Кафедра вычислительной техники



**ЛАБОРАТОРНАЯ РАБОТА №1**

**по дисциплине:** *периферийные устройства информационных систем*

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| Студент гр. *АВТ-819*, *АВТФ* | *Токарев В.Г.* |
| *Ванин К.Е.* |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Содержание

[Цель и задание 3](#_Toc33190341)

[Ход работы 3](#_Toc33190342)

[Вывод 5](#_Toc33190343)

# **Цель и задание**

- создайте новое приложение в среде разработки CooCox® CoIDE, в соответствии с имеющейся отладочной платой. Приложение должно управлять светодиодами, подключенным к выводам PB6, PB7, PB8, PB9, C13 платы STM32F103mini.

- используйте режим «Debug in RAM» для отладки Вашего приложения.

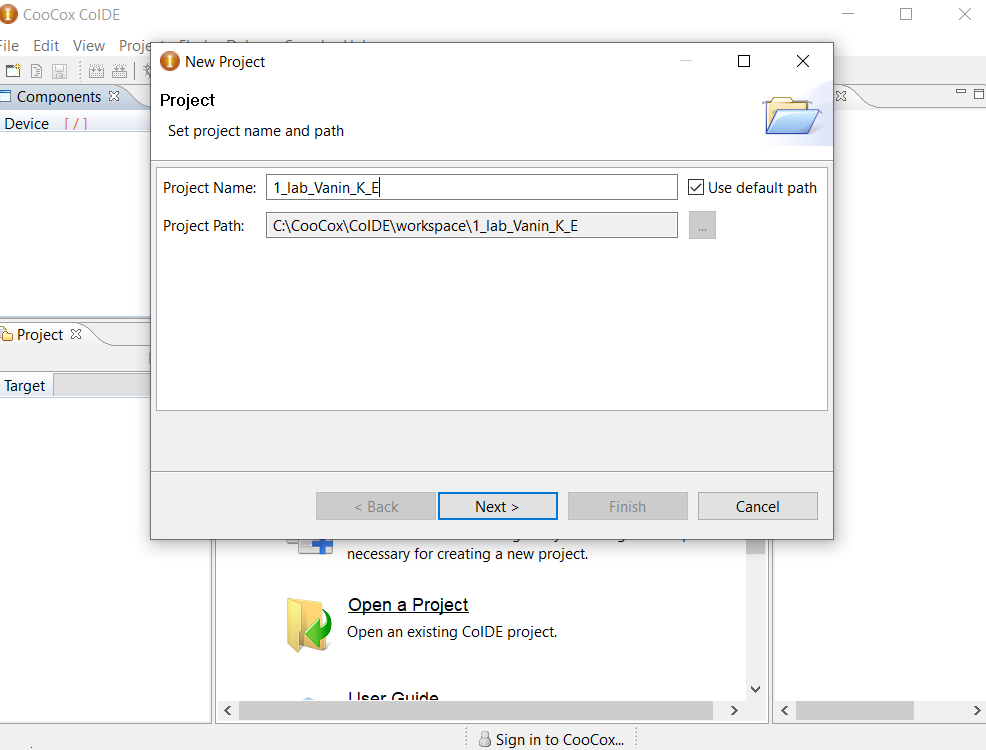
-в режиме отладки с использованием пошагового режима и точек останова изучите поведение регистров GPIOB, сравните значения в регистрах с поведением светодиодов.

- организуйте циклический опрос пользовательских кнопок WKUP(PA0) и USER(PA15), отображая на светодиодах состояние кнопок (нажата-отжата).

# **Ход работы**

Установим среду разработки CooCox CoIDE.

Запустим и создадим первый проект , выбрав плату STM32F103СB.



В окне репозитория отметим необходимые для проекта элементы: «C Library», «CMSIS Core», «CMSIS Boot», «RCC» и «GPIO».

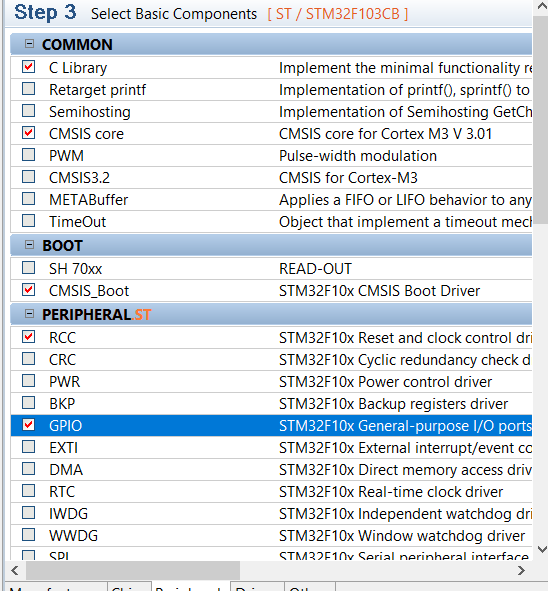


Рисунок 1.Выбор компонентов для разработки

Была разработана программа, которая последовательно включала и выключала светодиоды PС13, PB8. Также было выявлено, что PB и PC инициализируются противоположными значениями (0 и 1 соответственно).

