



AVR450: 为 SLA、NiCd、NiMH 和 Li-Ion 电池设计的充电器

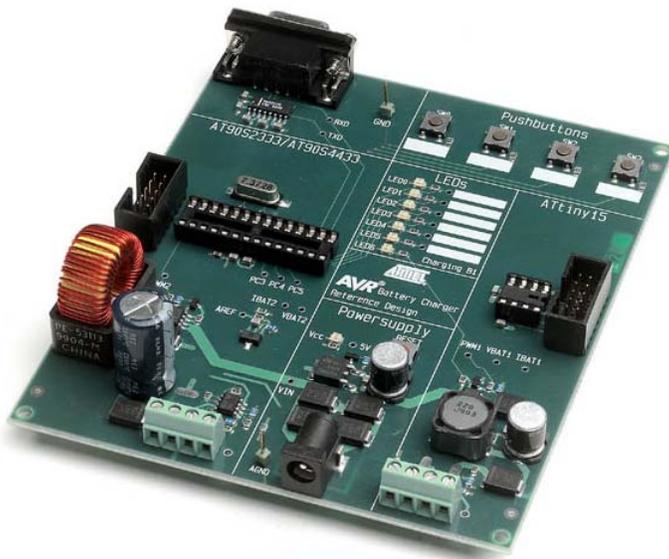
特点

- 完整的电池充电器设计方案
- 模块化的“C”源代码和极紧凑的汇编代码
- 低成本
- 支持多数电池类型
- 快速充电算法
- 10 位 ADC 实现高精度测量
- 可选的串行接口
- 充电参数极易修改
- 片内 EEPROM 可用于存储电池信息

说明

本参考设计完全实现了电池充电器设计的最新技术，可以对各种流行的电池类型进行快速充电而无须修改硬件，从而围绕单个硬件平台实现一个完整的充电器产品系列。只需要将新的充电算法通过 ISP 下载到处理器的 FLASH 存储器就可以得到新的型号。很显然，这种方法可以大大缩短新产品上市的时间，而且只需要库存一种硬件。本设计提供完整的适合 SLA、NiCd、NiMH 和 Li-Ion 电池的库函数。

Figure 1. 电池充电器参考设计



8 位 AVR[®]
微处理器

应用实例

本文是英文数据手册的中文翻译，其目的是方便中国用户的阅读。它无法自动跟随原稿的更新，同时也可能存在翻译上的错误。读者应该以英文原稿为参考以获得更准确的信息。

Rev. 1659A-AVR-03/02



参考设计实现了两个充电器，分别由高端产品 AT90S4433 和高集成度、低成本的 8 引脚器件 ATtiny15 构成。当然，也可以用任意一款带 A/D 转换器、PWM 输出、具有足够程序存储器的 AVR 器件来实现电池充电器。

介绍

随着越来越多的手持式电器的出现，对高性能、小尺寸、重量轻的电池充电器的需求也越来越大。电池技术的持续进步也要求更复杂的充电算法以实现快速、安全的充电。因此需要对充电过程进行更精确的监控，以缩短充电时间、达到最大的电池容量，并防止电池损坏。AVR 已经在竞争中领先了一步，被证明是下一代充电器的完美控制芯片。

Atmel AVR 微处理器是当前市场上能够以单片方式提供 Flash、EEPROM 和 10 位 ADC 的最高效的 8 位 RISC 微处理器。由于程序存储器为 Flash，因此可以不用象 MASK ROM 一样，有几个软件版本就库存几种型号。Flash 可以在发货之前再进行编程，或是在 PCB 贴装之后再通过 ISP 进行编程，从而允许在最后一分钟进行软件更新。

EEPROM 可用于保存标定系数和电池特性参数，如保存充电记录以提高实际使用的电池容量。10 位 A/D 转换器可以提供足够的测量精度，使得充好后的容量更接近其最大容量。而其他方案为了达到此目的，可能需要外部的 ADC，不但占用 PCB 空间，也提高了系统成本。

AVR 是目前唯一的针对“C”这样的高级语言而设计的 8 位微处理器。AT90S4433 参考设计就是用“C”写的，说明用高级语言进行软件设计是多么的简单。C 代码似的此设计很容易进行调整以适合当前和未来的电池。而 ATtiny15 参考设计则是用汇编语言写的，以获得最大的代码密度。

工作原理

电池充电是通过逆向化学反应将能量存储到化学系统里实现的。由于使用的化学物质的不同，电池有自己的特性。设计充电器时要仔细了解这些特性以防止过度充电而损坏电池。

AVR 8 位 RISC MCU

参考设计包括两个独立的充电器。一个使用的是 AT90S4433，另一个使用的是 ATtiny15。AT90S4433 设计用于说明用 C 实现电池充电器的高效性，而 ATtiny15 设计则是目前市场上可以得到的集成度最高、成本最低的电池充电器。AT90S4433 可通过 UART 与 PC 接口以监控电池电压和温度。表 1 说明了两个设计的不同。

Table 1. 设计差异

	AT90S4433	ATtiny15
编程语言	C	汇编
代码大小	1.5K 字节	<350 字节
电流测量	需外部运放调节增益	片内具有差分增益调节功能
PWM 频率	14 kHz, 8 位	100 kHz, 8 位
时钟	外部晶体, 7.3 MHz	片内经过标定的 RC 振荡器, 1.6 MHz
串行接口	Yes	No
ISP	Yes	Yes

电池技术

现代消费类电器主要使用如下四种电池：

- 密封铅酸电池 (SLA)
- 镍镉电池 (NiCd)
- 镍氢电池 (NiMH)
- 锂电池 (Li-Ion)

在正确选择电池和充电算法时需要了解这些电池的背景知识。

密封铅酸电池 (SLA)

密封铅酸电池主要用于成本比空间和重量更重要的场合，如 UPS 和报警系统的备份电池。SLA 电池以恒定电压进行充电，辅以电流限制以避免在充电过程的初期电池过热。只要电池单元电压不超过生产商的规定（典型值为 2.2V），SLA 电池可以无限制地充电。

镍镉电池 (NiCd)

NiCd 电池目前使用得很普遍。它的优点是相对便宜，易于使用；缺点是自放电率比较高。典型的 NiCd 电池可以充电 1,000 次。失效机理主要是极性反转。在电池包里第一个被完全放电的单元会发生反转。为了防止损坏电池包，需要不间断地监控电压。一旦单元电压下降到 1.0V 就必须停机。NiCd 电池以恒定电流的方式进行充电。

镍氢电池 (NiMH)

在轻重量的手持设备中如手机、手持摄像机，等等镍氢电池是使用最广的。这种电池的容量比 NiCd 的大。由于过充电会造成 NiMH 电池的失效，在充电过程中进行精确地测量以在合适的时间停止是非常重要的。和 NiCd 电池一样，极性反转时电池也会损坏。

NiMH 电池的自放电率大概为 20%/月。和 NiCd 电池一样，NiMH 电池也为恒定电流充电。

锂电池 (Li-Ion)

和本文中所述的其他电池相比，锂电池具有最高的能量 / 重量比和能量 / 体积比。锂电池以恒定电压进行充电，同时要有电流限制以避免在充电过程的初期电池过热。当充电电流下降到生产商设定的最小电流时就要停止充电。过充电将造成电池损坏，甚至爆炸。

电池的安全充电

现代的快速充电器 (即电池可以在小于 3 个小时的时间里充满电, 通常是一个小时) 需要能够对单元电压、充电电流和电池温度进行精确地测量, 在充满电的同时避免由于过充电造成的损坏。

充电方法

SLA 电池和锂电池的充电方法为恒定电压法要限流; NiCd 电池和 NiMH 电池的充电方法为恒定电流法, 且具有几个不同的停止充电的判断方法。

最大充电电流

最大充电电流与电池容量 (C) 有关。最大充电电流往往以电池容量的数值来表示。例如, 电池的容量为 750 mAh, 充电电流为 750 mA, 则充电电流为 1C (1 倍的电池容量)。若涓流充电时电流为 C/40, 则充电电流即为电池容量除以 40。

