TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC QUỐC GIA TP.HCM
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

VI XỬ LÝ - VI ĐIỀU KHIỂN (TN) (CO3010)

Lab 1

LED Animations

GVHD: Lê Trọng Nhân
Phạm Công Thái

Sinh viên: Huỳnh Bùi Ngọc Khoa - 2211589

TP.Hồ Chí Minh, 09/2024

Contents

1 Exercise 1

Report 1: Depict the schematic from Proteus simulation in this report. The caption of the figure is a downloadable link to the Proteus project file (e.g. a github link).
Note: For this Report 1, I will combine the schematic in exercise 1 and exercise 2 in one figure below.

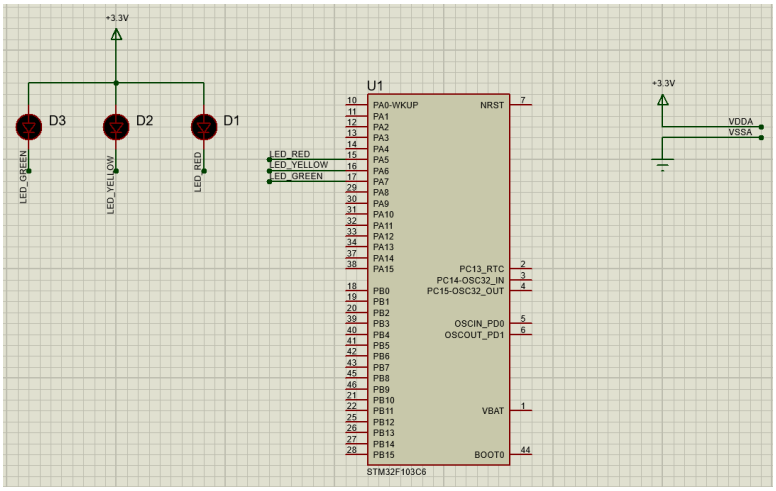

Figure 1: Link to the figure

Report 2: Present the source code in the infinite loop while of your project. If a user-defined functions is used, it is required to present in this part. A brief description can be added for this function (e.g. using comments). A template to present your source code is presented bellow.

```
int cnt = 0;
while (1) {
    switch (cnt) {
    case 4:
        cnt = 0;
    case 0:
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
        break;
    case 2:
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
        HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, RESET);
        break;
    default:
        break;
    }
    cnt++;
    HAL_Delay(1000);
}
```

Listing 1: Source code for Exercise 2

2 Exercise 2

Report 1: Present the schematic.
Checkout Report 1 of Exercise 1.

Report 2: Present the source code in while.

```
int cnt = 10;
void set_LEDS(GPIO_PinState redstate, GPIO_PinState yellowstate, GPIO_PinState
    greenstate) {
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, redstate);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, yellowstate);
    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, greenstate);
}
while (1) {
    switch (cnt) {
    case 10:
        set_LEDS(1, 1, 0);
        break;
    case 7:
        set_LEDS(1, 0, 1);
        break;
    case 5:
        set_LEDS(0, 1, 1);
        break;
    case 1:
        cnt = 11;
    default:
        break;
    }
    cnt--;
    HAL_Delay(1000);
}
```

Listing 2: Source code for Exercise 2

3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure bellow.
Note: The below schematic is used for Exercise 3, 4 and 5.

