

ЗУ v2.0z  
**МИКРОПРОГРАММА**  
(under construction)

**Вступление**

**Новая микро- и макро- программы совместимы только с версией печатной платы v2.0. Прошивки от предыдущих версий (1.x) к этой плате не подходят! Для плат предыдущих версий будут новые прошивки, базирующиеся на новом подходе, описанном в данной статье.**

Эта статья как и все статьи на этом сайте пишется с целью дать информацию по теме: микропроцессоры, ЖКИ, программирование, AVR, Mega32, Си, связь микропрограммы с Windows и прочие сопутствующие темы. Досталась эта информация с большим трудом. Надеюсь комунибудь пригодится.

Заявленная микропрограмма пишется одновременно с этой статьей. Такой способ помогает собрать все данные воедино и как бы вести дневник. Попробуем сформулировать что нового в новой микропрограмме:

1. В предыдущей версии было слишком много настроек и много специальных знаний и обозначений. Такое многообразие пугает.
2. В предыдущей версии была реализована идея шаблонов. В нее было заложено много возможностей, от этого пострадал размер. Новый алгоритм шаблона сжат и упрощен.
3. Новая идея этой версии: ЗУ — это программируемый БП. Для каждой химии пишется свой макрос.
4. Если идея сработает, то постараемся на сэкономленной памяти сделать авто определение и авто заряд.

**Как читать исходники.**

Не смотря на то, что программа содержит много специфических операторов, и это отпугивает читателей, программа является самым полным описанием самой себя и всего этого проекта. При известной степени тренировки чтение программы становится таким же легким, как и чтение обычного текста. До любой самой непонятной фразы, написанной человеком, можно докопаться и понять.

В программе сконцентрирован большой объем точной, взаимосвязанной информации, который невозможно воспринять сразу. Программу можно просматривать тематически: Как сделано то или это? Где оно встречается? Что будет, если заменить один блок на другой.

В микропрограмме много комментариев. Первое время можно читать только комментарии. Код программы содержит самую полную информацию, но к нему надо привыкать. Комментарии дублируют код и задают общий ход мысли, создавая иллюзию понимания.

Для того чтобы понимать код программы, надо впитать основы процессоров и язык программирования на котором написана микропрограмма. Ранее была сделана попытка максимально просто объяснить программирование в статье: **«Программирование на WinAVR Си зарядного устройства (ZU v1.7) на AT Mega32»** по адресу <http://avrcpp.narod.ru/doc/cpp.pdf>

Если отодвинуть от себя текст программы и взглянуть на блок-схему, можно увидеть весь проект в грубом приближении. Очень полезно зацепиться за эту схему и сказать себе. А-

а-а мне все ясно. Вся ваша программа - это просто инициализация переменных и бесконечный цикл в котором идет опрос клавиатуры, рисование экрана и управление процессами зарядки в зависимости от измеренных параметров.

Теперь можно углубиться в детали: А как сделана клавиатура? Что это за переменная? Можно узнать в файле описателей «MAIN.H». Если сделать поиск этой переменной по всем файлам проекта, то можно узнать какие подпрограммы ее используют: какие подпрограммы присваивают значение, какие считывают. Если в голове есть схема программы, то ваши розыски складываются в единую картину. Картина может поплыть от переизбытка информации, тогда нужно остановиться до следующего дня или до следующей чашки чая.

Пытаться рассказать как точно устроена программа — очень сомнительное занятие. Все же, постараюсь дать общие идеи и отослать к фрагментам программы. Некоторые фрагменты, которые останутся в тексте статьи, через некоторое время будут отличаться от реального текста в файлах. Только окончательная прошивка есть истина. Только текст, из которого скомпилирована и слинкована последняя прошивка, является подлинным.

Перед тем, как начать писать программу, надо взять листок бумаги и нарисовать блок-схему. Все крупные квадратики блок-схемы можно оформить в виде подпрограмм, тогда главный файл проекта «MAIN.CPP» будет понятен. Сначала внутренности подпрограмм будут «пустые», но наполненные комментариями, потом рядом с комментариями появится код. Программа заработает полностью только после написания последней подпрограммы, а результат хочется видеть по-быстрее. Поэтому план будет таков: постараться написать минимально рабочую прошивку которая инициализирует процессор и выведет на экран хоть что-нибудь. Если нам это удастся, то дело пошло, теперь мы можем контролировать как работают все наши подпрограммы.

Каждый день надо делать маленький шаг и тогда через месяц мы окажемся на большой высоте. И после каждого шага надо оглянуться на результат и дать себе оценку. В некоторые дни окажется, что шаги были сделаны не правильно, и тогда кажется, что работа была проделана напрасно. Это не так. Хождение вокруг проблемы - это тоже достижение, освещающее окрестности и укрепляющее нашу уверенность в правильном направлении. Постепенно мы становимся «местным» жителем, излазившим все окрестности и можем с уверенностью ориентироваться.

Первая прошивка — это сильный импульс на всю оставшуюся работу. Второй этап это формирование механизма шаблонов и всей структуры меню. Далее идут много мелких задач: клавиатура, измерения, управление балансиром, динамиком, вентилятором. Далее налаживание связи с большим компьютером и управление силовыми транзисторами. Формирование алгоритмов зарядки — это предпоследний и самый логически сложный этап.

Для осуществления настройки ЗУ будем использовать большой компьютер. Сложные экстраполяции при подборке коэффициентов и вписывание их в соответствующее место прошивки, все это сделаем через большой компьютер. Тогда нет необходимости занимать в микропрограмме место под сложный интерфейс и расчеты. Это экономия.