过热

电池充电是将电能传输到电池的过程。能量以化学反应的方式保存了下来。但不是所有的电能都转化为了电池中的化学能。一些电能转化成了热能, 对电池起了加热的作用。当电池充满后, 若继续充电, 则所有的电能都将转化为电池的热能。在快速充电时这将使电池快速升温, 若不及时停止充电就会造成电池的损坏。因此, 在设计电池充电器时, 对温度进行监控并及时停止充电是非常重要的。

停止充电的判别方法

电池的不同应用场合及工作环境限制了对判断停止充电的方法的选择。有时候温度不容易测得，但可以测得电压，或者是其他情况。本文以电压变化率 ($-dV/dt$) 为基本的判断停止充电的方法，而以温度和绝对电压值为辅助和备份。但是本文所描述的硬件支持以下讲述的所有的方法。

t – 时间

这是决定何时停止充电的最简单的方法。通常用于快速充电时的后备方案。有时也作为普通充电 (14 - 16 小时) 方法的基本方案。适用于各种电池。

V – 电压

当电压超出上限时停止充电。通常与恒定电流充电配合使用。最大电流由电池决定，通常为 1C。为了防止充电时电流过大导致电池过热，此时电流限制是非常关键的。这个方法为锂电池的基本充电和停止方案。实际锂电池充电器往往在达到最大电压之后还继续进行第二阶段的充电，以达到 100% 的电池容量。对于 NiCd 电池和 NiMH 电池本方法可以作为后备的判断停止充电方案。

$-dV/dt$ – 电压变化率

这个判断停止充电的方法利用了负的电压变化率。对于某些类型的电池，当电池充满后继续充电将导致电压的下降。此时本方案就非常合适了。这个方法通常用于恒定电流充电，适用于对 NiCd 电池和 NiMH 电池的快速充电。

I – 电流

当充电电流小于某个预先设定的数值时停止充电。通常用于恒定电压充电法。适用于 SLA 电池和锂电池。

T – 温度

绝对温度可以作为 NiCd 电池和 NiMH 电池停止充电的依据，但是更适合于作为备份方案。温度超出设定值时任何电池都得停止充电。

dT/dt – 温度上升速率

快速充电时温度的变化率可以作为停止充电的依据。请参考电池生产商的规范 (NiCd 电池的典型值为 $1^{\circ}\text{C}/\text{min}$) – 适用于 NiCd 电池和 NiMH 电池。

DT – 超出环境温度的温度值

当电池温度和环境温度之差超过一定门限时需要停止充电。此方法可以作为 NiCd 电池和 SLA 电池停止充电的方案。在寒冷环境中充电时这个方法比绝对温度判定法更好。由于大多数系统往往只有一个温度探头，只好将充电之前的温度作为环境温度。

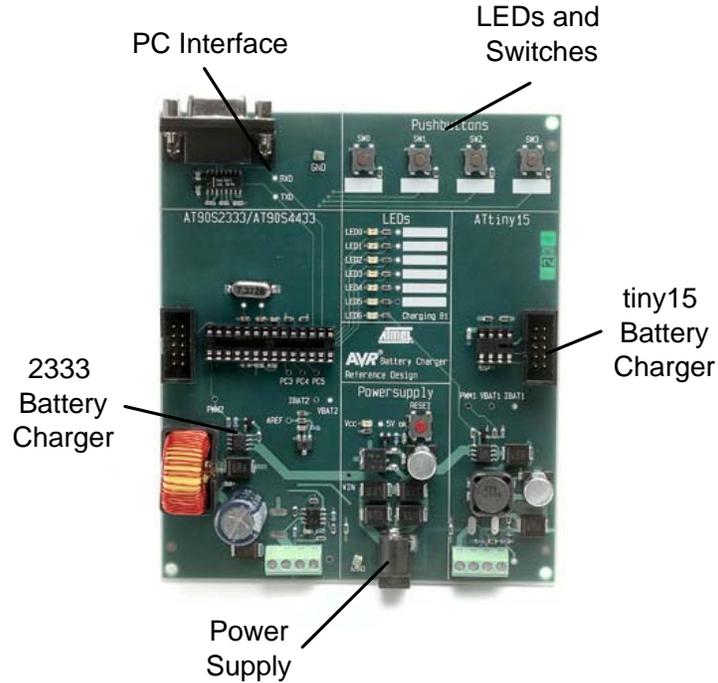
$dV/dt = 0$ – 零电压差

这个方法与 $-dV/dt$ 方法极其类似，而且在电压不会再升高的情况下更准确。适用于 NiCd 电池和 NiMH 电池。

硬件实现

参考设计包括两个完整的电池充电器设计。如图2所示，本设计可以分为5个主要的模块。

Figure 2. 电池充电器参考设计的主要模块



电源

包括模拟基准源、按键和 LED。输入电压由 D9 - D12 进行整流，然后通过 C13 进行滤波。整流后的输入电压可以在板上标有“ V_{IN} ”处测得。 V_{IN} 同时提供给 BUCK 转换器和 LM7805。LM7805 为单片机提供 5V 电源。此电压可以在标有“ V_{CC} ”的测试点处测得。若标有“5V OK”的 LED 亮则说明电源就绪。

PC 接口

与 AT90S4433 的 UART 相连，可用来获取电池充电时的数据。这些数据可以导入到 spreadsheet 以显示电池特性。进一步的信息请参阅应用手册“VR451: Data Logger for Battery Charger Reference Design”，以获取例子的源代码以及用于导入数据的 Microsoft® Excel spreadsheet。使用 ATtiny15 电池充电器时也可以用 AT90S4433 来获取数据。

LED 和按键

电路板上有几个 LED 和按键用于调试和监控。在目前的应用中只用到了几个。

- LED0: 与 AT90S4433 的 PortB0 连接，用于指示当前充电模式为快速模式还是涓流模式。
- LED1: 与 AT90S4433 的 PortB2 连接
- LED2: 与 AT90S4433 的 PortB3 连接
- LED3: 与 AT90S4433 的 PortB0 连接，用于指示系统出现了错误
- LED4: 无连接
- LED5: 无连接
- LED6: 与 ATtiny15 的 PortB1 连接，用于指示当前的 PWM 频率。
- V_{CC} Power: 指示当前的电源状态
- SW0: 与 AT90S4433 的 PortD4 连接，用于启动充电器
- SW1: 与 AT90S4433 的 PortD5 连接
- SW2: 与 AT90S4433 的 PortD6 连接

- SW3: 与 AT90S4433 的 PortD7 连接
- RESET: 复位程序以便从错误中恢复

ISP 接口

两个充电器都有 10 引脚的 ISP 插座。可以通过 AVRISP PC 软件对程序 FLASH 和 EEPROM 数据存储进行编程。

ATtiny15 及 100 kHz Buck 变换器

ATtiny15 的特性使其非常适合于设计电池充电器。片内的 100 kHz PWM 与 buck 变换器相连。高速开关频率以及高精度降低了对外围线圈和电容的要求。电路板上加入了测试点以便于监控 PWM 输出, 输入电压和输入电流。ATtiny15 具有内部增益调节功能以放大两个 ADC 引脚之间的差分信号, 从而避免了使用外部运放。充电电流流过 0.25W 的电阻, 并由两个 ADC 取得差分信号。充电器的电源在附录 2 里说明。

