ДЗ №1. Программирование
(только АТ)

Смысл задания состоит в подготовке к выполнению ЛР 1, которая подразумевает владение языком С и приемами отладки (break point, watch) в среде AVR Studio. Сдача ДЗ - в виде кода (на флешке). Код должен:

- быть оформлен в виде проекта AVR Studio, в настройках проекта должна быть отключена оптимизация;
- должен компилироваться, запускаться;
- должен быть правильно отформатирован (всюду количество tab'ов должно быть равно количеству открытых операторных скобок);
- должен позволять посмотреть ответ в отладчике для всех задач (надо выключить оптимизацию).

Решение задачи должно быть оформлено в виде отдельной функции. Данные в функцию 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 и переменным необходимо дать адекватные имена. Хотя бы транслитом (“Hotya by translitom”), но лучше по-английски. Помните: программу выполняет компьютер, но читает человек.

Пример (задача Integer 24). Требуется вычислить номер дня недели по номеру дня с начала года, считая, что 1 января - понедельник. Номера дней недели и дней года начинаются с нуля. Для решения задачи надо просто взять остаток от деления номера дня в году на 7 (дней в неделю). Логично выделить эту операцию в отдельную функцию и придумать ей удобоваримое название day_of_week или DayOfWeek. Поскольку аргумент функции лежит в диапазоне 0-365, то 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;
}
```

Для проверки функции следует придумать хотя бы один, а лучше несколько наборов данных для проверки. Для данного примера можно проверить, например, первый понедельник года (0 день в году) и второй понедельник года (7 день в году).

```
int main()
{
    unsigned char week_day1 = day_of_week(0);
    unsigned char week_day2 = day_of_week(7);
}
```

Учитывая то, что заданий много, все эти тестовые примеры будет неудобно запускать из **main**, поэтому каждое задание с тестами оформляется в виде отдельной функции, эти функции затем вызываются из **main**. Из названия функции должен быть ясен его номер и параграф задачника. Пример выполнения задания см. на след. странице

Назначенные консультанты из группы АСУ делают первичную проверку ДЗ у группы АТ и если их готовы слушать, делятся своими знаниями по программированию. Если более детально, консультант должен:

- убедиться, что код не слизан;
- проверить правильность решения ДЗ, расстановку отступов в коде и проч. требования (см. выше)
- проконсультировать в случае проблем;
- указать и проконтролировать исправление индусского кода, если такой встретится, хотя вас могут убеждать, что "все же и так же работает!";

– если у кого-то слишком много проблем, выявить их и по согласованию с преподавателем упростить задание, чтобы не заставлять человека прыгать выше головы;
– в отчете записать какие исправления были у каждого консультируемого (пишите прямо в исходниках в комментариях, комментарии соберите в одном месте)

Защита ДЗ – после первичной проверки у консультантов надо ответить 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.). Вопросы будут

Пример выполнения ДЗ по программированию:

```
unsigned char day_of_week(unsigned int day_of_year) {
    return day_of_year % 7;
}

void task4_24() {
    unsigned char week_day1 = day_of_week(0);
    unsigned char week_day2 = day_of_week(7);
}

void task5_15() {
    // решаем задачу 15
}
// ... аналогично оформляем остальные задачи