Ни одна программа не бывает без ошибок. Чем больше программа, тем больше взаимосвязей. Программист должен продумать все возможные дорожки по которым будет бегать программа. Предусмотреть все невозможно. Некоторые ошибки всплывают через несколько лет, казалось бы безошибочной работы. Продумать работу программы если все идет хорошо — это полдела. Нужно также продумать что будет делать программа при возникновении непредвиденной ошибки «exception» в любой ее части. Эти два полдела — это как артериальные и венозные древовидные системы проникающие друг в друга и образующие единый организм. В нашем ЗУ непредвиденные ошибки это в основном ошибочные действия пользователя или неожиданная реакция аккумулятора.

Ни один программист не дает гарантий относительно своей программы. Несмотря на все желание и старание, предусмотреть все жизненные ситуации нельзя, поэтому все программы продаются и распространяются «как есть». Вся ответственность ложится на

пользователя. А ответственность велика. ЗУ и аккумулятор могут вспыхнуть как огненный шар. «Талантливые» пользователи могут довести до такого результата и заводское ЗУ известного производителя.

К сожалению, у ЗУ нет ушей, носа, глаз и рук, а есть только датчик температуры, который пользователь забыл подключить также как и балансир, а по общему напряжению и току аккумулятор дает похожие на правду анализы. Как можно предусмотреть все? Ну может форма графика что-то подскажет, но как ее анализировать? Смотреть минимумы и максимумы на первой и второй производной? Не круто ли для 32 килобайт памяти в которую всунуто два канала, балансиры и большой экран с большим количеством пользовательского текста? Постараемся. Посмотрим что выйдет.

### **1 Шаг. Запуск интерфейса ЗУ.**

В статье «Как работает ЗУ» уже отмечалось, что при включении питания силовые транзисторы могут открыться и устроить КЗ, если не побеспокоиться об управляющих сигналах на процессоре. И, хотя, мы делали схему, чтобы транзисторы были закрыты, когда процессор в Z-состоянии (все ножки в состоянии высокого сопротивления), после включения процессора сигналы могут быть любыми. Чтобы избежать КЗ, первым делом микропрограмма должна выставить безопасную конфигурацию на ногах процессора.

Самая первая подпрограмма — подпрограмма инициализации. Берем нашу табличку: «**Описание процессора АТ Mega32**» и педантично выставляем на ногах процессора нужное направление и величину сигнала. При написании этой подпрограммы придется постоянно заглядывать в заводское описание процессора АТ MEGA 32. Обращаю ваше внимание: в некоторых переводах на русский язык есть ошибки. Сам на них нарывался, так что лучше смотреть английский текст первоисточника.

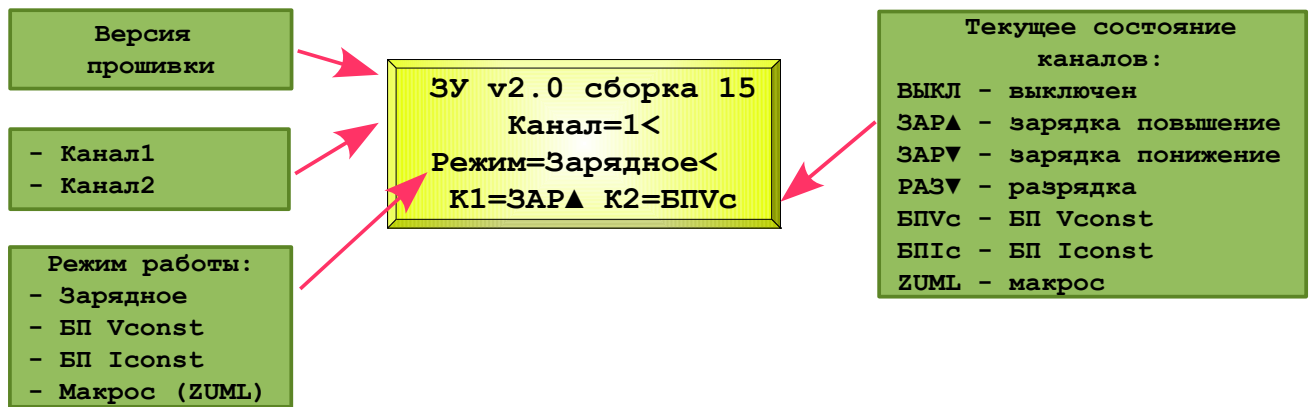
### **2 Шаг. Первая прошивка.**

Первая прошивка выполняет всего две задачи: инициализация всех ног процессора, программирование всех счетчиков, таймеров и прерываний — это раз; и, во вторых — испытание механизма вывода видеопамати на ЖКИ. [Прошивка](#) заработала и, тем самым, можно успокоиться и двигаться вперед.

### **3 Шаг. Придумываем меню.**

Теперь надо придумать меню: удобное, оригинальное, информативное. Все изыски шаблонов будут зависеть от меню. Весь интерфейс будет вращаться вокруг меню. Список слов и фраз будет построен из проекта меню. Способы отображения и обновления зависят от меню. Вот что удалось на фантазировать:

## Шаблон 0 — шаблон главного экрана при включении.



На главном экране отображается название зарядного и состояние «запущенности» каналов (коротко). Главный экран позволяет запустить или остановить выбранный канал в одном из 4х режимов (зарядное, БП Vconst, БП Iconst, макрос).

Каждый режим имеет свой шаблон для задания параметров запуска канала в этом режиме.

Редактируемое значение будет моргать или подсвечиваться особым образом.

При включении ЗУ с нажатой кнопкой ОК, можно будет войти в режим настроек.