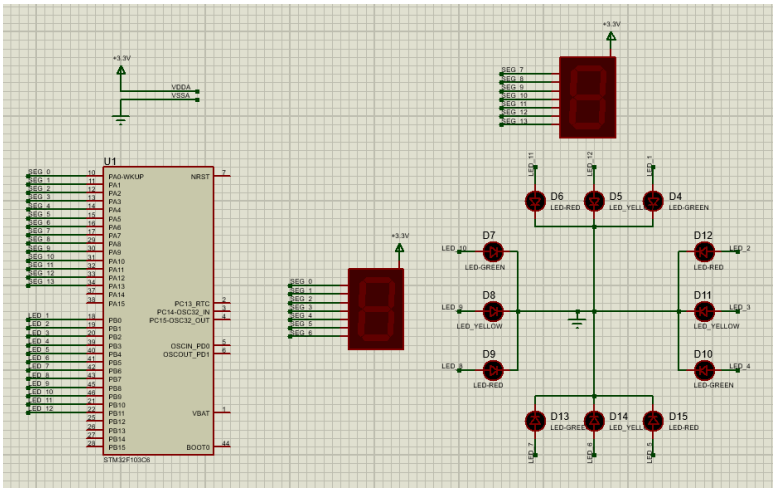

Figure 2: Link to the figure

```
GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
      LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
      LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
      LED_11_GPIO_Port, LED_12_GPIO_Port };
uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
        LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
        LED_11_Pin, LED_12_Pin };
GPIO_PinState LEDS_state[4][12] = { { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
        { 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1 }, { 0, 0, 0, 1, 1, 0, 0, 0, 0,
                1, 1, 0 }, { 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0 } };
void set_LEDS(const GPIO_PinState *state) {
    for (int i = 0; i < 12; i++) {
        HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], state[i]);
    }
}
int cnt = 10;
while (1) {
    switch (cnt) {
    case 10:
        set_LEDS(LEDS_state[0]);
        break;
    case 7:
        set_LEDS(LEDS_state[1]);
        break;
    case 5:
        set_LEDS(LEDS_state[2]);
        break;
    case 2:
        set_LEDS(LEDS_state[3]);
        break;
    case 1:
        cnt = 11;
```

```
33          break;
34      default:
35          break;
36      }
37      cnt--;
38      HAL_Delay(1000);
39  }
```

Listing 3: Source code for Exercise 3

4 Exercise 4

Report 1: Present the schematic.
Checkout the schematic of Exercise 3

Report 2: Present the source code for display7SEG function.

```
GPIO_TypeDef *SEG_PORTS[12] = { SEG_0_GPIO_Port, SEG_1_GPIO_Port,
        SEG_2_GPIO_Port, SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
        SEG_6_GPIO_Port };
uint16_t SEG_PINS[12] = { SEG_0_Pin, SEG_1_Pin, SEG_2_Pin, SEG_3_Pin,
        SEG_4_Pin, SEG_5_Pin, SEG_6_Pin };
GPIO_PinState LEDS_state[10][7] = { { 0, 0, 0, 0, 0, 0, 1 }, { 1, 0, 0, 1,
        1, 1, 1 }, { 0, 0, 1, 0, 0, 1, 0 }, { 0, 0, 0, 0, 1, 1, 0 }, { 1, 0,
        0, 1, 1, 0, 0 }, { 0, 1, 0, 0, 1, 0, 0 }, { 0, 1, 0, 0, 0, 0, 0 }, {
        0, 0, 0, 1, 1, 1, 1 }, { 0, 0, 0, 0, 0, 0, 0 }, { 0, 0, 0, 0, 1, 0,
        0 } };
void set_LEDS(GPIO_PinState *L_LEDS_state) {
    for (int i = 0; i < 7; i++) {
        HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_state[i]);
    }
}
void display7SEG(int number) {
    switch (number) {
    case 0:
        set_LEDS(LEDS_state[0]);
        break;
    case 1:
        set_LEDS(LEDS_state[1]);
        break;
    case 2:
        set_LEDS(LEDS_state[2]);
        break;
    case 3:
        set_LEDS(LEDS_state[3]);
        break;
    case 4:
        set_LEDS(LEDS_state[4]);
        break;
    case 5:
        set_LEDS(LEDS_state[5]);
        break;
    case 6:
        set_LEDS(LEDS_state[6]);
        break;
    case 7:
        set_LEDS(LEDS_state[7]);
        break;
    case 8:
        set_LEDS(LEDS_state[8]);
        break;
    case 9:
        set_LEDS(LEDS_state[9]);
        break;
    default:
        break;
    }
}
int counter = 0;
while (1) {
    if (counter >= 10)
```

```
55          counter = 0;
56      display7SEG(counter++);
57      HAL_Delay(1000);
58  }
```

