

Федеральное государственное автономное учреждение высшего
образования
Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.04 – Программная инженерия
Дисциплина: «Проектирование вычислительных систем»

Лабораторная работа №1
Вариант 11

Группа: Р34111

Выполнили:
Емельянов Д С
Иванов М С

Преподаватель:
Пинкевич Василий Юрьевич

Санкт-Петербург
2024 г.

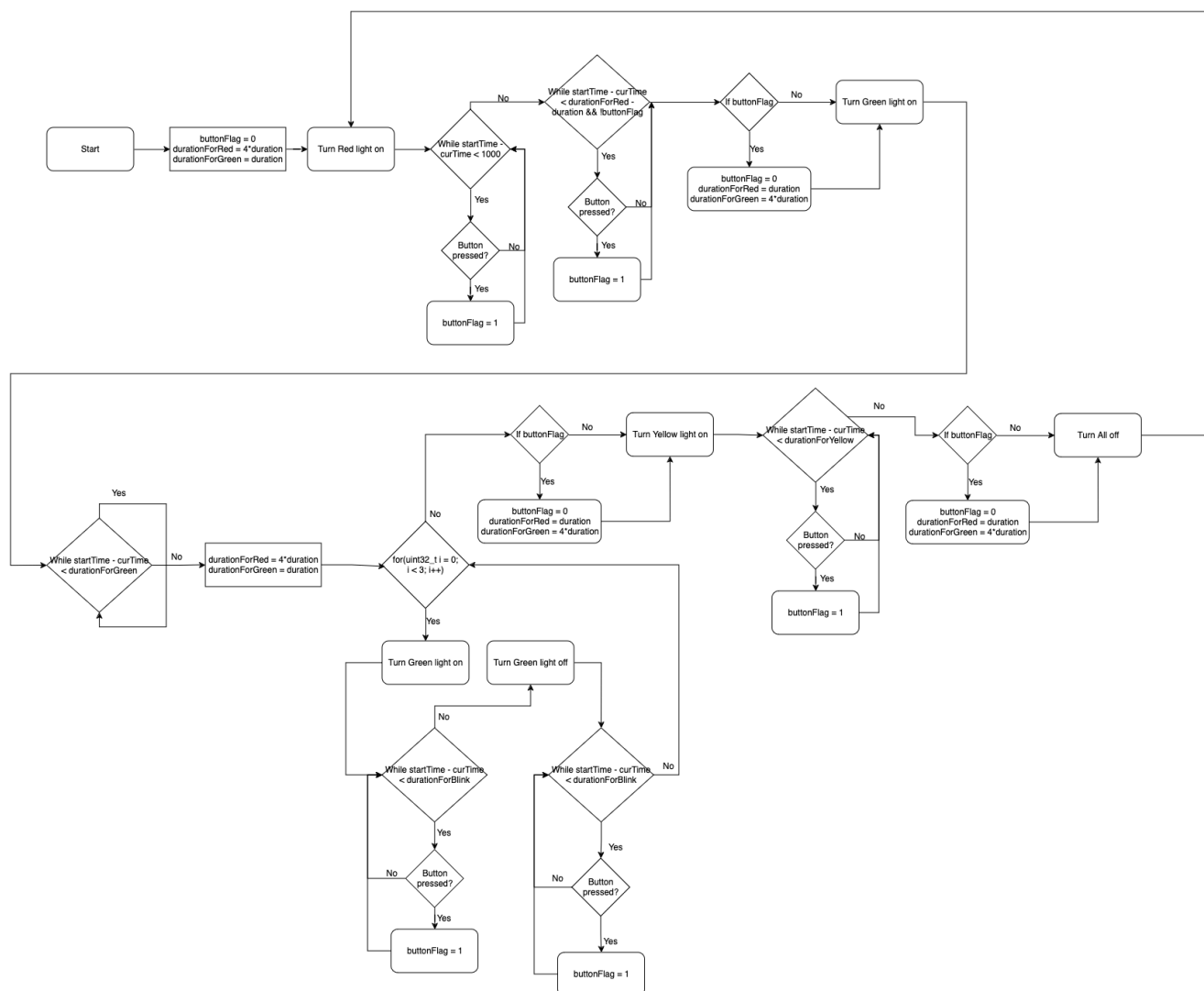
Основное задание

Разработать и реализовать драйверы управления светодиодными индикаторами и чтения состояния кнопки стенда SDK-1.1M (индикаторы и кнопка расположены на боковой панели стенда). Функции и другие компоненты драйверов должны быть универсальными, т. е. пригодными для использования в любом из вариантов задания и не должны содержать прикладной логики программы. Функции драйверов должны быть неблокирующими, то есть не должны содержать задержек на определенное время с использованием активного ожидания (функция HAL_Delay() и собственные варианты аналогичной реализации), а также активного ожидания событий в циклах. Написать программу с использованием разработанных драйверов в соответствии с вариантом задания.

