

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

Індивідуальні завдання
щодо лабораторних робіт при вивченні дисципліни
“Проектування мехатронних систем”
для студентів спеціальності

122 — “Комп’ютерні науки” заочної форми навчання

Індивідуальні завдання щодо лабораторних робіт при вивченні дисципліни “Проектування мехатронних систем” для студентів спеціальності 122 — “Комп’ютерні науки” заочної форми навчання (рос. мов.) Укл.: А. І. Гуда, А. Ю. Зимогляд. — Дніпро: НМетАУ, ІВК “Системні технології”, 2019. — 13 с.

Викладено завдання до лабораторних робіт дисципліни “Проектування мехатронних систем”, на основі яких студент освоює основні питання дисципліни.

Призначена для студентів для студентів 122 — “Комп’ютерні науки” заочної форми навчання.

Укладачі:

А. І. Гудав, д-р техн. наук, проф., доц. каф. ІТС

А. Ю. Зимогляд, асс. каф. ІТС.

Відповідальний за випуск О. І. Михальов, д-р техн. наук, проф.

Рецензент

В. І. Корсун, д-р техн. наук, професор (НГУ)

Друкується за авторською редакцією.

Затверджено на засіданні кафедри інформаційних технологій і систем, протокол №9 від 06.03.2019.

Підписано до друку 15.03.2019. Формат 60x84 1/16. Папір друк.

Національна металургійна академія України
49600, Дніпро-5, пр. Гагаріна, 4
ІВК “Системні технології”

Лабораторна работа № 1 (2019).

Управление GPIO

Написать программу для контроллера STM32F446RET на отладочной плате STM32F446 Nucleo 64 на языке C/C++ (по выбору), реализующую заданный (по варианту задания) цикл управления группой светодиодов. Для разработки и начальной конфигурации рекомендуется использовать программные комплексы STM32CubeMX или STM32CubeIDE. Для записи программы в контроллер можно использовать встроенные средства STM32CubeIDE, скрипт “STFL” (выдаётся преподавателем) или же команду

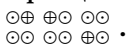

```
st-flash --reset write <file_name.bin> 0x8000000
```

К отладочной плате подключён блок светодиодов с ограничивающими резисторами, которые, в свою очередь, подключены к Gnd. Для подключения используются выводы GPIO C0—C3.

Программа должна выполнить следующие действия:

- Произвести инициализацию HAL;
- Настроить RCC таким образом, чтобы обеспечить тактирование ядра MCU с частотой 168 или 180 MHz (по выбору) при использовании внешнего тактового сигнала 8 MHz. Настройку тактирования шин APB1 и APB2 произвести так, что бы обеспечить максимальные допустимые частоты при данных условиях.
- Настроить GPIO таким образом, чтобы обеспечить работу светодиодов при данном способе подключения.
- В основном бесконечном цикле программы управлять блоком светодиодов. Способ управления — на выбор (HAL, прямой доступ к регистрам). Ограничение — не используемые в работе биты регистра вывода должны сохранять своё состояние. Для задержки выполнения между фазами рекомендуется использовать функцию HAL_Delay.

Варианты заданий (используется 1 + остаток от деления номера студента в списке на количество вариантов):

1. Вращение по часовой стрелке, одновременно — один горящий светодиод .
2. Вращение против часовой стрелки, одновременно — один горящий светодиод .

3. Вращение по часовой стрелке, одновременно — три горящих светодиода $\oplus\oplus$
 $\oplus\ominus \oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus \oplus\oplus$.
4. Вращение против часовой стрелки, одновременно — три горящих светодиода $\oplus\oplus \oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus \oplus\oplus \oplus\oplus$.
5. Попеременное включение диагоналей $\oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus$.
6. Параллельное смещение вверх $\oplus\oplus \oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus \oplus\oplus$.
7. Параллельное смещение вправо $\oplus\oplus \oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus \oplus\oplus$.
8. Параллельное смещение влево $\oplus\oplus \oplus\oplus \oplus\oplus$
 $\oplus\oplus \oplus\oplus \oplus\oplus$.