Listing 4: Source code for Exercise 4

5 Exercise 5

```
GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
LED_11_GPIO_Port, LED_12_GPIO_Port };
uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
LED_11_Pin, LED_12_Pin };
GPIO_PinState LEDS_state[4][12] = { { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
        { 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1 }, { 0, 0, 0, 1, 1, 0, 0, 0, 0,
                1, 1, 0 }, { 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0 } };
void set_LEDS(const GPIO_PinState *state) {
    for (int i = 0; i < 12; i++) {
        HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], state[i]);
    }
}
GPIO_TypeDef *SEG_PORTS[14] = { SEG_0_GPIO_Port, SEG_1_GPIO_Port,
SEG_2_GPIO_Port, SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
SEG_6_GPIO_Port, SEG_7_GPIO_Port, SEG_8_GPIO_Port, SEG_9_GPIO_Port,
SEG_10_GPIO_Port, SEG_11_GPIO_Port, SEG_12_GPIO_Port,
SEG_13_GPIO_Port };
uint16_t SEG_PINS[14] = { SEG_0_Pin, SEG_1_Pin, SEG_2_Pin, SEG_3_Pin,
SEG_4_Pin, SEG_5_Pin, SEG_6_Pin, SEG_7_Pin, SEG_8_Pin, SEG_9_Pin,
SEG_10_Pin, SEG_11_Pin, SEG_12_Pin, SEG_13_Pin };
GPIO_PinState LEDS_7SEG_state[6][7] = { { 0, 0, 0, 0, 0, 0, 1 }, { 1, 0, 0,
        1, 1, 1, 1 }, { 0, 0, 1, 0, 0, 1, 0 }, { 0, 0, 0, 0, 1, 1, 0 }, { 1,
        0, 0, 1, 1, 0, 0 }, { 0, 1, 0, 0, 1, 0, 0 } };
void set_7SEG_X(const GPIO_PinState *L_LEDS_X_state) {
    for (int i = 0; i < 7; i++) {
        HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_X_state[i]);
    }
}
void set_7SEG_Y(const GPIO_PinState *L_LEDS_Y_state) {
    for (int i = 0; i < 7; i++) {
        HAL_GPIO_WritePin(SEG_PORTS[i + 7], SEG_PINS[i + 7],
                L_LEDS_Y_state[i]);
    }
}
int cnt = 10;
while (1) {
    switch (cnt) {
    case 10:
        set_LEDS(LEDS_state[0]);
        set_7SEG_Y(LEDS_7SEG_state[3]);
        set_7SEG_X(LEDS_7SEG_state[5]);
        break;
    case 9:
        set_7SEG_Y(LEDS_7SEG_state[2]);
        set_7SEG_X(LEDS_7SEG_state[4]);
        break;
    case 8:
        set_7SEG_Y(LEDS_7SEG_state[1]);
        set_7SEG_X(LEDS_7SEG_state[3]);
        break;
    case 7:
        set_LEDS(LEDS_state[1]);
        set_7SEG_Y(LEDS_7SEG_state[2]);
        set_7SEG_X(LEDS_7SEG_state[2]);
        break;
    case 6:
```

```
60          set_7SEG_Y(LEDS_7SEG_state[1]);
61          set_7SEG_X(LEDS_7SEG_state[1]);
62          break;
63      case 5:
64          set_LEDS(LEDS_state[2]);
65          set_7SEG_Y(LEDS_7SEG_state[5]);
66          set_7SEG_X(LEDS_7SEG_state[3]);
67          break;
68      case 4:
69          set_7SEG_Y(LEDS_7SEG_state[4]);
70          set_7SEG_X(LEDS_7SEG_state[2]);
71          break;
72      case 3:
73          set_7SEG_Y(LEDS_7SEG_state[3]);
74          set_7SEG_X(LEDS_7SEG_state[1]);
75          break;
76      case 2:
77          set_LEDS(LEDS_state[3]);
78          set_7SEG_Y(LEDS_7SEG_state[2]);
79          set_7SEG_X(LEDS_7SEG_state[2]);
80          break;
81      case 1:
82          set_7SEG_Y(LEDS_7SEG_state[1]);
83          set_7SEG_X(LEDS_7SEG_state[1]);
84          cnt = 11;
85          break;
86      default:
87          break;
88      }
89      cnt--;
90      HAL_Delay(1000);
91  }
```