<pre>Канал1      &gt;Старт? Задача=Зарядка&lt; Тип=LiPo&lt;   S (шт) =5&lt; Скорость=Быстро&lt;</pre>	<p><b>Шаблон 1 — запуск зарядного</b></p> <p>Шаблон запуска зарядки и разрядки позволяет выбрать функцию ЗУ (зарядка, тренировка, хранение), тип аккумулятора, количество банок и скорость зарядки.</p>
<pre>Канал2      &gt;Старт? Режим=БП Vconst V (В) =14.3&lt;</pre>	<p><b>Шаблон 2 — запуск БП Vconst</b></p> <p>Шаблон запуска БПVс выбирает напряжение которое будет поддерживаться всеми доступными методами. т.е. в зависимости от нагрузки ЗУ будет, менять скаженность, стараться поддерживать напряжение на заданном уровне.</p>
<pre>Канал2      &gt;Старт? Режим=БП Iconst&lt; I (А) =4.1&lt;</pre>	<p><b>Шаблон 3 — запуск БП Iconst</b></p> <p>Шаблон запуска БПс выбирает ток который будет поддерживаться постоянным независимо от нагрузки, пока это возможно.</p>
<pre>Канал2      &gt;Старт? Режим=МАКРО&lt; Прог=11&lt;</pre>	<p><b>Шаблон 4 — запуск МАКРО</b></p> <p>Шаблон запуска макроса выбирает выполняемую программу. Программа это набор однобайтовых инструкций, задающих напряжение или ток в зависимости от имеющихся измеренных параметров ЗУ.</p>
<pre>Канал1=РАЗ▼ 01:12:12 I (А) =3.5   P (W) =74.1 V (В) =14.3  Tc (°C) =50 E (мА) =12345  T (°C) =34</pre>	<p><b>Шаблон 5 — работа Зарядное</b></p> <p>Во время работы ЗУ отображаются различные измеренные параметры. Перед остановкой все параметры запоминаются для вывода результатов работы.</p>
<pre>Канал2=ЗАР▲ 00:20:23 Режим=БП Vconst V (В) =14.3&lt;  Tc (°C) =50 I (А) =3.0    T (°C) =34</pre>	<p><b>Шаблон 6 — работа БП Vconst</b></p>
<pre>Канал2=ЗАР▲ 00:20:23 Режим=БП Iconst I (А) =4.1&lt;  Tc (°C) =50 V (В) =14.3  T (°C) =34</pre>	<p><b>Шаблон 7 — работа БП Iconst</b></p>
<pre>Канал2=ЗАР▲ 00:20:23 Режим=МАКРО  Прог=11 V (В) =14.3   ФАЗА=12 I (А) =3.0    T (°C) =34</pre>	<p><b>Шаблон 8 — работа МАКРО (текущие показания)</b></p>
<pre>Канал2=МЗ▲ 00:20:23 T (°C) &lt;40   ФАЗА=12 if (V&gt;20.0) I (А) =0.1 if (V&lt;20.0) I (А) =0.0</pre>	<p><b>Шаблон 9 — работа МАКРО (текущая фаза условие)</b></p>
<pre>K1=РАЗ▼    &gt;V4 (В) =3.5 V1 (В) =3.5  V5 (В) =3.5 V2 (В) =3.4&lt; V6 (В) =3.5 V3 (В) =3.4  V7 (В) =3.5</pre>	<p><b>Шаблон 10 — Балансир напряжения</b></p> <p>В случае использования балансира есть возможность посмотреть напряжение и заряд на каждой банке. Угловой скобкой показана отключенная банка при балансировке.</p>

<b>K1=PA3▼ &gt;E4=12345</b> <b>E1=12345 E5=12345</b> <b>E2=12345&lt; E6=12345</b> <b>E3=12345 E7=12345</b>	Шаблон 11 — Балансир заряд
<b>Канал1 01:12:12</b> <b>Режим=Зарядное</b> <b>Стоп?&lt;</b>	Шаблон 12 — Стоп
<b>Канал1 01:12:12</b> <b>Задача=Зарядка</b> <b>Успешное окончание.</b> <b>Еобщ(мАч)=12000</b>	Шаблон 13 — Результат отчет

#### 4 Шаг. Кодировем слова и шаблоны.

Записываем слова:

Код	Слово	Код	Слово	Код	Слово
0	ЗУ v2.0 сборка 15	25	LiPo	50	
1	Старт?	26	LiIo	51	
2	Стоп?	27	LiFe	52	
3	Режим=	28	LiS	53	
4	Задача=	29	I(A)=	54	
5	Тип=	30	V(B)=	55	
6	Скорость=	31	T(°C)=	56	
7	Зарядка	32	S(шт)=	57	
8	Тренировка 1ц	33	P(W)=	58	
9	Тренировка 2ц	34	Tc(°C)=	59	
10	Тренировка 3ц	35	E(Ач)=	60	Успешное окончание
11	Хранение	36	Еобщ(мАч)=	61	Останов по перегреву
12	Автозаряд	37	Выкл	62	Есть дельтапик
13	Зарядное	38	Зар▼	63	Ток обнулится
14	БП Vconst	39	Зар▲	64	Зашкал температуры
15	БП Iconst	40	Раз▼	65	Зашкал тока
16	Макро	41	БПVc	66	Зашкал напряжения
17	Медленно	42	БПIc	67	Аккумулятор неадекватный
18	Нормально	43	ZUML	68	Перегрев ЗУ
19	Быстро	44	K1=	69	Низкое питание
20	Прог=	45	K2=	70	Неверная полярность
21	Фаза=	46	Канал1	71	Время истекло
22	Pb	47	Канал2	72	Неверный режим
23	NiCa	48	V1(B)=	73	
24	NiMh	49	E1=	74	

Записываем коды функций шаблонов, чтобы охватить все придуманные возможности. Потом закодируем шаблоны. Далее написание второй прошивки.

Общие правила шаблонов:

- Печать всех элементов шаблона производится от текущей позиции.

- Текущая позиция меняется командами установки курсора (0-79).
- Конец слова определяется по коду 0.
- Конец шаблона определяется по коду 255.