Вопросы для самопроверки:

1. Каким образом надо настраивать GPIO для управления светодиодами в Вашем случае?
 2. Какие ещё способы подключения возможны и какая настройка будет необходима при этом?
 3. Как Вы настраивали блок RCC?
 4. Какие регистры позволяют изменять состояние выводов GPIO?
 5. Какие регистры позволяют узнать состояние выводов GPIO?
 6. Какие регистры управляют режимами работы выводов GPIO?
-

Лабораторна работа № 2 (2019).

Использование NVIC и EXTI

Написать программу для контроллера STM32F446RET на отладочной плате STM32F446 Nucleo 64 на языке C/C++ (по выбору), демонстрирующую с помощью обработчика прерывания заданную реакцию (по варианту задания) на изменение уровня внешнего логического входа. Индикацию реакции реализовать с помощью группы светодиодов. Для разработки и начальной конфигурации рекомендуется использовать программные комплексы STM32CubeMX или STM32CubeIDE.

В качестве источника сигнала используется нажимная кнопка, подключённая в вывод C13. Нажатое состояние соответствует низкому уровню сигнала, отпущенному — высокое. К отладочной плате подключён блок светодиодов с ограничивающими резисторами, которые, в свою очередь, подключены к Gnd. Для подключения используются выводы GPIO C0—C3.

Программа должна выполнить следующие действия:

- Произвести инициализацию HAL;
- Настроить RCC таким образом, чтобы обеспечить тактирование ядра MCU с частотой 168 или 180 MHz (по выбору) при использовании внешнего тактового сигнала 8 MHz. Настройку тактирования шин APB1 и APB2 произвести так, что бы обеспечить максимальные допустимые частоты при данных условиях.
- Настроить GPIO C0—C3 для индикации с помощью светодиодов.
- Определить имя и создать обработчик прерывания. Допускается как использование функций HAL (callback function), так непосредственная работа с регистрами
- Настроить EXTI и GPIO C13 для генерации прерывания по заданному событию.
- Настроить NVIC, выбрав приоритет используемого прерывания. Разрешить прерывание.
- В основном бесконечном цикле программы отображать состояние с помощью светодиодов.

Варианты заданий (используется 1 + остаток от деления номера студента в списке на количество вариантов):

1. Счёт в прямом направлении. Индикация — 4 бита.
2. Счёт в обратном направлении. Индикация — 4 бита.

3. Изменение частоты индикации. Произвольное кол-во бит.
4. Изменение скважности индикации. Произвольное кол-во бит.

Вопросы для самопроверки:

1. Сколько бит выделено на приоритет в архитектуре ARM и в используемом контроллере? Как они связаны?
 2. Как можно разделить биты приоритета на вытесняющий приоритет и суб-приоритет? Как это реализовано у Вас в программе и почему?
 3. Как было выбрано значение приоритета для Вашей задачи?
 4. Как снимается запрос на прерывание от EXTI? Что будет, если его не снять?
 5. Какое событие EXTI приводит к прерыванию в Вашей программе?
-

Лабораторна работа № 3 (2019).

Использование UART для обмена данных с компьютером

Написать программу для контроллера STM32F446RET на отладочной плате STM32F446 Nucleo 64 на языке C/C++ (по выбору), обеспечивающую связь с компьютером с помощью UART. Индикацию состояния реализовать с помощью группы светодиодов. Для разработки и начальной конфигурации рекомендуется использовать программные комплексы STM32CubeMX или STM32CubeIDE.

В данной работе используется USB-UART конвертор, включенный в состав платы STM32F446 Nucleo 64. Он подключён к UART2 (GPIOA.2 и GPIOA.3). Название соответствующего файла устройства (например, “/dev/tty_usb0”) уточняйте у преподавателя. Для связи можно использовать программы `picocom`, `minicom`, `putty` и другие (см. `manual`). К отладочной плате подключён блок светодиодов с ограничивающими резисторами, которые, в свою очередь, подключены к Gnd. Для подключения используются выводы GPIO C0—C3.

Перед написанием программы следует определить требуемые настройки UART. По умолчанию — 115200/N/1.

