**ДЗ №1. Программирование-1**

Сдача ДЗ - в виде кода (на флешке, не на бумаге и не по электронной почте).

Код должен:

* быть оформлен в виде проекта AVR Studio,
* должен компилироваться, запускаться
* должен быть правильно отформатирован (количество tab'ов, говорил на семинаре).
* и  должен позволять посмотреть ответ в отладчике для всех задач (надо выключить оптимизацию).

**ДЗ №2. Позиционные системы счисления**

Принимается на бумаге. Должны быть продемонстрированы вычисления, приводящие к ответу (чтобы проверить правильность рассуждений и умение производить вычисления без компьютера). Результат округлить до ближайшего в соответствии с таблицей (при округлении, например, до 4 десятичных разрядов, необходимо вычислить 5 разрядов).

|  |  |
| --- | --- |
| **Таблица. Разрядность при округлении результата** | |
| **Преобразование** | **Количество разрядов в дробной части числа** |
| bin2dec | 4 десятичных разряда |
| bin2hex | 4 шестнадцатеричных разряда |
| dec2hex | 4 шестнадцатеричных разряда |
| hex2bin | 16 двоичных разряда |
| hex2dec | 4 десятичных разряда |
| dec2bin | 8 двоичных разрядов |

**Вопросы к защите ДЗ2**

* Формула позиционной системы счисления. Представление дробных чисел с фиксированной точкой. Диапазон значений разрядов чисел. Разрядность дробной и целой части числа.
* Обоснование алгоритма перевода dec2bin для целой и дробной части. Возможно ли применение алгоритма для перевода dec2hex?
* Обоснование связи между двоичной и шестнадцатеричной системой счисления для целой и дробной части.

**ДЗ №3. Программирование-2**

Актуальны все требования к коду из ДЗ №1.

Задания должны быть оформлены в виде функций. Данные в функцию должны передаваться не через глобальную переменную, а через аргументы функции. Типы входных и выходных аргументов должны быть обоснованы с точки зрения задачи. Необходимо обоснованно выбрать разрядность применяемого целого типа (char, int, long) и его знаковость (signed, unsigned). Функциям необходимо дать адекватные имена. Хотя бы транслитом, но лучше по- английски.

**Пример (задача Integer 24)**. Функцию, вычисляющую номер дня недели по номеру дня в году логично назвать day\_of\_week или DayOfWeek. Поскольку аргумент функции лежит в диапазоне 1-365, поэтому наиболее близкий тип данных – **unsigned int** (0-65535). Возвращаемое значение лежит в диапазоне 0-6, поэтому тип результата – **unsigned char** (0-255).

**unsigned char** day\_of\_week(**unsigned int** day\_of\_year)

{

**return** day\_of\_year % 7;

}

Помните: программу выполняет компьютер, но читает человек. Описывать эти рассуждения на бумаге не надо, но надо быть готовым проговорить их устно.

**Замечания по разделам**

* + Раздел 4. Все переменные целые.
  + Раздел 5. Ветвление if использовать не надо. Требуется вычислить булево выражение.
  + Раздел 6. Вместо отсутствующего в «С» булевого типа следует использовать char.

Защита ДЗ – ответы на вопросы по написанному коду. Вопросы будут направлены на проверку знаний по материалу, изложенному на семинарах (типы переменных в «С», ветвление if, цикл while, подпрограммы (функции) и др.).

**Лабораторная работа №1. Порты ввода/вывода**

**(Срок защиты – до 10 ноября)**

**Требования к коду:**

Дух требований: если внешне программа работает, но ее структура плохая (труднообъяснимая логика, не использованы приемы, изложенные на семинарах и упрощающие код, **программа списана у соседа без понимания**), то и программа плохая. Плохие программы не принимаются, пока не станут хорошими.

* Код должен быть отформатирован по тем же правилам, что и ДЗ №1,3 (количество tab'ов внутри операторных скобок).
* Должны использоваться именованные константы (#define), где это целесообразно (например, для указания количество элементов массива, длительность паузы, **и в других подобных случаях!**).
* Использование массивов. Например, для светофора **не надо** писать код типа:

**while** (1) {

PORTA = 0b01111111; // код красного светодиода

\_delay\_ms(1000);

PORTA = 0b01011111; // код красного и зеленого светодиода

\_delay\_ms(1000);

// и т.д. для остальных цветов

}

В этом **и подобных** случаях используйте массивы.