AT90S4433 及 14kHz Buck 变换器

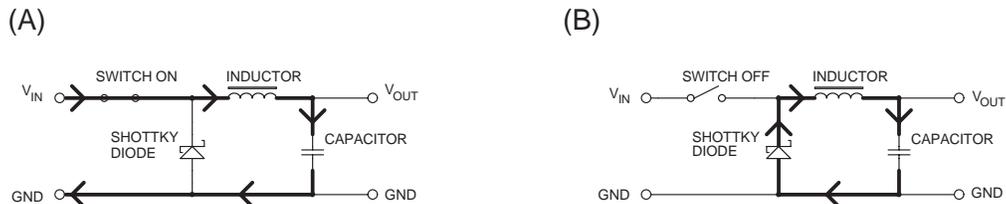
90S4433 型充电器使用了外部运放来放大电流信号, 以保证精度。充电器可以与 PC 进行通讯, 用来监控充电参数及调试充电算法。

充电器可以对 SLA、NiCd、NiMH 和 Li-Ion 电池进行充电。充电算法包括快充模式和涓流模式, 以最大限度减少充电时间, 获得最大容量。充电器的电源在附录 2 里说明。

Buck 变换器

AT90S4433 方案和 ATtiny15 方案中的 buck 变换器是类似的, 都包括一个由 AVR 通过双极性 NPN 晶体管驱动的 P 沟道 MOSFET 开关管。开关管与电感、二极管和电容相连 (参见 Figure 3)。一个额外的二极管用来防止在断电时电池向微处理器供电。当开关管接通时 (在下图中以一个开关表示), 电流以 Figure 3A 的方式流动。电容通过电感被充电 (电感也吸收了能量)。当开关管打开时 (Figure 3B), 电感试图保持电流, 从而导致电流流过二极管、电感和电容。这就是一个工作周期。如果减少占空比开通时间减少, 断开时间增加, 输出电压也将下降。反之输出电压将增加。在占空比为 50% 时 buck 变换器的效率最高。

Figure 3. Buck 变换器开关原理



电压基准

电压基准由 TL431 CPK 提供。\$A_{REF}\$ 通过电阻 R34 和 R10 设置。计算公式为：

$$A_{REF} = V_{REF} \left(1 + \frac{R_{34}}{R_{10}} \right) = 2.495 \left(1 + \frac{4K7}{10K} \right) = 3.67V$$

这个数值是高精度 (\$A_{REF}\$ 的数值小一些) 与高信噪比 (\$A_{REF}\$ 的数值大一些) 的折中。本例两个参考设计的电压基准是一样的。

电池温度

电池温度通过负温度系数 (NTC) 电阻测量。其阻值在 25°C 近似为 10 kΩ。NTC 为分压器的一部分, 由基准电压供电。

NTC 两端电压的精度和从电压测量电路取得的一样。

精度：

$$\frac{3.67V}{1024steps} = 3.58 \frac{mV}{step}$$

量化单位数目 (step) 可以由下面的公式得到：

$$N = 1024 \cdot \frac{R_{NTC}}{R_{NTC} + 10k\Omega}$$

由于 NTC 的电阻不是线性的，从 ADC 返回的结果很难计算出实际的温度。此时可以利用查询表来查找对应的温度（参见表 2）。表中数据说明，在 ADC 数值为 400 到 675 时，单位温度间隔为 0.5°C。当 ADC 数值为 400 时温度近似为 37°C，675 则近似为 8.6°C。利用这个表并对头文件 B_DEF.H 做一些小改动就可以实现对各种 NTC 电阻的操作。ATtiny15 电池充电器参考设计假定测温电阻的线性度足以测量温度变化，因此它采用与常数进行比较的方式对温度进行监控。

表中的数据是按照 NTC 数据手册以及 NTC 分压器计算得到的。

Table 2. NTC 温度分级

ADC 读数	温度 (°C)	多少个 0.5°C	NTC (Ω) 阻值
675	8.6	5	19341
650	11	4	17380
625	14	6	15664
600	16	5	14151
575	18.8	5	12806
550	21.2	5	11603
525	23.6	5	10521
500	26.2	5	9542
475	28.8	4	8652
450	32	6	7840
425	34	4	7095
400	37	5	6410
375	39.4	5	5778

AT90S4433 电池充电器

这一节讲述基于 AT90S4433 的电池充电器的工作原理。

参数

振荡器频率： $f_{OSC} = 7.3728 \text{ MHz}$

饱和电压： $V_{sat} = 0.5V$

输入电压： $V_I = 15V$

输出电压： $V_O = 1.5V$

最大输出电流： $I_{O,max} = 1.5A$

8 位 PWM:

$$T = \frac{510}{f_{OSC}} = 69.199\mu s$$

占空比为 50% 时：

$$t_{on} = \frac{69.199\mu s}{2} = 34.60\mu s$$

电感值：

$$L = \frac{(V_I - V_{sat} - V_O)t_{on}}{2I_{o,max}} = \frac{(15V - 0.5V - 1.5V)34.60\mu s}{2 \cdot 1.5A} = 149.9\mu H$$

$$t_{on} = \frac{L \cdot 2 \cdot I_{O,max}}{V_I - V_{sat} - V_O} = \frac{150\mu H \cdot 3A}{15V - 0.2V - 1.5V} = 33.83\mu s$$

计算得占空比为：

$$\frac{t_{on}}{T} = \frac{33.83\mu s}{69.199\mu s} = 0.489 = 48.9\%$$

AT90S4433 测量电路

电池电压

为了监控电池正负两极之间的充电电压，参考设计使用了一个运放。为了选择合适的测量范围，需要决定电池的类型以及电池里有多少个电池单元，还要选择合适的输入电压 ($V_1 - V_2$) 和分压电阻。测量电池电压的运放电路为普通的差分运放电路。运放的输出电压公式如后面所示。ADC 的测量范围为 $A_{GND} \sim A_{REF}$ (3.67V)，而运放的输出电压 (V_{BAT2}) 应该界于下式决定的范围之间：

$$V_{BAT2} = \frac{R_a}{R_b} * (V_1 - V_2)$$

式中：

- V_{BAT2} 是运放输出电压 (AVR A/D 的输入)
- V_1 为电池正极
- V_2 为电池负极
- R_a 和 R_b 是用来为运放设置增益的电阻
- R_a 等于 R_{10} 和 R_{12}
- R_b 等于 R_6 和 R_7

最大充电电压为：

$$(V_1 - V_2) = \frac{R_b}{R_a} * A_{REF} = \frac{33k\Omega}{10k\Omega} * 3.67 = 12.1V$$

运放增益为：

$$G_{U1B} = \frac{Ra}{Rb} = \frac{10\text{ k}\Omega}{33\text{ k}\Omega} = 0.303$$

电池测量精度为：

$$\frac{ADC\text{ resolution}}{G_{U1B}} = \frac{3.58\text{ mV}}{0.303} = 11.82 \frac{\text{mV}}{\text{step}}$$

充电电流

充电电流通过 0.033W 的敏感电阻 (R1) 获取。为了提高测量精度，这个电压通过运放放大，然后馈送到 ADC。

电压放大系数为：

$$1 + \frac{R_5}{R_2} = 1 + \frac{39\text{ k}\Omega}{680\Omega} = 58.4$$

从而运放输出电压为：

$$V_{ibat2} = \left(1 + \frac{R_5}{R_2}\right) \cdot I_{Shunt} \cdot R_6$$

亦即：

$$V_{ibat2} = 1.926 \cdot I_{Shunt}$$

能够测量的最大电流：

$$I_{BAT_{max}} = \frac{3.58}{1.926} = 2.0\text{ A}$$

精度为：

$$\frac{2000\text{ mA}}{1024\text{ steps}} = 1.95 \frac{\text{mA}}{\text{step}}$$