Задание по варианту

Сымитировать работу светофора пешеходного перехода. Светофор циклически переключает цвета в следующем порядке (порядок условный, соответствие реальному светофору не соблюдается): красный, зелёный, зелёный мигающий, жёлтый, снова красный и т. д. По умолчанию период горения красного в четыре раза больше периода горения зеленого. Если во время горения зеленого мигающего, желтого или красного нажимается кнопка, светофор запоминает необходимость скорейшего переключения на зелёный. После нажатия кнопки последовательность переключения цветов не нарушается, но ближайший период горения красного должен быть сокращен до $\frac{1}{4}$ своего обычного периода. Если кнопка нажата во время горения красного, когда он уже горит более $\frac{1}{4}$ периода, то сразу происходит переключение на зеленый. Следующий красный должен снова гореть полную длительность, если только снова не будет нажата кнопка.

Блок схема



Код программы

Конфигурация пинов

```
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15, GPIO_PIN_RESET);
```

```
/*Configure GPIO pin : PC15 */
```

```

GPIO_InitStruct.Pin = GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

```

```

/*Configure GPIO pins : PD13 PD14 PD15 */
GPIO_InitStruct.Pin = GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

```

Функция ожидания заданного временного интервала

```

uint8_t wait(uint32_t duration, uint8_t returnWhenButtonPressed) {
    uint32_t startTime = HAL_GetTick();
    uint8_t buttonFlag = 0;
    while((HAL_GetTick() - startTime) < duration){
        if ((HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15) == GPIO_PIN_RESET)) {
            buttonFlag = 1;
            if (returnWhenButtonPressed > 0) {
                return buttonFlag;
            }
        }
    }
    return buttonFlag;
}

```

Функция выключения всех световых индикаторов

```

void turnAllOff(){
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
}

```

Функция включения определенного светового индикатора

```

void turnSpecificLightOn(uint16_t light){
    turnAllOff();
    uint16_t pin = getPinByColor(light);
    HAL_GPIO_WritePin(GPIOD, pin, GPIO_PIN_SET);
}

```

```
}
```

Функция получения определенного пина в зависимости от цвета

```
uint16_t getPinByColor(LightColors color) {  
    switch (color) {  
        case GREEN:  
            return GPIO_PIN_13;  
        case YELLOW:  
            return GPIO_PIN_14;  
        case RED:  
            return GPIO_PIN_15;  
        default:  
            return 0;  
    }  
}
```

Основная программа

```
uint32_t duration = 1000;  
uint32_t blinkDuration = 500;  
uint32_t durationForGreen = duration;  
uint32_t durationForRed = 4 * duration;  
uint32_t durationForYellow = 2000;  
uint8_t buttonFlag = 0;  
/* USER CODE END 2 */  
  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
while (1){  
    turnSpecificLightOn(RED);  
    buttonFlag = wait(duration, 0);  
    if (buttonFlag == 0 && duration < durationForRed) {  
        buttonFlag = wait(durationForRed - duration , 1);  
    }  
  
    if (buttonFlag == 1) {  
        durationForRed = duration;  
        durationForGreen = duration * 4;  
        buttonFlag = 0;  
    }  
}
```

```

    }

    turnSpecificLightOn(GREEN);
    wait(durationForGreen, 0);

    durationForRed = 4 * duration;
    durationForGreen = duration;

    for(uint32_t i = 0; i < 3; i++){
        turnAllOff();
        buttonFlag = wait(blinkDuration, 0);
        turnSpecificLightOn(GREEN);
        buttonFlag = wait(blinkDuration, 0) | buttonFlag;
        if (buttonFlag == 1) {
            durationForRed = duration;
            durationForGreen = duration * 4;
            buttonFlag = 0;
        }
    }

    turnSpecificLightOn(YELLOW);
    buttonFlag = wait(durationForYellow, 0);
    if (buttonFlag == 1) {
        durationForRed = duration;
        durationForGreen = duration * 4;
        buttonFlag = 0;
    }
}

```

Вывод

Во время выполнения лабораторной работы мы познакомились со стендом SDK 1.1M и изучили работу интерфейсов ввода-вывода в микроконтроллерах. А также разработали программу для управления световыми индикаторами, имитирующую работу светофора.