Списки команд шаблонов:

Код	Параметры	Описание
0-79	-	Установка курсора от начала видеопамати 0-19(первая строка)20-29(вторая)...
81-87	-	«V1(B)=» - «V7(B)=»
91-97	-	«E1=» - «E7=»
100	I	Напечатать целое число из переменной №I [x] (номер канала)
101	I	Напечатать целое число из переменной №I [xx] (номер программы, номер фазы, температура)
102	I	Напечатать целое число из переменной №I [xxxx] (заряд банки в балансире)
103	I	Напечатать целое число из переменной №I [xxxxx] (заряд всего аккумулятора)
110	I	Напечатать из переменной №I [x.x] (ток и напряжение на одной банке)
111	I	Напечатать из переменной №I [x.xx] (ток и напряжение на одной банке)
112	I	Напечатать из переменной №I [xx.x] (общее напряжение)
113	I	Напечатать из переменной №I [xx.xx] (общее напряжение)
120	I	Напечатать время из переменной №I [xx:xx:xx]
130	N	Напечатать слово номер Nconst
140	P, N	Напечатать слово номер которого лежит в переменной P+Nconst

Код переменной	Тип	Описание (переменные относящиеся к каналу дают значение отображаемого канала)
0	long	Время работы ЗУ [xx:xx:xx]
1	BYTE	Tc - температура радиатора [xx] 0-99
2	BYTE	Номер отображаемого канала [x]
3	BYTE	Номер отображаемого шаблона
4	BYTE	Stad - стадия работы с шаблоном (выбор параметра на шаблоне для подсвечивания и изменения)
5	BYTE	Режим отображаемого канала (0 - Зарядное, 1- БП Vconst, 2 - БП Iconst, 3 - Макро)
6	BYTE	Текущее состояние канала (0 - не запущен, 1 - запущен)
7	BYTE	Задача (0 - Зарядка, 1 - тренировка 1ц, 2 - тренировка 2ц, 2 - тренировка 3ц, 4 - хранение)
8	BYTE	Тип (0 - Pb, 1 - NiCa, 2 - NiMh, 3 - LiPo, 4 - Lilon, 5 - LiFe, 6 - LiS)
9	BYTE	Количество банок (0-20) [xx]
10	BYTE	Скорость (0 - медленно, 1 - нормально, 2 - быстро)
11	long	Время работы канала после запуска [xx:xx:xx]
12	long	Время работы очередной фазы [xx:xx:xx]
13	BYTE	Коротко состояние канала (ВЫКЛ — выключен, ЗАР▲ - зарядка повышение, ЗАР▼ - зарядка понижение, РАЗ▼ - разрядка, БПVc - БП Vconst, БПc - БП Iconst, ZUML - макрос)
14	float	Vc - требуемое напряжение [x.xx] 0.1-25.5
15	float	Ic - требуемый ток [x.x] 0.1-9.9
16	BYTE	fVI - true-преимущество V, false-преимущество I
17	BYTE	Номер МАКРО-программы [xx] 0-99
18	BYTE	Номер фазы [xx] 0-99

19	float	I - измеренный ток [х.х]
20	float	V - измеренное напряжение [хх.х]
21	float	T - измеренная температура [хх] 0-99
22	float	E - рассчитанный заряд [ххххх]
23	float	P - рассчитанная мощность [хххх]
24	float	V1 - напряжение на банке балансира [х.х] 0-4.2в
25	float	V2 - напряжение на банке балансира [х.х] 0-4.2в
26	float	V3 - напряжение на банке балансира [х.х] 0-4.2в
27	float	V4 - напряжение на банке балансира [х.х] 0-4.2в
28	float	V5 - напряжение на банке балансира [х.х] 0-4.2в
29	float	V6 - напряжение на банке балансира [х.х] 0-4.2в
30	float	V7 - напряжение на банке балансира [х.х] 0-4.2в
31	float	E1 - заряд на банке балансира [хххх] мАч
32	float	E2 - заряд на банке балансира [хххх] мАч
33	float	E3 - заряд на банке балансира [хххх] мАч
34	float	E4 - заряд на банке балансира [хххх] мАч
35	float	E5 - заряд на банке балансира [хххх] мАч
36	float	E6 - заряд на банке балансира [хххх] мАч
37	float	E7 - заряд на банке балансира [хххх] мАч
38	BYTE	b7 b6 b5 b4 b3 b2 b1 b0 - выключенная банка(и) балансира
39	BYTE	Код сообщения об остановке канала
100	BYTE	Коротко состояние 1 канала
200	BYTE	Коротко состояние 2 канала



Канал1 >Старт?  
Задача=Зарядка<  
Тип=LiPo< S(шт)=5<  
Скорость=Быстро<

#### Шаблон 1 — запуск зарядного для ЖКИ (4x20)

Шаблон запуска зарядки и разрядки позволяет выбрать функцию ЗУ (зарядка, тренировка, хранение), тип аккумулятора, количество банок и скорость зарядки.

Кодировка шаблона (подробно):

0 - Поставить курсор в начало первой строки  
140, 2, 46- Вывести слово 0 «Канал» от текущего положения курсора  
14 - Встать на 14 позицию на 1 строке  
130, 1 - Вывести слово 1 «Старт?»  
20 - Поставить курсор в начало второй строки  
130, 4 - Вывести слово 4 «Задача=»  
140, 7, 7 - Вывести слово (содержимое переменной 7 плюс смещение 7)  
40 - Поставить курсор в начало третьей строки  
130, 5 - Вывести слово 5 «Тип=»  
140, 8, 22- Вывести слово (содержимое переменной 8 плюс смещение 22)  
52 - Встать на 12 позицию на 3 строке  
130, 32 - Вывести слово 32 «S(шт)=»  
101, 9 - Вывести содержимое переменной 9 (количество банок)  
60 - Поставить курсор в начало четвертой строки  
130, 6 - Вывести слово 5 «Скорость=»  
140, 10, 17- Вывести слово (содержимое переменной 10 плюс смещение 17)  
255 - Конец шаблона

Все шаблоны ищите в файле **hd44780\_20x4.h**

Ch1 CHRГ LiPo  
S=10 FAST Старт?