从而可计算得到某一个量化单位数目对应的电流值：

$$I_{Shunt} = N \cdot 1.95 \frac{\text{mA}}{\text{step}}$$

而电流值对应的量化单位数目为：

$$N = \frac{I_{Shunt}}{1.95 \frac{\text{mA}}{\text{step}}}$$

ATtiny15 电池充电器

这一节介绍基于 ATtiny15 的参考设计的工作原理。片内 1.6 MHz 的 RC 振荡器通过 PLL 产生 25.6 MHz 的 PWM 振荡频率。参考设计没有使用分压用的电阻，从而最大电压限制为 3.67V，适合包含 1-2 个电池单元的 NiCd 电池或 NiMh 电池。如果想得到更高的电压，只要简单地加上分压电阻，使被测电压限制于 0-3.67V 即可。本节在最后一节将介绍如何计算这些电阻值。

参数

振荡器频率：	$f_{OSC} = 25.6 \text{ MHz}$
饱和电压：	$V_{sat} = 0.5V$
输入电压：	$V_I = 12V$
输出电压：	$V_O = 1.5V$
最大输出电流：	$I_{O,max} = 1.5A$
8 位 PWM:	

$$T = \frac{255}{f_{OSC}} = 9.96 \mu S$$

占空比为 50% 时开态时间为：

$$t_{on} = \frac{9.96 \mu s}{2} = 4.43 \mu s$$

电感为：

$$L = \frac{(V_I - V_{sat} - V_O) t_{on}}{2 I_{O,max}} = \frac{(15V - 0.5V - 1.5V) 4.43 \mu s}{2 \cdot 1.5A} = 21.1 \mu H$$

$$t_{on} = \frac{L \cdot 2 \cdot I_{O,max}}{V_I - V_{sat} - V_O} = \frac{22 \mu H \cdot 3A}{15V - 0.5V - 1.5V} = 4.88 \mu s$$

计算得到占空比为：

$$\frac{t_{on}}{T} = \frac{4.88 \mu s}{9.96 \mu s} = 0.449 = 44.9 \%$$

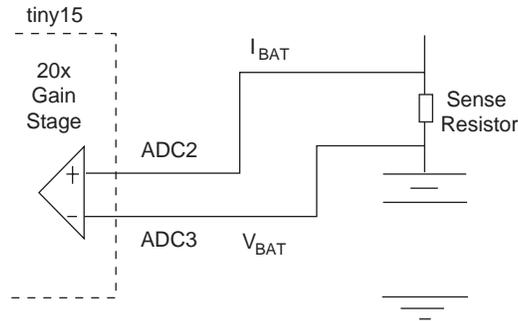
ATtiny15 测量电路

电池电压

充电电压直接从电池正极测得。使用高于基准电压 (3.67V) 的充电电压时，需要增加两个电阻将其分压到 0-3.67V 的范围。这个输入引脚也是测量充电电流时的差分输入负端，如图 4 所示。电流信号经过 0.25W 的敏感电阻转换为电压信号，然后输入到 20x 增益放大阶段。

所有的测量值的精度为 10 比特 (1024 级)。

Figure 4. 测量电压和电流



电压精度由 A_{REF} 决定。

精度：

$$\frac{3.67V}{1024steps} = 3.58 \frac{mV}{step}$$

为了选择合适的测量范围，需要决定要充什么类型的电池，以及电池里有几个单元。ADC 的测量范围为 $A_{GND} \sim A_{REF}$ (3.67V)。分压器的输出电压 (V_{ADC}) 范围是。

$$V_{ADC} = \frac{Rb}{Ra + Rb} * (Vb)$$

式中：

- V_{ADC} 是分压器的输出电压 (AVR A/D 的输入)
- Vb 为电池电压
- Ra 和 Rb 是对电池电压进行分压的电阻
- Ra 等于 $R8$
- Rb 等于 $R16$

用来分压的电阻 $R9$ 和 $R17$ 必须分别等于 $R8$ 和 $R16$ 。在参考设计中 $R8 = R9 = 3.7 \text{ kW}$ ， $R16 = R17 = 2.2 \text{ kW}$ 。

最大充电电流为：

$$V_{bat} = \left(1 + \frac{R_8}{R_{16}}\right) V_{ADC} = \left(1 + \frac{3,7}{2,2}\right) 3,67 = 9,8V$$

充电电流

充电电流的测量是通过 0.25W 的敏感电阻实现的。电阻上的电压首先要放大 20 倍以提高精度，然后再进行 A/D 转换。

ADC 输入电压为：

$$V_{ibat} = 20 \left(\frac{R_B}{R_B + R_A} \right) \cdot I_{shunt} R_{18}$$

式中：

- V_{ibat} 是 ADC 的模拟输入电压
- I_{shunt} 是流经 0.25 敏感电阻的电流
- R_a 和 R_b 是用来对敏感电阻上的电压进行分压的电阻。分压比与电压通路是一样的
- $R_a = R_9$
- $R_b = R_{17}$

$$V_{ibat} = 1.864 \cdot I_{shunt}$$

能够测量的最大电流是：

$$I_{shunt\max} = \frac{3.67}{1.864} = 1.96A$$

精度为：

$$\frac{196mA}{1024steps} = 1.92 \frac{mA}{step}$$

电流的量化单位数目为：

$$N = \frac{I_{BAT}}{1.92 \frac{mA}{step}}$$

量化单位数目的电流为：

$$I_{BAT} = N \cdot 1.92 \frac{mA}{step}$$

软件实现

本节介绍 AT90S4433 参考设计的 C 代码软件实现。ATtiny15 的汇编代码也遵从同样的原理。具体请参考汇编程序的说明。

在编译时要确定电池类型。

软件可以进行扩展以支持多个电池同时充电。一个直接的方案是在进行涓流充电时对各个电池进行分时充电。若每个电池的电池单元数目一样，则 SLA 电池和 Li-Ion 电池 可以恒定电压的方式并行充电。每个电池单元的充电电流是受限的，电压也一样。

“电池特性” (b_car.h) 的所有数据都根据标度因子计算得到。这些数据在包含文件里定义，在编译时计算，在程序运行时以常数方式处理。所有从 ADC 输出的数据都可以直接与这些常数进行比较。也就是说，在程序运行过程当中不需要进行实时计算，从而节省了计算时间和程序空间。计算公式以及数据都是从“测量电路”一节获取的。请参见第 9 页的“AT90S4433 测量电路”和第 12 页的“ATtiny15 测量电路”。

对于 NiCd 电池，如果电池温度在允许范围之内，充电程序就会启动。在温度超出限制，或电压超过最大值，或超出最大快速充电时间时停止。

检测电池已经充满的普通方法是检测温度上升速率 (dT/dt) 和电压降低速率 (-dV/dt)。因此，充电器会每隔一分钟检测一次温度，每隔一秒钟检测一次电压。这些数据将与上一次数据进行比较。一旦电池充满，充电状态就自动切换到涓流充电，充电程序跳转到 trickle_charge() 函数。

trickle_charge() 循环检测充电状态、温度 / 电压的改变，并适当地调节充电电流。一旦温度或电压超标，错误标志置位，函数终止。若没有错误，用户也没有改变充电状态，函数将一直循环工作。

用户设置

本充电器是一个多用途充电器，可以对四种电池进行充电，一次可以充多个电池单元。开发人员只要简单地在编译代码时修改某些参数即可。为了防止对电池和环境造成破坏，在编译前一定要注意选择正确的参数。

改变电池类型

每一个电池类型都有一个 C 文件和 h 文件。在编译之前将需要的电池文件添加到编译器里，并在 B_Def.h 文件里将其他电池类型的定义屏蔽掉。

改变电池数目

修改 B_Def.h 文件的“cells”参数

改变电池容量

修改 B_Def.h 文件的“capacity”参数

改变锂电池单元电压

修改 B_Def.h 文件的“cell_voltage”参数

改变 ADC 量化单位的数值

如果根据测量一节修改了电阻值，B_Def.h 文件的“voltage_step”参数和“current_step”参数必须得到正确的修改。这很重要，否则可能损坏充电器。