Управление последовательным включением и выключением светодиодов PC13, PB8:

int main(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_6|GPIO\_Pin\_7|GPIO\_Pin\_8|GPIO\_Pin\_9;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB, &GPIO\_InitStructure);

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_13;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

int i;

while(1)

{

GPIOC->ODR ^= GPIO\_Pin\_13;

for(i=0;i<0x100000;i++);

GPIOB->ODR ^= GPIO\_Pin\_8;

for(i=0;i<0x100000;i++);

}

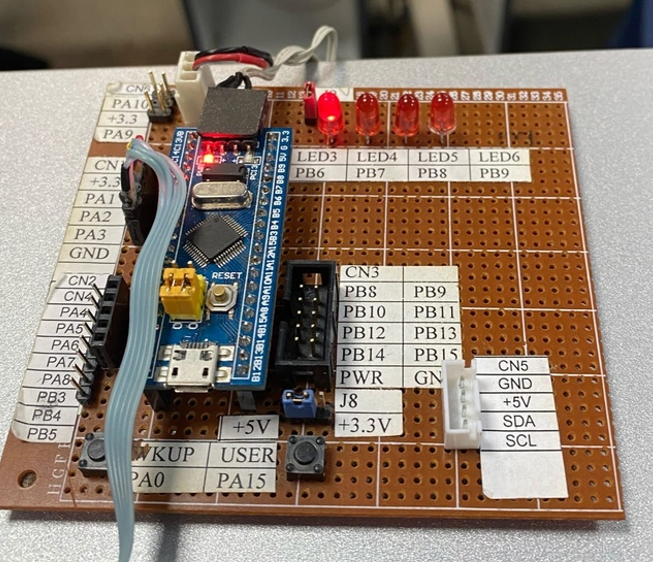


Рис. 1 Волнообразное изменение состояния светодиодов, подключенных к портам PB6, PB8, PB9:

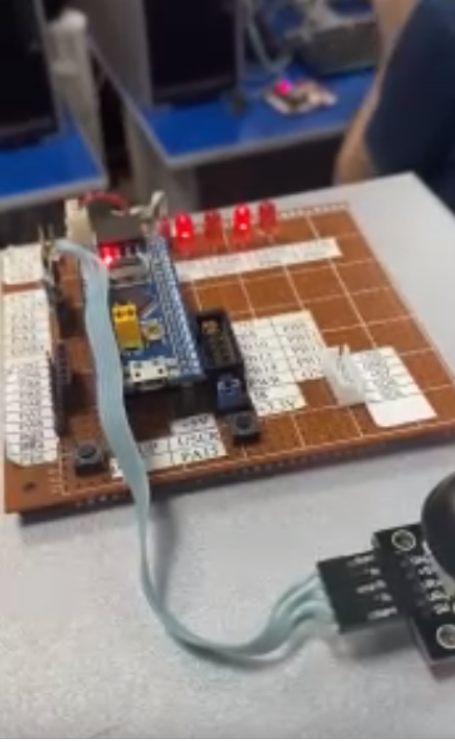


Рис. 2 Волнообразное изменение состояния светодиодов, подключенных к портам PB6, PB8, PB9:

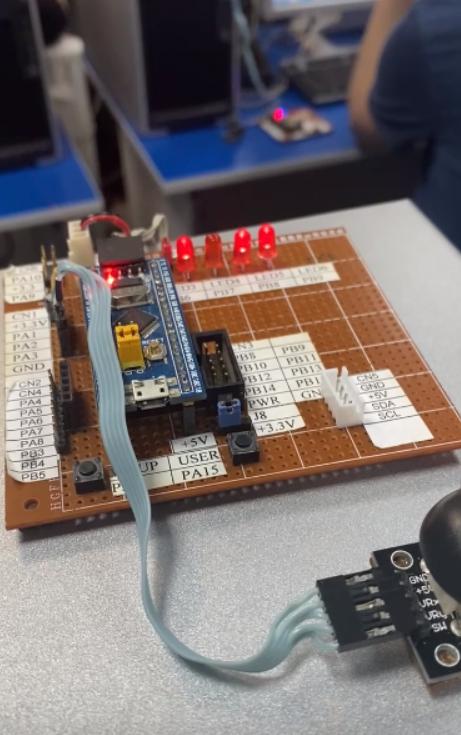


Рис.3 Волнообразное изменение состояния светодиодов, подключенных к портам PB6, PB8, PB9:

int main(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_6|GPIO\_Pin\_7|GPIO\_Pin\_8|GPIO\_Pin\_9;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB, &GPIO\_InitStructure);

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_13;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

int i;

}

while(1)

{

GPIOB->ODR ^= GPIO\_Pin\_6;

for(i=0;i<0x100000;i++);

GPIOB->ODR ^= GPIO\_Pin\_8;

for(i=0;i<0x500000;i++);

GPIOB->ODR ^= GPIO\_Pin\_9;

for(i=0;i<0x100000;i++);

}

}

Была разработана программа, согласно которой при нажатии на кнопку PA0 загорается светодиод PB6,при отпускании кнопки светодиод PB6 гаснет. При нажатии на кнопку PA15 загорается светодиод PB8, при отпускании светодиод PB8 гаснет.

Изменение состояния светодиода по нажатию кнопки:

int main(void)

{

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_13;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

int i;

while(1)

{

uint8\_t Flag = GPIO\_ReadInputDataBit (GPIOA, GPIO\_Pin\_15);

if (Flag) GPIOC->ODR ^= GPIO\_Pin\_13;

}

}

# **Вывод**

В результате выполнения лабораторной работы было выполнено знакомство со средой разработки CooCox CoIDE, были изучены принципы управления портами контроллера STM32F103C8. Были написаны программы для изменения состояния светодиода по нажатию кнопки, а также для волнообразного переключения светодиодов.