#### Шаблон 1 — запуск зарядного для ЖКИ (2x16)

Шаблон запуска зарядки и разрядки позволяет выбрать функцию ЗУ (зарядка, тренировка, хранение), тип аккумулятора, количество банок и скорость зарядки.

Кодировка шаблона (подробно):

0 - Поставить курсор в начало первой строки  
140, 2, 44- Вывести слово 0 «Ch»  
5 - Переместить курсор в 5 позицию  
140, 7, 7 - Вывести слово (содержимое переменной 7 плюс смещение 7)  
12 - Переместить курсор в 12 позицию  
140, 8, 22- Вывести слово (содержимое переменной 8 плюс смещение 22)  
16 - Поставить курсор в начало второй строки  
130, 32 - Вывести слово 33 «S=»  
101, 9 - Вывести содержимое переменной 9 (количество банок)  
21 - Переместить курсор в 5 позицию на 2 строке  
140, 10, 17- Вывести слово (содержимое переменной 10 плюс смещение 17)  
26 - Переместить курсор в 10 позицию на 2 строке  
130, 1 - Вывести слово 2 «Старт?»  
255 - Конец шаблона

Подробнее смотрите файл **hd44780\_16x2.h**

В случае использования ЖКИ (16x2), меняется логика движения по меню - **menu\_16x2.cpp**

## Измерения (АЦП)

Теперь мы можем отображать любую информацию на экране через шаблоны или просто через подпрограммы вывода на экран. Следующий этап: необходимо обеспечить «интерактив». Пользователь нажимает на кнопки, а ЗУ исполняет. Т. е. следующий этап - это обеспечить работу клавиатуры.

Клавиатура у нас сделана через измерение напряжения на «KEY»:

<b>Off</b>	$V_{KEY} = 0\text{в}$	- Кнопки не нажаты.
<b>SA4 нажато</b>	$V_{KEY} = 0.45\text{в}$	- ОК
<b>SA3 нажато</b>	$V_{KEY} = 1.02\text{в}$	- ВВЕРХ (UP)
<b>SA2 нажато</b>	$V_{KEY} = 2.02\text{в}$	- ВНИЗ (DOWN)
<b>SA1 нажато</b>	$V_{KEY} = 2.5\text{в}$	- ОТМЕНА (CANCEL)

Вывод: следующий этап измерение аналоговых сигналов с помощью встроенного АЦП процессора. Заодно подготовим все необходимое для проведения всех измерений в программе.

Попробуем собрать воедино и обдумать что и как надо измерять:

Сигнал	Название	Точность дин./ст.	Периодичность	Усреднение дин./ст.	Комментарий
1IC	Ток зарядки 1к	$\pm 0.1$	5 раз в секунду	8	Необходимо отслеживать ток в динамике. Точность не критична.
1V	Напряжение 1к	$\pm 0.1$ $\pm 0.05$	5 раз в секунду	8 32	Необходимо отслеживать напряжение в динамике. Точность не критична. В статике точность нужна.
2IC	Ток зарядки 2к	$\pm 0.1$	5 раз в секунду	8	Необходимо отслеживать ток в динамике. Точность не критична.
2V	Напряжение 2к	$\pm 0.1$ $\pm 0.05$	5 раз в секунду	8 32	Необходимо отслеживать напряжение в динамике. Точность не критична. В статике точность нужна.
1BAL	8 напряжений 1к	$\pm 0.01$	1 раз в 5 секунд	32	Нужна высокая точность, но достаточно редко.
2BAL	8 напряжений 2к	$\pm 0.01$	1 раз в 5 секунд	32	Нужна высокая точность, но достаточно редко.
IzM IZ0	Напряжение на минусе 1к	$\pm 0.1$	5 раз в секунду	8	Для определения неверной полярности.
IzM IZ1	Напряжение на минусе 2к	$\pm 0.1$	5 раз в секунду	8	Для определения неверной полярности.
IzM IZ2	Напряжение питания	$\pm 0.1$	5 раз в секунду	8	Точность не нужна. Для определения недостаточного или завышенного напряжения питания.
IzM 1ID	Ток разрядки 1к	$\pm 0.1$	5 раз в секунду	8	Ток разрядки ограничен настройкой схемы. Точность не нужна.
IzM 2ID	Ток разрядки 2к	$\pm 0.1$	5 раз в секунду	8	Ток разрядки ограничен настройкой схемы. Точность не нужна.
IzM 1T	Температура 1к	$\pm 0.1$	1 раз в 5 секунд	8	Величина медленно меняющаяся. Точность не нужна.
IzM 2T	Температура 2к	$\pm 0.1$	1 раз в 5 секунд	8	Величина медленно меняющаяся. Точность не нужна.
IzM CT	Температура схемы	$\pm 0.1$	1 раз в 5 секунд	8	Величина медленно меняющаяся. Точность не нужна.
KEY	Клавиатура	$\pm 0.1$	5 раз в секунду	1	Требуется 3 измерения подряд из одного диапазона, для фиксации нажатия конкретной кнопки.