源代码文件

以下为 Atmel 提供的源代码文件。

Table 3. C 源代码文件

文件名	说明	代码大小 ⁽¹⁾
lo4333.h	AT90S4333 的头文件	
cstartup.s90	C 编译器的 Start-up 文件	
Lnk0t.xcl	链接器的命令文件，针对 AT90S4433 进行了优化	
B_def.h	定义电池类型，电压，容量和电压单位数值	
Bc.h	常数和宏定义的头文件	
Bc.c	主程序，使用所有的电池类型	474 字节
SLA.h	SLA 电池的头文件，包含充电器参数和函数声明	
SLA.c	SLA 电池的的源代码	446 字节
NiCd.h	NiCd 电池的头文件，包含充电器参数和函数声明	

Table 3. C 源代码文件

文件名	说明	代码大小 ⁽¹⁾
NiCd.c	NiCd 电池的的源代码	548 字节
NiMh.h	NiMh 电池的头文件，包含充电器参数和函数声明	
NiMh.c	NiMh 电池的的源代码	514 字节
Liion.h	Li-ion 电池的头文件，包含充电器参数和函数声明	
Liion.c	Li-ion 电池的的源代码	690 字节

Note: 1. 是 version 1.0 的代码大小，以 IAR C Ver.1.41C 编译，选择最高一级代码大小优化。

Table 4. 汇编源代码

文件名	说明	代码大小
bc.inc	寄存器、A/D 通道和常数声明	
tn15def.inc	ATtiny15 的头文件	
NiCd.inc	NiCd 电池的头文件，包含充电器参数	
NiCd.asm	NiCd 电池的的源代码	324 字节
NiMh.inc	NiMh 电池的头文件，包含充电器参数	
NiCd.asm	NiMh 电池的的源代码	328 字节
Liion.inc	Li-ion 电池的头文件，包含充电器参数	
Liion.asm	Li-ion 电池的的源代码	340 字节

BC.C

这个 C 模块包含主程序、设置函数、UART 函数、实时时钟函数以及中断处理例程。

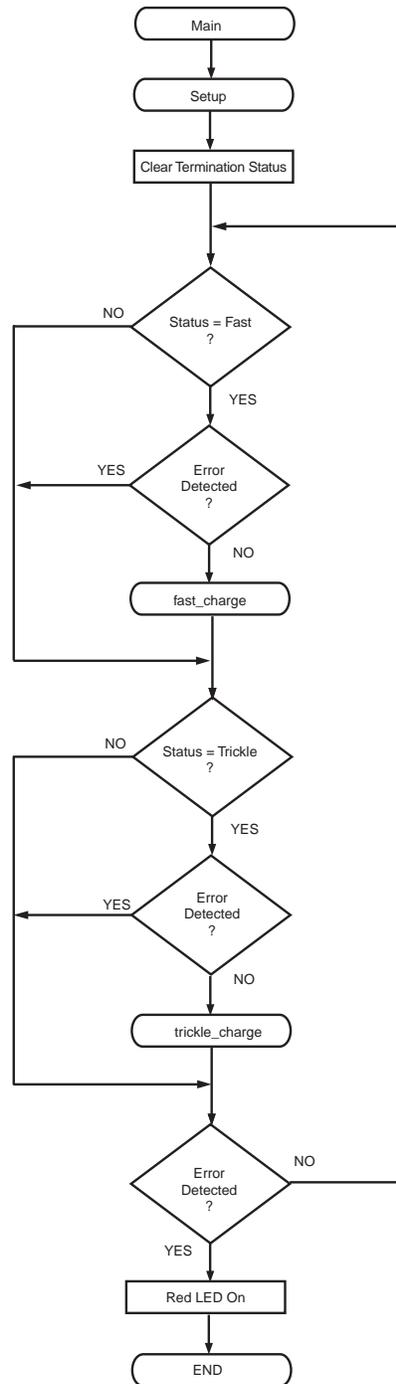
“setup” 函数处理所有的初始化工作。其中 UART 被初始化，实时时钟设置为 0。之后程序以空闲模式循环，直至全局状态变量改变。

PWM 运行后实时时钟启动，并随着 PWM 停止而停止。这保证了只有在充电时时间才有意义。另一方面，这个方法的缺点是 与时间有关的测量数据 (dV/dt 或 dT/dt) 有可能不准确。

用户可以通过按按钮来切换充电状态并将引发中断。在中断例程中，按照所按的不同按钮，系统或者切换到快速充电模式，或者切换到涓流充电模式。主程序按照变量 “charge status” 调用不同的函数。

BC.C 还包括某些适用于各种电池类型的公共函数。最重要的两个为

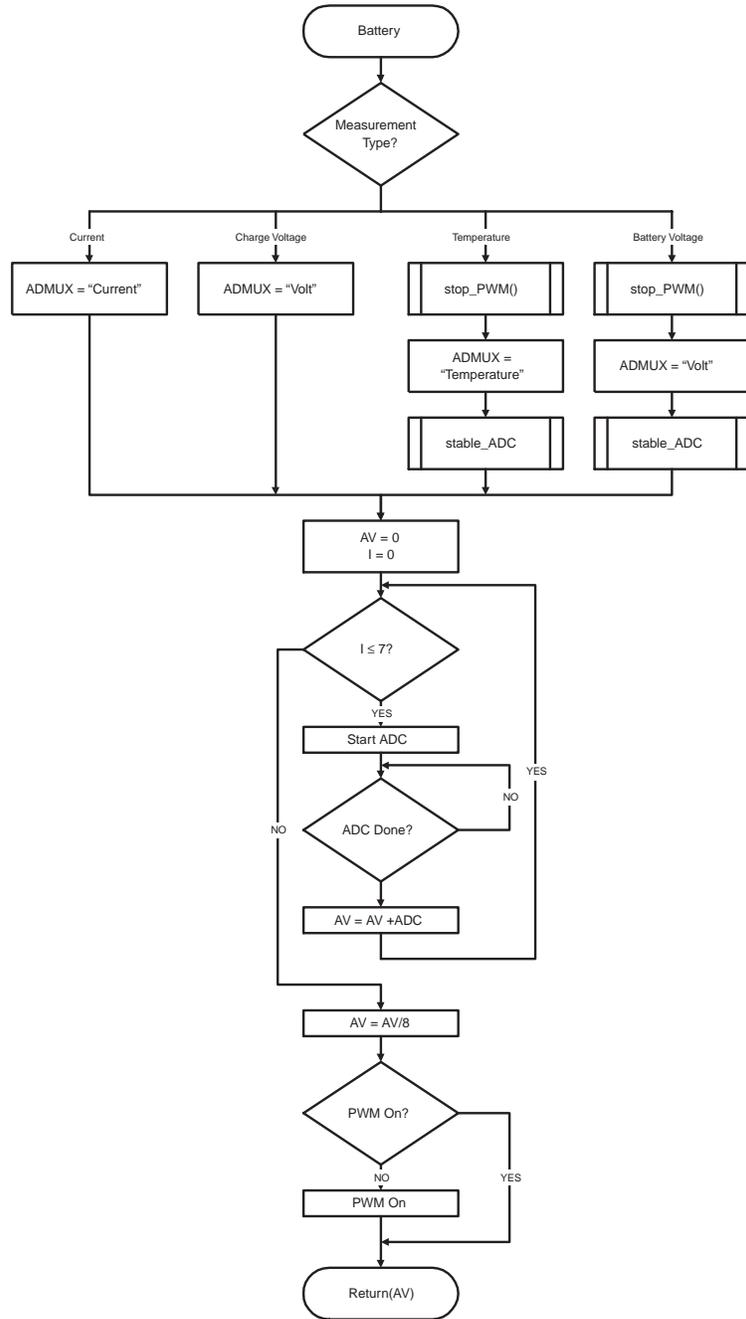
Figure 5. Main() 函数



int battery (Unsigned Char Value)