Программа должна выполнить следующие действия:

- Произвести инициализацию HAL;
- Настроить RCC таким образом, чтобы обеспечить тактирование ядра MCU с частотой 168 или 180 MHz (по выбору) при использовании внешнего тактового сигнала 8 MHz. Настройку тактирования шин APB1 и APB2 произвести так, что бы обеспечить максимальные допустимые частоты при данных условиях.
- Настроить GPIO C0—C3 для индикации с помощью светодиодов.
- Настроить GPIO для заданного UART.
- Включить и настроить UART. При необходимости — настроить NVIC и прерывания UART.
- В основном бесконечном цикле программы выполнять требуемые действия по обмену и индикации. При необходимости — часть действий производить в обработчике прерываний.

Варианты заданий (используется 1 + остаток от деления номера студента в списке на количество вариантов):

1. Приход символов ‘0’–‘3’ инвертирует состояние соответствующего светодиода. Текущее состояние выводится в виде строки “+-”.

2. Приход символов '0'–'3' включает соответствующий светодиод, а '4'–'7' — выключает. Текущее состояние выводится в виде строки "++-".
3. Пришедшие символы накапливаются в буфере, и по приходу "Enter" строка выводится. При этом инвертируется состояние "младшего" светодиода.
4. Пришедшие символы выводятся сразу по их приходу. По приходу "Enter" инвертируется состояние "младшего" светодиода.

Вопросы для самопроверки:

1. Какие линии связи используются в UART и как определяется их уровни?
 2. Когда можно записывать следующий символ для передачи?
 3. Как определяется завершение приёма следующего символа?
 4. Что такое overrun? Как определить и выключить это состояние?
 5. Из каких элементов состоит одна посылка в UART?
-

Лабораторна работа № 4 (2019).

Использование таймера в режиме PWM для управления электродвигателем и сервоприводом

Написать программу для контроллера STM32F446RET на отладочной плате STM32F446 Nucleo 64 на языке C/C++ (по выбору), реализующую управление по заданному закону электродвигателем или сервоприводом с помощью режима PWM таймера. Индикацию фаз управления реализовать с помощью группы светодиодов. Для разработки и начальной конфигурации рекомендуется использовать программные комплексы STM32CubeMX или STM32CubeIDE.

В качестве таймера следует использовать TIM1. Используемый канал — любой. При управлении коллекторным двигателем используется внешняя ключевая схема. Управление сервоприводом — непосредственное. В обоих случаях используется внешнее питание. Значения рабочих частот и временных интервалов уточняйте к преподавателя. Перед включением схему обязательно **показать преподавателю!** К отладочной плате подключён блок светодиодов с ограничивающими резисторами, которые, в свою очередь, подключены к Gnd. Для подключения используются выводы GPIO C0—C3. Начальную отладку можно проводить на дополнительном светодиоде, подключенном в выбранному каналу таймера.

Перед написанием программы следует, исходя из заданной частоты, входной частоты таймера и диапазона заполненности PWM сигнала рассчитать значения регистров PSC, ARR, CCRx.

Программа должна выполнить следующие действия:

- Произвести инициализацию HAL;
- Настроить RCC таким образом, чтобы обеспечить тактирование ядра MCU с частотой 168 или 180 MHz (по выбору) при использовании внешнего тактового сигнала 8 MHz. Настройку тактирования шин APB1 и APB2 произвести так, что бы обеспечить максимальные допустимые частоты при данных условиях.
- Настроить GPIO C0—C3 для индикации с помощью светодиодов.
- Настроить GPIO для выбранного канала таймера.
- Включить и настроить заданный таймер. Настроить его выбранный канал.
- В основном бесконечном цикле программы изменять настройку выбранного канала для реализации заданного поведения.

Варианты заданий (используется 1 + остаток от деления номера студента в списке на количество вариантов):

1. Ступенчатое управление скоростью вращения коллекторного двигателя.

2. Плавное управление скоростью вращения коллекторного двигателя.
3. Ступенчатое управление положением вала сервопривода.
4. Плавное управление положением вала сервопривода.

Вопросы для самопроверки:

1. Чем определяется входная частота таймера?
 2. Чем определяется счётная частота таймера? Какой диапазон частот возможен для Вашей программы?
 3. Чем определяется выходная частота канала PWM таймера?
 4. Чем определяется заполненность сигнала канала PWM таймера? Какой минимальный шаг изменения?
 5. Какие режимы PWM существуют? Какой используется в Вашей программе и почему?
-

Лабораторна работа № 5 (2019).