Помним базовые понятия относительно измерений АЦП:

1. Все измерения делаются относительно земли, кроме измерений балансира (B3-B7), которые делаются относительно минуса этой же банки.
2. Все измеряемые величины (все токи, градусы, нажатия кнопок и все напряжения) — это напряжения от 0 до 2.5в причем желательно ближе к 2.5в, тогда точность измерений выше.
3. Максимальное напряжение 2.5в, превращается в число от 0 до 1023 или 0x03FF.
4. Точность измерения  $2.5\text{в}/1024 \approx 0.0025$ . При измерении 5в, точность будет 0.005в. При измерении 30в точность будет 0.03. С учетом помех по питанию, точность будет хуже. Ситуацию можно попытаться исправить за счет многократного измерения и потом усреднения. Если помехи будут «белые», т. е. будут играть одинаково и вверх и вниз, то за счет усреднения мы их влияние уберем. Если помехи будут играть всегда в плюс или всегда в минус, то такое действие помех мы тоже уберем. А, вот, если помехи будут плавать - это плохо.
5. АЦП процессора реально состоит только из одного АЦП к которому подключается по очереди 8 ног процессора (ADC0-ADC7).
6. Каждый из сигналов ADC4 (1BAL), ADC5(2BAL), ADC6(IzM) еще ветвится (коммутируется) на 8

сигналов. Причем после переключения коммутатора надо выждать некоторое время для установления сигнала на измеряемой ноге процессора. Время приблизительно равно  $T \approx 10 * R_{18} C_{16} = 10 * 300 \text{ohm} * 370 \text{pF} = 10 * 300 * 370 * 10^{-12} \approx 0.000001 \text{s}$ . В принципе, можно не ждать. Посмотрим...

7. АЦП процессора осуществляет преобразование на частоте  $16 \text{MГц} / 128 = 125 \text{kГц}$  за 13-15 тактов, т. е. приблизительно 8000 измерений в секунду.
8. Большая часть наших измерений не требуют таких скоростей. Поэтому будем придерживаться следующего алгоритма измерений:
  - 5 раз в секунду измеряем клавиатуру, итоговые токи и напряжения обоих каналов. Запуск серии измерений происходит из счетчика секунд. Каждое измерение запускает следующее измерение, пока не закончится серия.
  - Измерения балансира запускаются из программы управления зарядкой после небольшого предварительного разряда и остановки обоих силовых. При этом также измеряются итоговые напряжения высокой точности.

Для обеспечения выбранных измерений необходимо правильно настроить АЦП процессора. Для этого добавим в файл «init.cpp» следующий код:

## Описание процессора AT Mega32

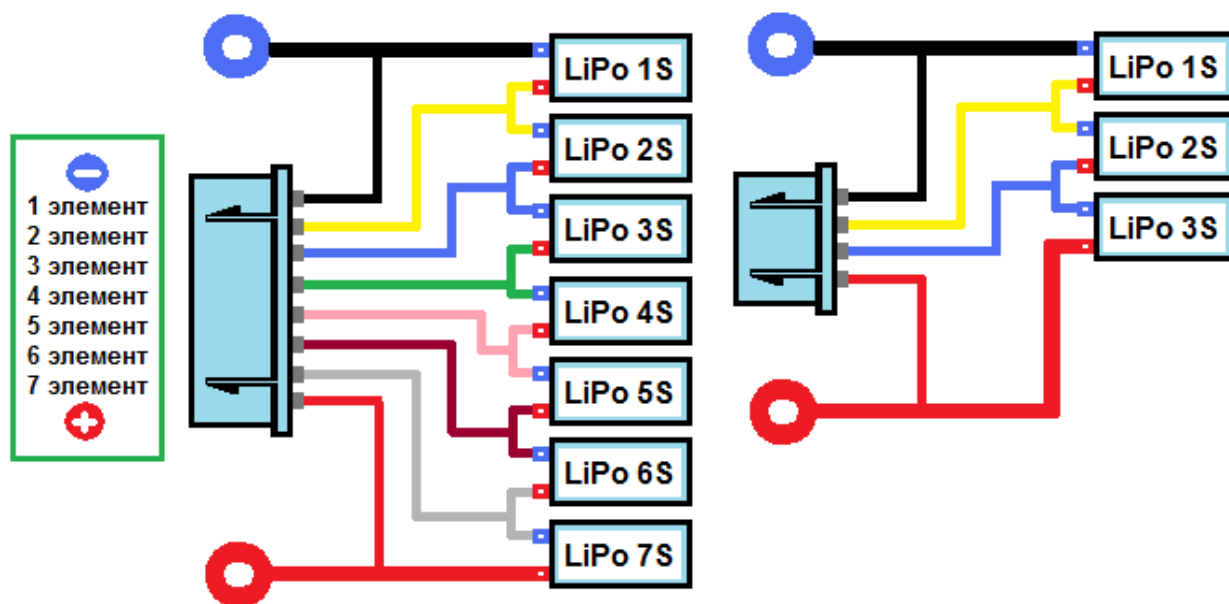
Нога	Название	Напр.	Активный уровень	Название по схеме	Описание
1	B5(MOSI)	OUT/in	UP	MU0	1. На разъем программирования ISP. 2. Адрес для коммутации напряжений с балансира канала 1 на АЦП процессора ADC4/1BAL. 3. Адрес для коммутации напряжений с балансира канала 2 на АЦП процессора ADC5/2BAL. 4. Адрес для коммутации прочих напряжений для измерения на АЦП процессора ADC6/IZM.
2	B6(MISO)	OUT/out	UP	MU1	
3	B7(SCK)	OUT/in	UP	MU2	
4	/RESET	in	down	RESET	На разъем ISP. Программирование процессора.
5	VCC	POWER	+5в	VCC	Питание процессора.
6	GND	POWER	0в	GND	Земля процессора.
7	XTAL2	-	-	XTAL2	Внешний кварцевый резонатор 16 МГц.
8	XTAL1	-	-	XTAL1	Внешний кварцевый резонатор 16 МГц.
9	D0(RXD)	IN	UP	RXD	Последовательное чтение информации, поступающей через USB разъем с компьютера (команды для выполнения).
10	D1(TXD)	OUT	UP	TXD	Последовательная отправка информации через USB на компьютер (данные о ходе зарядки/разрядки)
11	D2(INT0)	OUT	↑	RCLK	Для управления балансирами обоих каналов. Перенос битов из сдвигового регистра в регистр хранения и отражение на выходах 74HC595D.
12	D3(INT1)	OUT	DOWN	1DS	Для управления балансиром 1 канала. Последовательный вход для сдвигового регистра 74HC595D.
13	D4(OC1B)	OUT	UP	2PWM	ШИМ 2 канала для понижающего, повышающего, разряжающего режимов.
14	D5(OC1A)	OUT	UP	1PWM	ШИМ 1 канала для понижающего, повышающего, разряжающего режимов.
15	D6(ICP)	OUT	↑	CLK	Для управления балансирами обоих каналов. Сдвиг битов в сдвиговом регистре 74HC595D без отражения на выходах.
16	D7(OC2)	OUT	UP	BRIGHT	ШИМ, задающий яркость подсветки ЖКИ.
17	VCC	POWER	+5в	VCC	Питание процессора.
18	GND	POWER	0в	GND	Земля процессора.
19	C0(SCL)	OUT	UP	DB7	D7 бит шины данных ЖКИ
20	C1(SDA)	OUT	UP	DB6	D6 бит шины данных ЖКИ
21	C2(TCK)	OUT	UP	DB5	D5 бит шины данных ЖКИ
22	C3(TMS)	OUT	UP	DB4	D4 бит шины данных ЖКИ
23	C4(TDO)	OUT	↑	E	E строб передачи данных ЖКИ
24	C5(TDI)	OUT	-	RS	RS=0 передача команды на ЖКИ RS=1 передача данных на ЖКИ
25	C6(TOSC1)	OUT	UP	COOL	Включить вентилятор охлаждения радиатора
26	C7(TOSC2)	OUT		2DS	Для управления балансиром 2 канала. Последовательный вход для сдвигового регистра 74HC595D.
27	AVCC	POWER	5в	AVCC	Питание измерительной схемы процессора. Повышенная помехоустойчивость.
28	GND	POWER	0в	GND	Измерительная земля.
29	AREF	IN	2.5в	AREF	Измерительная опора. Все измерения производятся относительно 2.5в с градацией в 1024 уровня. Точность ~ 0.0025в. За счет 16 измерений и усреднения точность должна быть несколько выше.
30	A7(ADC7)	IN	-	KEY	0в – Кнопки не нажаты. (SA4)0.45в — ОК (SA3)1.02в — ВВЕРХ (UP) (SA2)2.02в — ВНИЗ (DOWN) (SA1)2.5в — ОТМЕНА (CANCEL)
31	A6(ADC6)	IN	<2.5в	IZM	Измерение всяких напряжений: 1. Минус 1 канала грубо. 2. Минус 2 канала грубо. 3. 12в напряжение питания. 4. Ток разрядки 1 канала. 5. Ток разрядки 2 канала. 6. Температура 1 канала. 7. Температура 2 канала. 8. Температура радиатора 3У.
32	A5(ADC5)	IN	<2.5в	2BAL	Измерение напряжений с балансира 2 канала 1. Минус 2 канала точно. 2. Плюс 1 банки 2 канала относит. земли. 3. Плюс 2 банки 2 канала относит. земли. 4. Плюс 3 банки 2 канала относит. плюса 2 банки.

