TRƯỜNG ĐẠI HỌC BÁCH KHOA – TP. HỒ CHÍ MINH

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

****

Lab 1 - LED Animation

Microcontroller - Microprocessor (Lab)

COURSE ID: CO3009 - HK251

Name: Huỳnh Văn Thanh - Student ID: 2353083

Supervising Lecturer: Dr. VÕ TUẤN BÌNH

TP. Hồ Chí Minh – 2024

1 Exercise

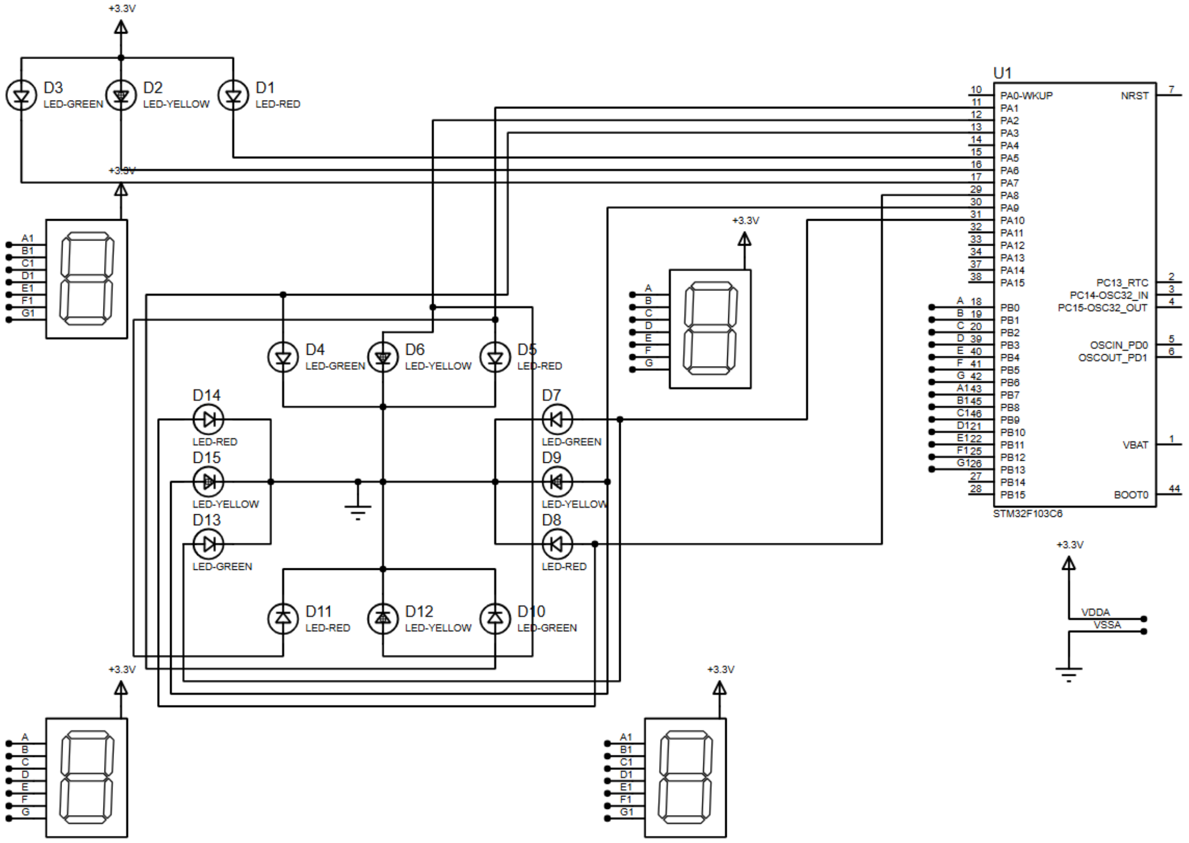
The GitHub link for the lab schematics is at here or in this link: https://github.com/TheodoreHenryAI/Lab_MCU-MPU-CO3009_Sem-251

The default while(1) code for most of the exercise is:

```
while (1) {
    // THE FUNCTION INPUT INSERTED HERE
    HAL_Delay(1000);
}
```

1.1 Exercise 1

1.1.1 Report 1:



1.1.2 Report 2:

```
int main(void) {
    // Initial state
    // RED ON, YELLOW OFF
    turnOn(RED);
    turnOff(YELLOW);

    // Infinite loop
    while (true) {
        // --- PHASE 1 ---
        // RED ON, YELLOW OFF (2s)
        turnOn(RED);
        turnOff(YELLOW);
        delay(2000);

        // --- PHASE 2 ---
        // RED OFF, YELLOW ON (2s)
        turnOff(RED);
        turnOn(YELLOW);
        delay(2000);
    }
}
```