// вызываем все задачи из одного места - функции main()
int main()
{
    task4_24(); // вызываем решение задачи 24 параграфа 4
    task5_15(); // вызываем решение задачи 15 параграфа 5
    // ... аналогично вызываем решения других задач
}
```

ДЗ №2. Целые беззнаковые (Позиционные системы счисления) (для всех групп)

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:

– Формула позиционной системы счисления (формула (1) из лекций). Представление дробных чисел с фиксированной точкой. Диапазон значений разрядов чисел. Разрядность дробной и целой части числа.

– Чему равны n, p во всех заданиях ДЗ?
– Какой вид примет формула (1) для беззнаковых типов unsigned char, unsigned int, unsigned long в архитектурах AVR, x86.
– Как влияет размерность беззнакового типа данных на диапазон представимых чисел?
– Операция приведения типов для целых беззнаковых переменных:
    - от меньшей размерности типа к большей
    - от большей размерности типа к меньшей

  Привести примеры кода, в которых осуществляются оба этих варианта.

  Какая из этих операций более опасна? Приведите пример потери данных при операции приведения типа.
– Что такое целочисленное переполнение? В каких случаях опасно целочисленное переполнение?
– 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. Целые знаковые (Дополнительный код) (только АТ)

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.

ДЗ №4. Вещественные числа (Формат IEEE-754)
(для всех групп)

Требования к оформлению:

Те же, что и ДЗ №2.

При переводе из десятичной системы в двоичную считать все разряды мантиссы, причем необходимо учесть округление до ближайшего.

Примерные вопросы к защите:

- Нормализованная запись числа
- Формат IEEE-754 с одинарной и двойной точности. Формула перевода битового набора в значение?
- Почему в формате IEEE-754 хранится только вещественная часть мантиссы?
- Максимальное значение числа, представимого в формате IEEE-754. Сравнить максимальное число с одинарной точностью (32 бита, float) с максимальным значением 32-битного беззнакового целого.
- Минимальное по модулю значение числа, представимого в формате IEEE-754 (машинный нуль). Чему оно равно в десятичном коде?
- Какое значение и битовый набор будет у переменной float, если записать в нее 0?

```
int a = 0;
float f = a;
```

- Учитывая величину машинного нуля, чему будет равно *f* после выполнения следующего кода:

```
float f = 0;
for (int index = 0; index < 40; ++index)
{
    f *= 10;
}
```

- Специальные значения: 0, -∞, +∞, NaN. Операции, в результате которых возникают спец. значения и способ их представления.
- Погрешность представления чисел с плавающей точкой
  - Чем обусловлена погрешность представления вещественных чисел – погрешностью мантиссы M, погрешностью порядка e?
  - Чем равен машинный эпсилон?
  - Какая погрешность представления, например, числа π? Что будет с погрешностью, если увеличить представляемое число?

Ссылки

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

Лабораторная работа №1. Порты ввода/вывода

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:

Базовые (основные) вопросы

- Сколько ножек у порта микроконтроллера и сколько ножек на схеме (рис. 5) в электронных лекциях по портам ВВ
- Сколько бит информации хранится в одном триггере. Сколько бит информации в регистре DDRA. Как связан триггер DDRA3 и регистр DDRA?
- Написать на распечатке значение битов (принять PUD =0):
  - При записи в триггер логической 1 (режим записи). Как связан уровень ноги PA3 и значение триггера PORTA3
  - При записи в PORTA3 лог. 0 (режим чтения)
- Таблица истинности логического И. Побитовое и логическое И на примере
  0b100 & 0b010
  0b100 && 0b010 (логическая и побитовая интерпретация чисел).
  Логическая и побитовая инверсия на примерах ~0b010, !0b010
- Временная диаграмма положения кнопки. Возможные комбинации текущего и предыдущего лог. уровня на ноге порта ВВ. Таблица истинности для определения положения кнопки

x	Y	R
1	1	?
0	1	?
0	0	?
1	0	?

- Реализация таблицы истинности
- Разобрать пример положения «нужной» и « ненужной» кнопки. Просто вычислить значение переменной buttons

Дополнительные вопросы:

- 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

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 С, позволяет хранить символы в рассмотренных кодировках и почему?
2. Представление строк в языке С. 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 wiki.markodelgroup.ru/lib/exe/fetch.php?media=pointers.rar).
3. Написать функцию определения длины строки.
4. Числа и строки
    a. Чем отличаются 0, ‘0’, “0”?
    b. Функция sprintf и ее применение для целых, вещественных чисел, и строк. Будут элементарные задания на понимание.
5. Вопросы про адресную арифметику
6. Типизированный и нетипизированный указатель.
7. Два типа разыменование указателя *p и p[k].