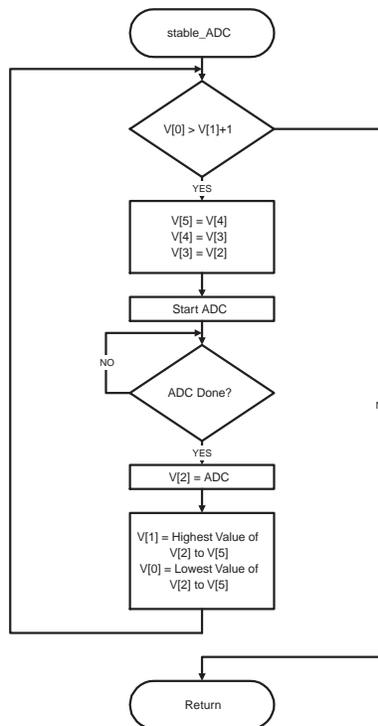
这个函数在进行 A/D 转换时被调用，并按照要求控制 ADC 寄存器和 PWM。它从 ADC 读取 8 位的结果并求平均值，然后将这个平均值传回给调用函数。

Figure 6. Battery() 函数



void stable_ADC (Void)

测量电池电压和温度时 `stable_ADC` 被调用。它保证 ADC 数据在定义的范围内是稳定的。这对于精确的测量是非常重要的。函数一直循环到获得三个 ADC 数据，而且最大值比最小值不能超出一个量化单位。

Figure 7. `stable_ADC()` 函数**BC.H**

模块包含各种处理位操作的宏定义，以及各种充电状态和停止充电的位屏蔽常数。

“charge status” 指示当前充电器的实际状态：快速充电、涓流充电，或出现了错误。对于 Li-Ion 电池和 SLA 电池，有变量指示充电模式，包括电压常数和电流常数，以及锂电池是否处于快速充电模式的最后阶段（称为“延迟”）。“termination” 则定义了停止快速充电的原因，或者是在什么地方检测到了错误，以用来调试。

B_DEF.H

这个模块定义了需要充电的电池类型。设计什么类型的充电器就需要使用什么类型的电池定义。

电池类型决定了充电及停止充电的算法。由于所有的电池都使用相同的函数，即 `battery()`、`fast_charge()` 和 `trickle_charge()`，若同时选择了多种电池类型，在程序编译时将出错。如果没有选择电池类型也会造成编译出错。

电池单元的数目决定了电池的电压以及相关的常数。在此假定所有的电池单元是串联在一起的。0 个单元并不是合理现象，但是参考设计对此并不在意：它将简单地输出 0 安培的充电电流。BUCK 变换器的电压范围以及电压测量电路决定了电流的上限。

电池容量（以 mA 为单位）定义了充电电流及相关的常数。

除了 SLA 电池，其他类型的电池在快速充电时工作于 1C 的保守方式。SLA 电池的充电电流为 2C。这也就为电池容量设定了上限。BUCK 变换器最大可以输出 1.5 A 的电流。SLA 的最大容量为 750 mAh，其他电池类型则为 1500 mAh。如果需要对 NiCd 电池和 NiMH 电池以更大的充电电流进行充电，BUCK 变换器必须做适当调整。电流大于 2 A 时电流测量电路也需要修改。如果需要对一个容量超出了上面的计算值的电池进行充电，一个方法是改变 BUCK 变换器，或者是降低充电电流。

对于锂电池，根据不同的生产商存在着两种单元电压。在编译时必须对这个电压值 (4.1V 或 4.2V) 进行编辑。选择锂电池后这个值将自动包含进代码里。如果选择了错误的数值，虽然在编译时不会出错，但在实际应用中将导致错误的充电算法，最终损坏电池和充电器。

ADC 量化单位的数值需要按照实际的测量电路进行修改。详见测量电路一节。

NTC 表定义了 ADC 的量化单位数值。一个量化单位代表 $0.5 \times C$ 的温度变化。这个查询表适用于 NiCd 电池。若选择了其他 NTC，这个表也需要修改。

SLA.C

充电方法

SLA 电池的快速充电算法采用恒定电压法。在充电之前首先用一个简单的 (但出人意料地有效) 方法确定出充电电压：给电池加载 1C (10 mA) 的恒定电流，测量到的电池电压即为充电电压。

电池首先用恒定电压进行充电 (电流可以变化)。当电流下降到 0.2C 以下时快速充电结束，开始涓流充电。

涓流充电同样为恒定电压方式，只是电压比快速充电时稍低。涓流充电的时间可以是一个固定值。

充电参数简介

快速充电：

固定的快速充电电压 = 电池单元数 * 2450 mV

涓流充电：

固定的涓流充电电压 = 电池单元数 * 2250 mV

一般充电过程结束的判别标准：

绝对最低温度 $T = 0 \times C$

绝对最高温度 $T = 45 \times C$

快速充电过程结束的判别标准：

最小电流门限 $I = 0.2C$

快速充电过程的错误：

快速充电时的最高温度 $T = 30 \times C$

最大快速充电时间 $t = 60 \text{ min}$ ，充电电流为 1C

最大的快速充电电流 $I = 2C$

涓流充电过程结束的判别标准：

无

Figure 8. SLA 电池的 Trickle_charge() 函数

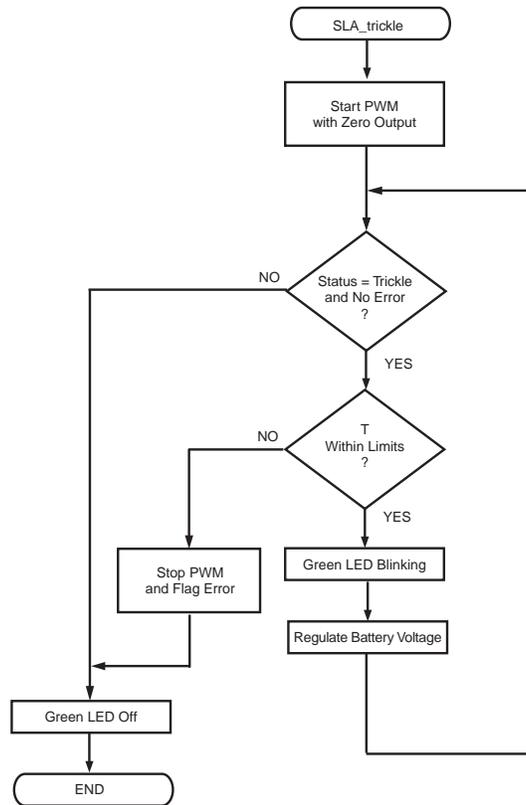


Figure 9. SLA 电池的 Fast_charge() 函数，第一部分 (2)