1.2 Exercise 2

1.2.1 Report 1:

Can be found at Exercise 1/Report 1

1.2.2 Report 2:

```c
int main(void) {
    // Infinite loop
    while (true) {
        // --- RED phase (5s) ---
        turnOn(RED);
        turnOff(YELLOW);
        turnOff(GREEN);
        delay(5000);

        // --- YELLOW phase (2s) ---
        turnOff(RED);
        turnOn(YELLOW);
        turnOff(GREEN);
        delay(2000);

        // --- GREEN phase (3s) ---
        turnOff(RED);
        turnOff(YELLOW);
        turnOn(GREEN);
        delay(3000);
    }
}
```

1.3 Exercise 3

1.3.1 Report 1:

Can be found at Exercise 1/Report 1

1.3.2 Report 2:

```c
void setLight(Road road, Color color) {
    turnOff(road.RED);
    turnOff(road.YELLOW);
    turnOff(road.GREEN);

    if (color == RED)    turnOn(road.RED);
    if (color == YELLOW) turnOn(road.YELLOW);
    if (color == GREEN)  turnOn(road.GREEN);
}

int main(void) {
    while (true) {
        // --- Phase 1: Road1 GREEN, Road2 RED (3s) ---
        setLight(Road1, GREEN);
        setLight(Road2, RED);
        delay(3000);

        // --- Phase 2: Road1 YELLOW, Road2 RED (2s) ---
        setLight(Road1, YELLOW);
        setLight(Road2, RED);
        delay(2000);

        // --- Phase 3: Road1 RED, Road2 GREEN (3s) ---
        setLight(Road1, RED);
        setLight(Road2, GREEN);
        delay(3000);

        // --- Phase 4: Road1 RED, Road2 YELLOW (2s) ---
        setLight(Road1, RED);
        setLight(Road2, YELLOW);
        delay(2000);

        // loop back to Road1 GREEN again
    }
}
```

1.4 Exercise 4

1.4.1 Report 1:

Can be found at Exercise 1/Report 1

1.4.2 Report 2:

```
void setLight(Road road, Color color) {
    turnOff(road.RED);
    turnOff(road.YELLOW);
    turnOff(road.GREEN);

    if (color == RED)    turnOn(road.RED);
    if (color == YELLOW) turnOn(road.YELLOW);
    if (color == GREEN)  turnOn(road.GREEN);
}

void display7SEG(int num) {
    // light up segments to show digit ‘num‘ (0-9)
}

int main(void) {
    int counter = 0;   // 7-seg counter (0-9, wraps)

    while (true) {
        // --- Phase 1: Road1 GREEN, Road2 RED ---
        setLight(Road1, GREEN);
        setLight(Road2, RED);
        for (i = 0; i < GREEN_SEC; i++) {
            display7SEG(counter);
            counter = (counter + 1) % 10;
            delay(1 second);
        }

        // --- Phase 2: Road1 YELLOW, Road2 RED ---
        setLight(Road1, YELLOW);
        setLight(Road2, RED);
        for (i = 0; i < YELLOW_SEC; i++) {
            display7SEG(counter);
            counter = (counter + 1) % 10;
            delay(1 second);
        }

        // --- Phase 3: Road1 RED, Road2 GREEN ---
        setLight(Road1, RED);
        setLight(Road2, GREEN);
        for (i = 0; i < GREEN_SEC; i++) {
            display7SEG(counter);
            counter = (counter + 1) % 10;
            delay(1 second);
        }

        // --- Phase 4: Road1 RED, Road2 YELLOW ---
        setLight(Road1, RED);
        setLight(Road2, YELLOW);
        for (i = 0; i < YELLOW_SEC; i++) {
            display7SEG(counter);
            counter = (counter + 1) % 10;
            delay(1 second);
        }

        // loop back to Phase 1 again
    }
}
```