* Обязательно использование побитовых операций вместо сравнений типа PIND == 0b11111011.
* Если есть очевидный повторяющийся или логически отдельный блок кода, его требуется выделить в подпрограмму. Входные переменные, их тип должны быть обоснованы. Тип результата также должен быть обоснован. Нельзя использовать глобальные переменные для передачи параметров в функции.

**Примерные вопросы к защите:**

* Какие устройства из курсов Электроника УЦА, ЦИП, можно подключить к микроконтроллеру. Назовите 3-5.
* Количество портов ввода/вывода микроконтроллера ATMega16. Их разрядность и возможность двунаправленной работы. Принципиальная схема выводов портов ввода/вывода ATMega16. Потребитель и источник сигнала. Пассивный и активный источник сигнала. Подтягивающий резистор.
* Программирование портов ввода/вывода. Назначение регистров DDRx, PORTx, PINx: как сконфигурировать режим ввода/вывода, как включить и выключить подтягивающий резистор, как записать данные в порт и прочитать из порта. Как управлять ножками портов отдельно: например, в режиме вывода подать на одну ножку порта лог. 1 (или лог. 0), а остальные оставить в неизменном состоянии.
* Регистрация момента нажатия кнопки в программе. Обобщение на случай нескольких кнопок (рассказывалось на семинаре 23.10.2012).
* Таблицы истинности побитовых (поразрядных) операций – И, ИЛИ, Искл. ИЛИ. Применение побитовых операций: проверка состояния бита («установлен» – лог. 1 / «сброшен» – лог. 0), установка или сброс бита. Трактовка переменных как битовых массивов и чисел в позиционной системе счисления. Сравнение побитовых и логических операций.
* Маски битов. Создание маски для некоторого бита по его номеру с помощью сдвиговых операций.

**Ссылки**

1. <http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh128/7.htm>
2. <http://easyelectronics.ru/rabota-s-portami-vvoda-vyvoda-mikrokontrollerov-na-si.html>
3. <http://avrlab.com/node/31>
4. <http://roboforum.ru/users/robot/02.htm>

**Лабораторная работа №2. Дисплей, строки**

**(Срок защиты – предварительно до 2 декабря)**

**Требования к коду:**

Такие же, как и к ЛР1.

**Вопросы к защите:**

1. Кодировки символов: ASCII, расширенная ASCII (на примере win-1251). Зачем нужна кодовая страница для расширенной кодировки ASCII? Кодировка символов дисплея из лабораторного комплекта. Объем памяти, требуемый для хранения символа в кодировках ASCII, Unicode. Какой тип данных языка C, позволяет хранить символы в рассмотренных кодировках и почему?
2. Представление строк в языке C. Зачем нужен символ 0x00 в конце строки? (объяснить на примере функции вычисления длины строки). Использование указателей при работе со строками (и вообще, любой вопрос, рассмотренный в [тьюториале по указателям](http://wiki.markodelgroup.ru/lib/exe/fetch.php?media=pointers.rar)).
3. Функция sprintf и ее применение для целых, вещественных чисел, и строк. Будут элементарные задания на понимание.

**Лабораторная работа №3. Прерывания и таймеры**

**(Срок защиты – предварительно до 20 декабря)**

**ДЗ №4. Программирование-3 (циклы while, for, подпрограммы)**

**(Срок защиты – до 17 ноября)**

Требования те же, что и для ДЗ №3. Задания на тему массивов (раздел 13 Array) находятся в задачнике Абрамяна, ч.2.

Желающие получить более сложное, но более интересное и короткое задание, просьба подойти к Южанину В.В. Те, кто уже получал такие задание, также подойдите и узнайте о сокращенном варианте ДЗ №4 для Вас.

**Замечания по заданиям:** ждите объявлений дополнительно!

**ДЗ №5. Дополнительный код и вещественные числа**

**(срок защиты – до 24 ноября)**

**Требования к оформлению:**

Принимается на бумаге. Должны быть продемонстрированы вычисления, приводящие к ответу (чтобы проверить правильность рассуждений и умение производить вычисления без компьютера).

**Примерные вопросы к защите:**

* Определение дополнительного кода. Вычитание чисел с помощью аппаратного сумматора и дополнительного кода.
* Операция распространения знака.
* Формат IEEE-754 с одинарной и двойной точности. Прямой код со знаком, мантисса, степень.
* Машинный эпсилон и машинный нуль. Специальные значения: 0, -∞, +∞, NaN. Операции, в результате которых возникают спец. значения и способ их представления.

**Ссылки**

1. <http://www.softelectro.ru/ieee754.html>
2. Уэйкерли Дж. Проектирование цифровых устройств, 2002

**ДЗ №6. Программирование-4 (массивы, строки)**

**(отменена)**