light\_cmd(D2, D\_OFF);

HAL\_Delay(20);

DelayCnt+=20;

}

else {

DelayCnt = 0;

}

if (P2\_STATUS == P\_INACTIVE) {

stateP2 = 4;

}

} else if (stateP2 == 4) {

Light\_cmd(D2, D\_OFF);

if (P2\_STATUS == P\_ACTIVE) {

stateP2 = 0;

}

}

}

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

while (1)

{

P1\_STATUS = PushButton\_read(P1);

P2\_STATUS = PushButton\_read(P2);

Baitap3\_3(P1\_STATUS, P2\_STATUS);

}

}

* 1. **Lập trình ứng dụng**
* Khi mới mở điện: Đèn D1, D2 tắt;
* Công tắt gạt chuyển trạng thái của đèn D1 như sau:
  + SW1\_ACTIVE: Trạng thái 1 đèn D1 tắt.
  + SW2\_ACTIVE: Trạng thái 2 đèn D1 sáng tắt chu kỳ 2s.
  + SW3\_ACTIVE: Trạng thái 3 đèn D1 sáng liên tục.
* Mỗi khi nhấn nút nhấn P1, đèn D2 chuyển 3 trạng thái (lặp lại):
  + Nhấn lần 1: Trạng thái 1 đèn D2 tắt.
  + Nhấn lần 2: Trạng thái 2 đèn D2 sáng tắt chu kỳ 2s.
  + Nhấn lần 3: Trạng thái 3 đèn D2 sáng liên tục.
* Mã nguồn chương trình (chỉ copy nội dung đã thêm hoặc chỉnh sửa từ code dự án)

void Baitap3\_4(uint16\_t SW\_STATUS) {

// State-machine logic

switch (stateSW)

{

case 0:

Light\_cmd(D1, D\_OFF);

Light\_cmd(D2, D\_OFF);

if (SW\_STATUS == SW1\_ACTIVE) stateSW = SW1\_ACTIVE;

break;

case SW1\_ACTIVE:

Light\_cmd(D1, D\_OFF);

if (SW\_STATUS == SW2\_ACTIVE) {

stateSW = SW2\_ACTIVE;

DelayCnt = 0; // Reset delay counter when entering blinking state

}

break;

case SW2\_ACTIVE:

if (DelayCnt < 1000) {

Light\_cmd(D1, D\_ON);

HAL\_Delay(20);

DelayCnt += 20;

}

else if (DelayCnt < 2000) {

Light\_cmd(D1, D\_OFF);

HAL\_Delay(20);

DelayCnt += 20;

}

else {

DelayCnt = 0; // Reset the blink counter only after one full ON/OFF cycle

}

if (SW\_STATUS == SW3\_ACTIVE) {

stateSW = SW3\_ACTIVE;

}

break;

case SW3\_ACTIVE:

Light\_cmd(D1, D\_ON);

if (SW\_STATUS == SW1\_ACTIVE) {

stateSW = SW1\_ACTIVE; // Reset to initial active state

}

break;

default:

stateSW = 0; // Fallback safety case

break;

}

}

int main(void)

{

// Initialize state variables

stateSW = 0;

DelayCnt = 0;

while (1)

{

// Read the status of push buttons and switch using available functions

P1\_STATUS = PushButton\_read(P1);

P2\_STATUS = PushButton\_read(P2);

uint8\_t SW\_STATUS = Switch\_read();

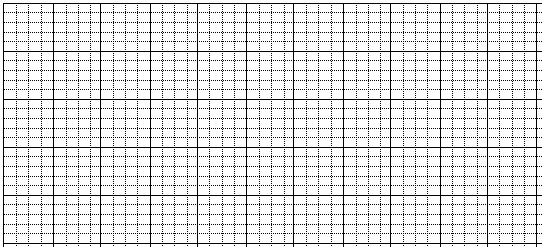
Baitap3\_4(SW\_STATUS);

}

}

* 1. **Lập trình ứng dụng**
* Khi mới mở điện: Đèn D1, D2 tắt;
* Mỗi khi nhấn nút nhấn P1 hoặc P2, đèn D1 chuyển 3 trạng thái (lặp lại):
  + Nhấn lần 1: Trạng thái 1 đèn D1 tắt.
  + Nhấn lần 2: Trạng thái 2 đèn D1 sáng tắt chu kỳ 2s.
  + Nhấn lần 3: Trạng thái 3 đèn D1 sáng liên tục.

Mã nguồn chương trình (chỉ copy nội dung đã thêm hoặc chỉnh sửa từ code dự án)



* 1. **Lập trình ứng dụng**
* Khi mới mở điện: Đèn D1, D2 tắt;
* Mỗi khi nhấn nút nhấn P1 đèn D1, D2 chuyển 2 trạng thái (lặp lại):
  + Trạng thái 1 đèn D1, D2 tắt.
  + Trạng thái 2 đèn D1 sáng tắt chu kỳ 2s. Đèn D2 tắt.
* Mỗi khi nhấn nút nhấn P2 đèn D1, D2 chuyển 2 trạng thái (lặp lại):
  + Trạng thái 1 đèn D1, D2 tắt.
  + Trạng thái 2 đèn D1 tắt. Đèn D2 sáng tắt chu kỳ 2s.
* Mỗi khi nhấn nút nhấn P3 đèn D1, D2 chuyển 2 trạng thái (lặp lại):
  + Trạng thái 1 đèn D1, D2 tắt.
  + Trạng thái 2 đèn D1, D2 sáng tắt so le chu kỳ 2s.
* Mỗi khi nhấn nút nhấn P4 đèn D1, D2 chuyển 2 trạng thái (lặp lại):
  + Trạng thái 1 đèn D1, D2 tắt.
  + Trạng thái 2 đèn D1, D2 sáng tắt cùng nhau chu kỳ 2s.

Mã nguồn chương trình (chỉ copy nội dung đã thêm hoặc chỉnh sửa từ code dự án)