Listing 5: Source code for Exercise 5

6 Exercise 6

Report 1: Present the schematic.

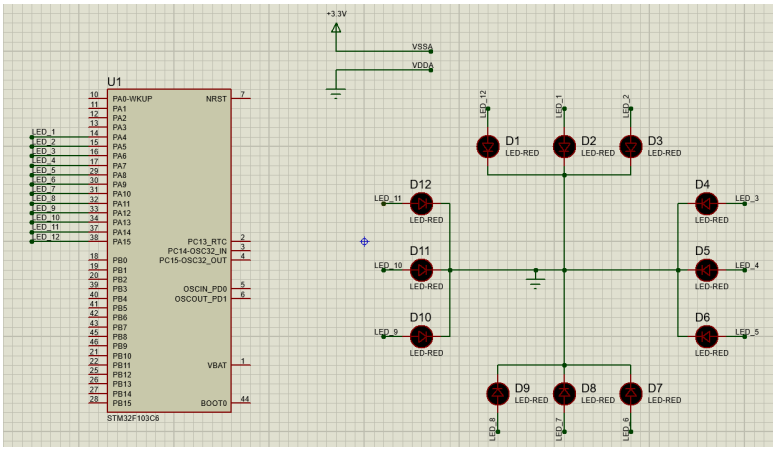

Figure 3: Link to the figure

Report 2: Implement a simple program to test the connection of every single LED. This testing program should turn every LED in a sequence.

```
GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
        LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
        LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
        LED_11_GPIO_Port, LED_12_GPIO_Port };
uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
        LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
        LED_11_Pin, LED_12_Pin };
int cnt = 0;
while (1) {
    if (cnt >= 12) {
        cnt = 0;
    }
    HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 1);
    HAL_Delay(1000);
    HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 0);
    cnt++;
}
```

Listing 6: Source code for Exercise 6

7 Exercise 7

```
void clearAllClock() {
    for (int i = 0; i < 12; i++) {
        HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], 0);
    }
}
```

Listing 7: Source code for Exercise 7

8 Exercise 8

```
void setNumberOnClock(int num) {
    HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 1);
}
```

Listing 8: Source code for Exercise 8

9 Exercise 9

```
void clearNumberOnClock(int num) {
    HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 0);
}
```

Listing 9: Source code for Exercise 9

10 Exercise 10

```
GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
LED_11_GPIO_Port, LED_12_GPIO_Port };
uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
LED_11_Pin, LED_12_Pin };
void clearAllClock() {
    for (int i = 0; i < 12; i++) {
        HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], 0);
    }
}
void setNumberOnClock(int num) {
    HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 1);
}
int cnt_hour = 0;
int cnt_min = 0;
int cnt_sec = 0;
while (1) {
    setNumberOnClock(cnt_hour);
    setNumberOnClock(cnt_min / 5);
    setNumberOnClock(cnt_sec / 5);
    HAL_Delay(1000);
    cnt_sec++;
    if (cnt_sec == 60) {
        cnt_sec = 0;
        cnt_min++;
    }
    if (cnt_min == 60) {
        cnt_min = 0;
        cnt_hour++;
    }
    if (cnt_hour == 12) {
        cnt_hour = 0;
    }
    clearAllClock();
}
```

Listing 10: Source code for Exercise 10