Использование протокола I2C для обмена данными с внешним устройством

Написать программу для контроллера STM32F446RET на отладочной плате STM32F446 Nucleo 64 на языке C/C++ (по выбору), реализующую обмен данным по протоколу I2C с заданным устройством и выводом полученных данных в UART. Индикацию состояния (дополнительно) можно реализовать с помощью группы светодиодов. Для разработки и начальной конфигурации рекомендуется использовать программные комплексы STM32CubeMX или STM32CubeIDE.

К отладочной плате подключён блок светодиодов с ограничивающими резисторами, которые, в свою очередь, подключены к Gnd. Для подключения используются выводы GPIO C0—C3.

Рекомендуется использовать I2C1 (GPIOB.8 и GPIOB.9). Перед написанием программы следует ознакомиться с описанием подключаемого устройства, изучить его режимы работы, регистры. Подключать устройство следует после загрузки прошивки. **Перед** включением схему обязательно **показать преподавателю!**

Программа должна выполнить следующие действия:

- Произвести инициализацию HAL;
- Настроить RCC таким образом, чтобы обеспечить тактирование ядра MCU с частотой 168 или 180 MHz (по выбору) при использовании внешнего тактового сигнала 8 MHz. Настройку тактирования шин APB1 и APB2 произвести так, что бы обеспечить максимальные допустимые частоты при данных условиях.
- Настроить GPIO C0—C3 для индикации с помощью светодиодов.
- Настроить GPIO для I2C и UART.
- Настроить UART для передачи данных в компьютер.
- Включить и настроить I2C в режиме мастера. В документации на подключаемое устройство найдите его адрес и используете его при дальнейшем обмене.
- В основном бесконечном цикле программы с заданным интервалом времени проводить опрос устройства и выводить данные в UART.

Варианты заданий (используется 1 + остаток от деления номера студента в списке на количество вариантов):

1. Получение данных от MEMS акселерометра и их отображение.

2. Получение данных от MEMS барометра BMP085/285 и их отображение.
3. Получение данных от ADC PCF8591 и их отображение.
4. Получение данных от MEMS магнитометра HMC5983 и их отображение.

Вопросы для самопроверки:

1. Какие линии данных используются в I2C и как следует настраивать соответствующие выводы GPIO контроллера?
 2. Как реализована адресация устройств и их регистров в I2C?
 3. Как и для чего реализуются состояния Start, Stop и Restart в I2C?
 4. При каких условиях можно подключить несколько устройств к одному каналу I2C?
-

Список рекомендованной литературы

1. Мехатроника: от структуры системы к алгоритму управления: / А. П. Губарев, О. В. Левченко. — К.: НТУУ “КПИ”, 2007. — 180 с.
2. Точность и надежность мехатронных систем: учеб. пособие / Б. М. Бржозовский, В. В. Мартынов, В. Я. Копп, М. Б. Бровкова. — Саратов: Саратов. гос. техн. ун-т, 2014. — 161 с.
3. Ядро Cortex-M3 компании ARM. Полное руководство / Джозеф Ю. — М.: Додека-XXI, 2012. — 552 с. ISBN 978-5-94120-243-0.
4. Beginning STM32 Developing with FreeRTOS, libopencm3 and GCC / Warren Gay // Standard Apress DOI: <https://doi.org/10.1007/978-1-4842-3624-6>
5. CortexTM-M3 Technical Reference Manual / ARM Limited. 2006 — 384 с.
6. UM1724. User manual. STM32 Nucleo-64 board / STMicroelectronics. — 2015 — 69 с.
7. RM0390. Reference manual. STM32F446xx advanced ARM®-based 32-bit MCUs / STMicroelectronics. — 2015 — 1323 с.
8. AN4776. Application note. General-purpose timer cookbook / STMicroelectronics. — 2017 — 73 с.
9. PM0214. Programming manual. STM32F3 Series, STM32F4 Series, STM32L4 Series and STM32L4+ Series Cortex®-M4 programming manual / STMicroelectronics. — 2017 — 260 с.