1.5 Exercise 5

1.5.1 Report 1:

Can be found at Exercise 1/Report 1

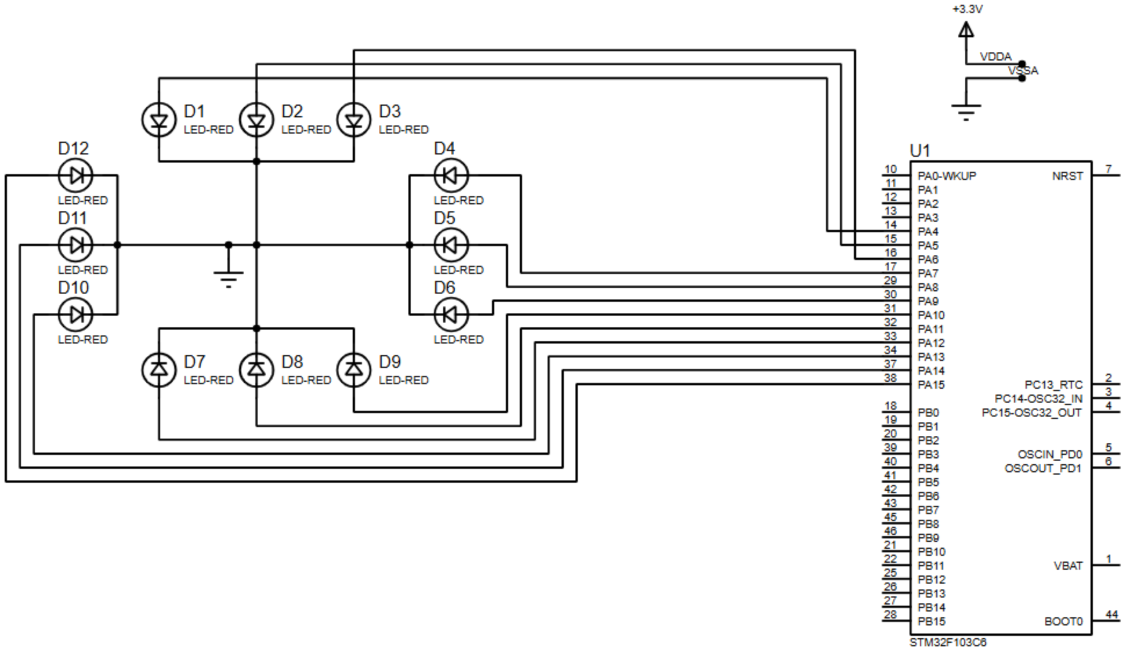
1.5.2 Report 2:

```
void setTrafficLight(Road road, State state) {
    turnOffAllLights(road);
    if (state == RED)    turnOn(road.RED);
    if (state == YELLOW) turnOn(road.YELLOW);
    if (state == GREEN)  turnOn(road.GREEN);
}

void display7SEG_R1(int num) { showDigitOnRoad1(num); }
void display7SEG_R2(int num) { showDigitOnRoad2(num); }

```

```
int main(void) {
    // Initial states
    r1_state = GREEN;    // Road1 starts green
    r2_state = RED;      // Road2 starts red
    r1_time  = 3;        // Road1 green lasts 3s
    r2_time  = 6;        // Road2 red lasts 6s

    setTrafficLight(Road1, r1_state);
    setTrafficLight(Road2, r2_state);

    while (true) {
        // --- update displays ---
        display7SEG_R1(r1_time);
        display7SEG_R2(r2_time);

        wait(1 second);

        // --- decrease timers ---
        r1_time--;
        r2_time--;

        // --- Road1 transitions ---
        if (r1_time < 0) {
            if (r1_state == GREEN)    { r1_state = YELLOW; r1_time = 2; }
            else if (r1_state == YELLOW) { r1_state = RED;    r1_time = 6; }
            else if (r1_state == RED)    { r1_state = GREEN;  r1_time = 3; }
            setTrafficLight(Road1, r1_state);
        }

        // --- Road2 transitions ---
        if (r2_time < 0) {
            if (r2_state == RED)      { r2_state = GREEN;  r2_time = 3; }
            else if (r2_state == GREEN)  { r2_state = YELLOW; r2_time = 2; }
            else if (r2_state == YELLOW) { r2_state = RED;    r2_time = 6; }
            setTrafficLight(Road2, r2_state);
        }
    }
}
```

1.6 Exercise 6

1.6.1 Report 1:



1.6.2 Report 2:

```
// Test: turn ON each LED in sequence, then OFF
for (i = 0; i < 12; i++) {
    turnOnLED(i);
    wait(200 ms);
    turnOffLED(i);
}
```

1.7 Exercise 7

1.7.1 Report 1:

Can be found at Exercise 6/Report 1

1.7.2 Report 2:

```
// Function: turn OFF all 12 LEDs
void clearAllClock() {
    for (i = 0; i < 12; i++) {
        turnOffLED(i);
    }
}
```

1.8 Exercise 8

1.8.1 Report 1:

Can be found at Exercise 6/Report 1

1.8.2 Report 2:

```
// Function: light exactly one LED, others OFF
void setNumberOnClock(int num) {
    clearAllClock();
    if (0 <= num < 12) {
        turnOnLED(num);
    }
}
```

1.9 Exercise 9

1.9.1 Report 1:

Can be found at Exercise 6/Report 1

1.9.2 Report 2:

```
// Function: turn OFF one specific LED (leave others unchanged)
void clearNumberOnClock(int num) {
    if (0 <= num < 12) {
        turnOffLED(num);
    }
}
```

1.10 Exercise 10

1.10.1 Report 1:

Can be found at Exercise 6/Report 1

1.10.2 Report 2:

```
// Initialize positions
second_pos = 0;
minute_pos = 0;
hour_pos   = 0;

while (true) {
    clearAllClock();

    // Light the 3 LEDs for hour, minute, second
    turnOnLED(hour_pos);
    turnOnLED(minute_pos);
    turnOnLED(second_pos);

    wait(1 second);

    // Advance second hand
    second_pos = (second_pos + 1) % 12;

    // If seconds wrapped, advance minute
    if (second_pos == 0) {
        minute_pos = (minute_pos + 1) % 12;

        // If minutes wrapped, advance hour
        if (minute_pos == 0) {
            hour_pos = (hour_pos + 1) % 12;
        }
    }
}
```