

- 1) Создать новый проект для написания программного кода на языке Си
- 2) Добавить в проект заголовочный файл `main.h`, который будет содержать прототип подпрограммы обработчика прерываний АЦП. В файле `stm8_interrupt_vector.c` в таблице векторов прерывания изменить имя подпрограммы обработчика прерывания АЦП. В файле `main.c` создать тело подпрограммы обработчика прерываний АЦП.
- 3) В основной программе с помощью бита `EOCIE` в регистре `ADC_CSR` разрешить прерывания АЦП по окончании преобразований и отключить маскирование прерываний. Включение АЦП осуществляется с помощью бита `ADON` в регистре `ADC_CR1`. Если бит `ADON` устанавливается в «1» в первый раз, то это включает АЦП, для запуска преобразований необходимо установить данный бит в «1» повторно. Добавьте в конце основной программы бесконечный цикл.
- 4) В обработчике прерываний необходимо в две различные переменные считать результат преобразования, который, по умолчанию хранится в регистрах `ADC_DRH` и `ADC_DRL`. Также в конце обработчика прерывания необходимо очистить флаг `EOC` в регистре `ADC_CSR`.
- 5) На плате соедините нулевой аналоговый вход контроллера (можно найти в документации на контроллер) с верхним выводом разъема `P69` (возле джойстика). Таким образом вы соедините нулевой аналоговый вход контроллера с выходом джойстика, отвечающим за наклон по оси `X`.
- 6) Загрузите программу в микроконтроллер, поставьте точку останова на команду очистки флага прерывания. Откройте окно просмотра переменных и добавьте переменные в которых будет храниться результат преобразования. Посмотрите значения переменных в 3 случаях: а) джойстик находится в среднем положении (положение по умолчанию); б) джойстик отклонён по оси `X` максимально вправо; в) джойстик отклонён по оси `X` максимально влево.
- 7) В основной программе, до запуска преобразования, с помощью бита `ALIGN` в регистре `ADC_CR2` изменить выравнивание данных на выравнивание по правому краю.
- 8) Повторить пункт 6
- 9) В документации на контроллер (Reference manual на стр.440) ознакомиться с особенностями хранения результата преобразования при различном выравнивании данных. **Обратите внимание на порядок чтения регистров при разном выравнивании данных.** Изменить в обработчике прерывания считывание результата таким образом, чтобы в зависимости от выставленного выравнивания данных, результат преобразования записывался в одну переменную. Так как микроконтроллеры семейства `stm8` имеют 10-ти разрядный АЦП, то результат преобразования должен лежать в диапазоне от 0 до 1023.
- 10) Проверить, работу программы для трех положений джойстика с разным выравниванием данных.
- 11) С помощью битов `CH[3:0]` в регистре `ADC_CSR` настроить вместо нулевого канала первый канал АЦП. Подключить к первому аналоговому входу микроконтроллера выход джойстика, отвечающий за наклон по оси `Y`. Проверить правильность работы программы.
- 12) С помощью бита `CONT` в регистре `ADC_CR1` задать режим непрерывного преобразования. Написать код программы изменения яркости светодиода с помощью джойстика. Яркость должна зависеть от угла наклона джойстика.
- 13) Написать код программы, выводящей прямоугольные импульсы на нулевой вывод порта `G`. Время импульсов должно изменяться при наклоне джойстика по оси `X`, а время паузы должно изменяться при наклоне джойстика по оси `Y`. АЦП должен включать каналы по очереди. Проверить правильность работы с помощью осциллографа.

Описание выводов микросхемы `AD7801`:

`D7-D0` - Входы параллельных данных

`CS` – Вход выбора микросхемы. Активный уровень – низкий.

`WR` – Вход записи данных. Активный уровень – низкий. Используется совместно с входом `CS`.

`PD` – данный вход используется для перевода микросхемы в режим низкого потребления (отключение). Активный уровень – низкий.

`LDAC` - Когда на этот логический вход подается низкий уровень, на выходе ЦАП обновляется содержимое его регистра ЦАП. Если `LDAC` постоянно соединен с низким уровнем, ЦАП обновляется по переднему фронту `WR`.

CLR - Когда на этот вход подается низкий уровень, регистр ЦАП загружается всеми нулями, а на выходе ЦАП уровень напряжения падает до нуля вольт.

Более подробное описание режимов работы микросхемы ЦАП AD7801BRZ можно найти в технической документации на серию микросхем AD7801.

14) Получить на выходе микросхемы ЦАП пилообразный сигнал с произвольным периодом.

15) Получить на выходе микросхемы ЦАП сигнал, соответствующий варианту из таблицы 1.

Таблица 1 – Исходные данные к заданию 15

Вар.	Форма сигнала	Вар.	Форма сигнала	Вар.	Форма сигнала
1		8		15	
2		9		16	
3		10		17	
4		11		18	
5		12		19	
6		13		20	
7		14		21	