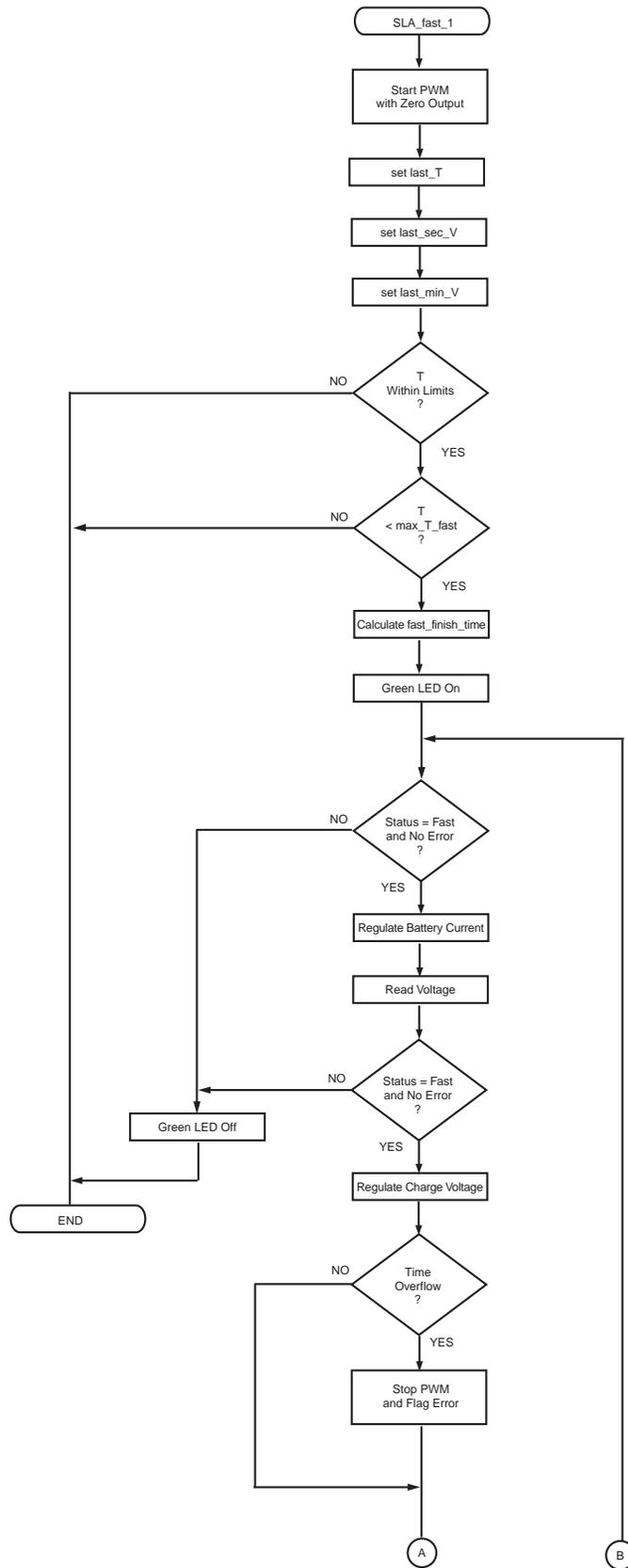
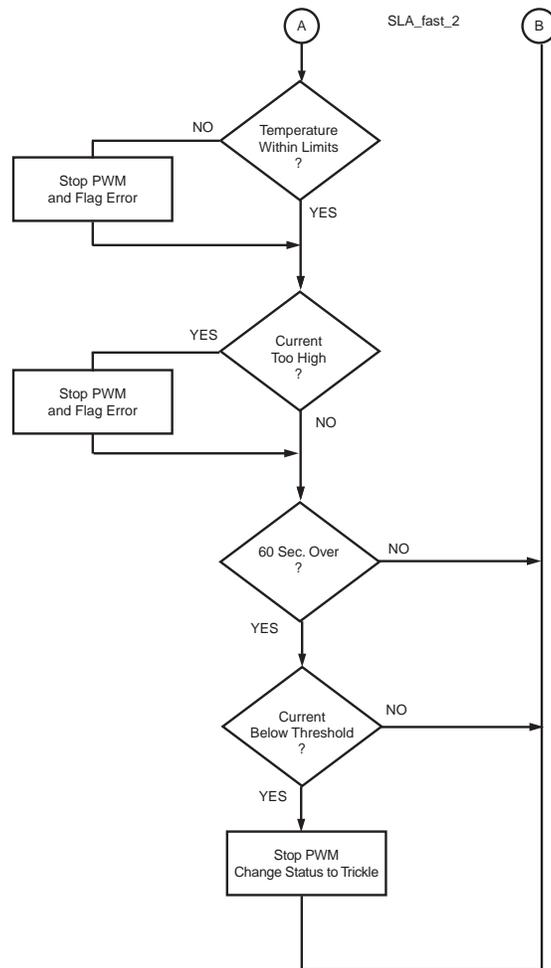


Figure 10. SLA 电池的 Fast_charge() 函数，第二部分 (2)



NiCd.C

充电方法

NiCd 电池的快速充电算法采用恒定电流法。在快速充电模式下充电电流设置为 1C；在涓流模式下则为 C/40。判断充电过程结束的方法是电压下降 (-dV/dt) 法。备用的结束充电判别方案还有最大充电电压、温升 (dT/dt) 以及最大充电时间。

电池充满后自动切换到涓流模式，MCU 开始执行 trickle_charge() 函数。

充电参数简介

充电条件：

快速充电：

快速充电电流 = 1C

涓流充电：

涓流充电电流 = 0.025C

一般充电过程结束的判别标准：

绝对最低温度 $T = 5 \times C$

绝对最高温度 $T = 40 \times C$

绝对最大充电电压 $V = \text{电池单元数} \times 1500 \text{ mV}$

快速充电过程结束的判别标准：

电压下降门限 $-dV/dt = 20 \text{ mV/min/单元}$

温升门限 $dT/dt = 1 \times C/\text{min}$

快速充电过程的错误：

快速充电时的最低温度 $T = 15 \times C$

最大快速充电时间 $t = 90 \text{ min}$ ，充电电流为 $1C$

Figure 11. NiCd 电池的 Trickle_charge() 函数

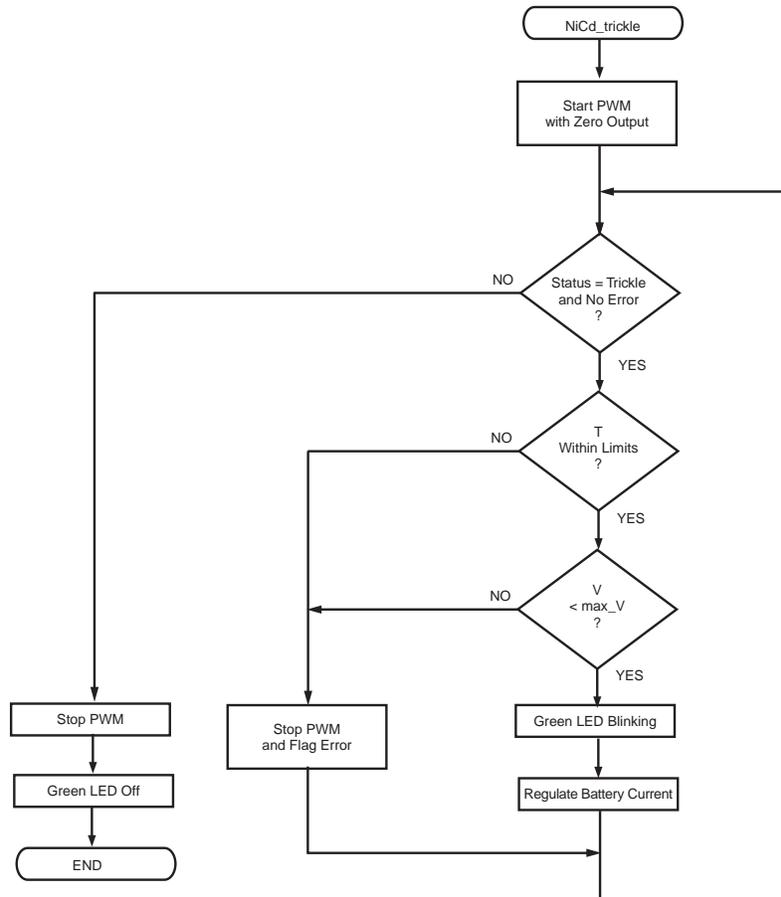


Figure 12. NiCd 电池的 Fast_charge() 函数，第一部分 (2)