					5. Плюс 4 банки 2 канала относит. плюса 3 банки. 6. Плюс 5 банки 2 канала относит. плюса 4 банки. 7. Плюс 6 банки 2 канала относит. плюса 5 банки. 8. Плюс 7 банки 2 канала относит. плюса 6 банки.
33	A4(ADC4)	IN	<2.5в	1BAL	Измерение напряжений с балансира 2 канала 1. Минус 1 канала точно. 2. Плюс 1 банки 1 канала относит. земли. 3. Плюс 2 банки 1 канала относит. земли. 4. Плюс 3 банки 1 канала относит. плюса 2 банки. 5. Плюс 4 банки 1 канала относит. плюса 3 банки. 6. Плюс 5 банки 1 канала относит. плюса 4 банки. 7. Плюс 6 банки 1 канала относит. плюса 5 банки. 8. Плюс 7 банки 1 канала относит. плюса 6 банки.
34	A3(ADC3)	IN	<2.5в	2V	Измерение напряжения 2 канала.
35	A2(ADC2)	IN	<2.5в	2IC	Измерение тока зарядки 2 канала.
36	A1(ADC1)	IN	<2.5в	1V	Измерение напряжения 1 канала.
37	A0(ADC0)	IN	<2.5в	1IC	Измерение тока зарядки 1 канала.
38	VCC	POWER	5в.	VCC	Питание процессора.
39	GND	POWER	0в.	GND	Земля процессора.
40	B0(XCK/T0)	OUT	UP	2A1	Режим 2 канала.
41	B1(T1)	OUT	UP	2A0	2A1 2A0 00 - бездействие 01 - понижающий режим зарядка 10 - повышающий режим зарядка 11 - разрядка
42	B2(AIN0/INT2)	OUT	UP	1A1	Режим 1 канала.
43	B3(AIN1/OC0)	OUT	UP	1A0	1A1 1A0 00 - бездействие 01 - понижающий режим зарядка 10 - повышающий режим зарядка 11 - разрядка
44	B4(/SS)	OUT	UP	BUZZ	Пищалка

### Прочие обозначения на схеме

Название по схеме	Активный уровень	Описание
+12	+11в <=> +15в	Напряжение питания схемы
1B0	1BT- *(1/2)в	Канал 1. Напряжение на минусе 1 канала относительно земли
1B1	(1BT- + 4.2)*100.68/200.68в	Канал 1. Напряжение на плюсе 1 банки относительно земли
1B2	(1BT- + 2*4.2)*68.62/268.62в	Канал 1. Напряжение на плюсе 2 банки относительно земли
1B3	Если на банке 4.2в то 2.11в	Канал 1. Напряжение на плюсе 3 банки относительно плюса 2 банки
1B4	Если на банке 4.2в то 2.11в	Канал 1. Напряжение на плюсе 4 банки относительно плюса 3 банки
1B5	Если на банке 4.2в то 2.11в	Канал 1. Напряжение на плюсе 5 банки относительно плюса 4 банки
1B6	Если на банке 4.2в то 2.11в	Канал 1. Напряжение на плюсе 6 банки относительно плюса 5 банки
1B7	Если на банке 4.2в то 2.11в	Канал 1. Напряжение на плюсе 7 банки относительно плюса 6 банки
1BB3	(1BT- +4.2*3)/2	Канал 1. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
1BB4	(1BT- +4.2*4)/2	Канал 1. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
1BB5	(1BT- +4.2*5)/2	Канал 1. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
1BB6	(1BT- +4.2*6)/2	Канал 1. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
1BB7	(1BT- +4.2*7)/2	Канал 1. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
1BL1	1BT- +4.2	Канал 1. Разъем балансира плюс 1 банки.
1BL2	1BT- +4.2*2	Канал 1. Разъем балансира плюс 2 банки.
1BL3	1BT- +4.2*3	Канал 1. Разъем балансира плюс 3 банки.
1BL4	1BT- +4.2*4	Канал 1. Разъем балансира плюс 4 банки.
1BL5	1BT- +4.2*5	Канал 1. Разъем балансира плюс 5 банки.
1BL6	1BT- +4.2*6	Канал 1. Разъем балансира плюс 6 банки.
1BL7	1BT- +4.2*7	Канал 1. Разъем балансира плюс 7 банки.
1BT-	Если ток 5А, то 0,05 Ом*5А	Канал 1. Минус батареи. Разъем балансира минус 1 банки.
1BT+	1BT- +4.2*х	Канал 1. Закорочен с плюсом старшей банки
1EN	DOWN	Канал 1. Разрешить присоединение минуса батареи к земле
1I	<2.5в	Канал 1. Ток зарядки
1ID	<2.5в	Канал 1. Ток разрядки
1IE	<2.5в	Канал 1. Ошибка тока зарядки. Превышение включает защиту
1T	<2.5в	Канал 1. Температура
1Y3	DOWN	Канал 1. Разрешение разрядки
2B0	1BT- *(1/2)в	Канал 2. Напряжение на минусе 2 канала относительно земли
2B1	(1BT- + 4.2)*100.68/200.68в	Канал 2. Напряжение на плюсе 1 банки относительно земли
2B2	(1BT- + 2*4.2)*68.62/268.62в	Канал 2. Напряжение на плюсе 2 банки относительно земли
2B3	Если на банке 4.2в то 2.11в	Канал 2. Напряжение на плюсе 3 банки относительно плюса 2 банки
2B4	Если на банке 4.2в то 2.11в	Канал 2. Напряжение на плюсе 4 банки относительно плюса 3 банки
2B5	Если на банке 4.2в то 2.11в	Канал 2. Напряжение на плюсе 5 банки относительно плюса 4 банки
2B6	Если на банке 4.2в то 2.11в	Канал 2. Напряжение на плюсе 6 банки относительно плюса 5 банки
2B7	Если на банке 4.2в то 2.11в	Канал 2. Напряжение на плюсе 7 банки относительно плюса 6 банки
2BB3	(1BT- +4.2*3)/2	Канал 2. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
2BB4	(1BT- +4.2*4)/2	Канал 2. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
2BB5	(1BT- +4.2*5)/2	Канал 2. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
2BB6	(1BT- +4.2*6)/2	Канал 2. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
2BB7	(1BT- +4.2*7)/2	Канал 2. Промежуточный сигнал на + входе ОУ (в режиме вычитателя)
2BL1	1BT- +4.2	Канал 2. Разъем балансира плюс 1 банки.
2BL2	1BT- +4.2*2	Канал 2. Разъем балансира плюс 2 банки.
2BL3	1BT- +4.2*3	Канал 2. Разъем балансира плюс 3 банки.
2BL4	1BT- +4.2*4	Канал 2. Разъем балансира плюс 4 банки.
2BL5	1BT- +4.2*5	Канал 2. Разъем балансира плюс 5 банки.
2BL6	1BT- +4.2*6	Канал 2. Разъем балансира плюс 6 банки.
2BL7	1BT- +4.2*7	Канал 2. Разъем балансира плюс 7 банки.
2BT-	Если ток 5А, то 0,05 Ом*5А	Канал 2. Минус батареи. Разъем балансира минус 1 банки.
2BT+	1BT- +4.2*х	Канал 2. Закорочен с плюсом старшей банки
2EN	DOWN	Канал 2. Разрешить присоединение минуса батареи к земле
2I	<2.5в	Канал 2. Ток зарядки
2ID	<2.5в	Канал 2. Ток разрядки
2IE	<2.5в	Канал 2. Ошибка тока зарядки. Превышение включает защиту
2T	<2.5в	Канал 2. Температура
2Y3	DOWN	Канал 2. Разрешение разрядки
CT	<2.5в	Температура радиатора внутри ЗУ
IZ0	1BT- *18/(200+18)в	Канал 1. Грубо измерение минуса батареи при переплюсовке включения
IZ1	2BT- *18/(200+18)в	Канал 2. Грубо измерение минуса батареи при переплюсовке включения
IZ2	<<+12>>*2.7/(18+2.7)в	Сигнал питания +12в для измерения

## Подключение к балансиру



## Заряжаемые аккумуляторы

Тип	Напряжение одного элемента (банки)	Зарядка	Разрядка
NiCd/NiMH	1.2V	1С-2С в зависимости от свежести дельтапик 0.2V	NiCd до 0.85V NiMH до 1V
LiIo	3.6V	1С или менее до 4.1V	до 2.5V или более
LiPo	3.7V	1С или менее до 4.2V	до 3.0V или более
LiFe	3.3V	4С или менее до 3.6V	до 2.0V или более
Pb (Lead-acid)	2.0V	0.4С или менее до 2.46V	до 1.75V или более