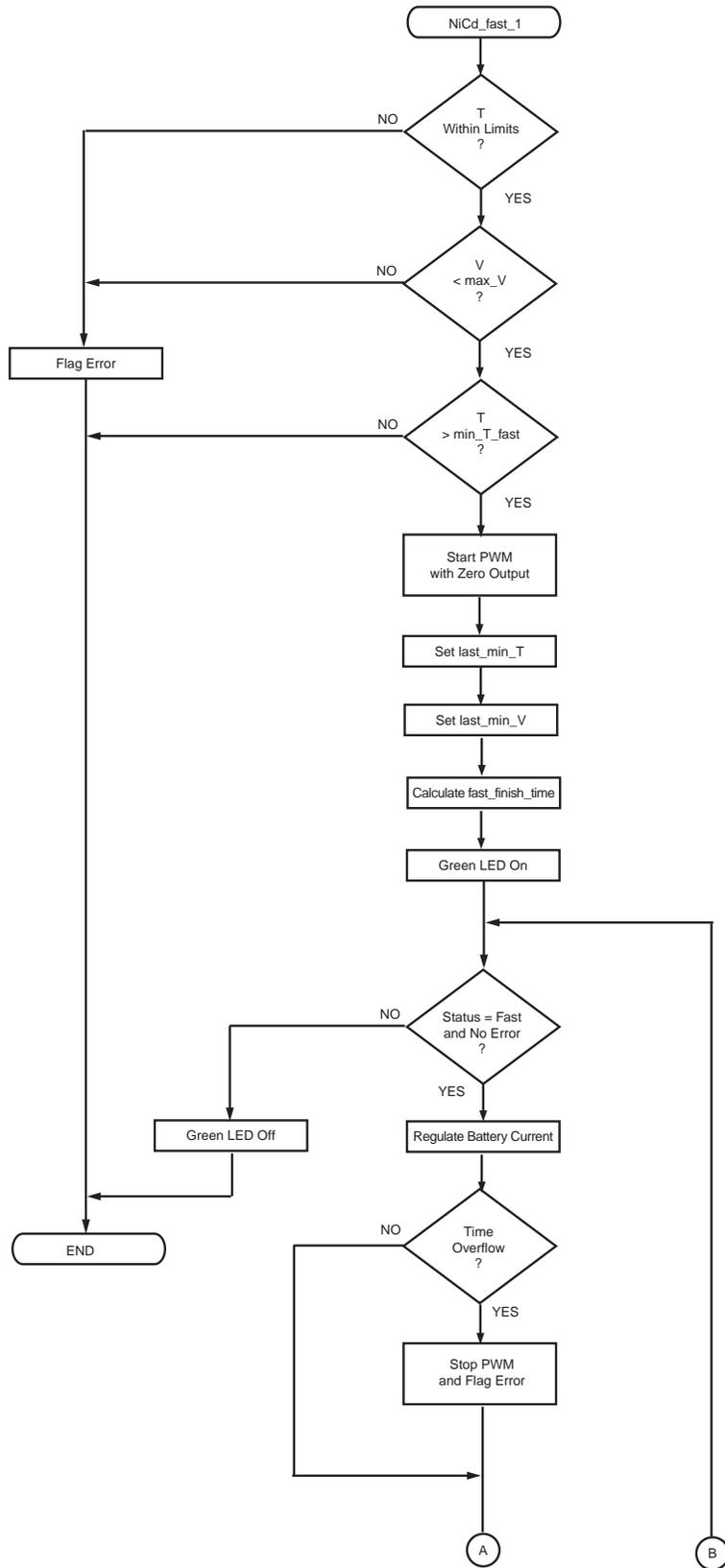
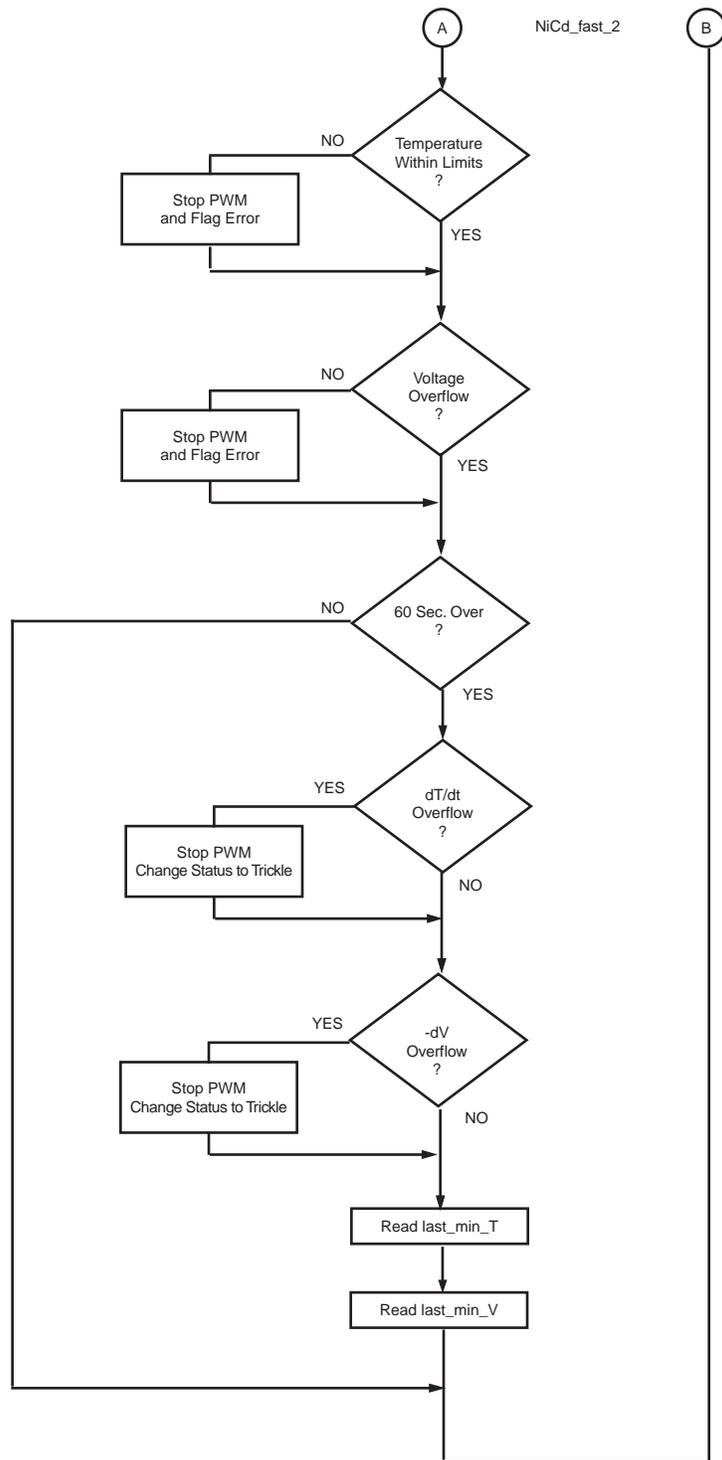


Figure 13. NiCd 电池的 Fast_charge() 函数，第二部分 (2)



NiMH.C

充电方法

NiMH 电池的快速充电算法采用恒定电流法。在快速充电模式下充电电流设置为 1C；在涓流模式下则为 C/40。判断充电过程结束的方法是电压下降 (-dV/dt) 法和温升 (dT/dt) 法。最大充电电压以及最大充电时间可以作为备用方案。

电池充满后自动切换到涓流模式，MCU 开始执行 trickle_charge() 函数。

充电参数简介

充电条件：

快速充电：

快速充电电流 = 1C

涓流充电：

涓流充电电流 = 0.025C

最大涓流充电时间 t = 90 min，充电电流为 0.025C

一般充电过程结束的判别标准：

绝对最低温度 T = 5×C

绝对最高温度 T = 40×C

绝对最大充电电压 V = 电池单元数 * 1500 mV

快速充电过程结束的判别标准：

温升门限 dT/dt = 0.5×C/min

快速充电过程的错误：

快速充电时的最低温度 T = 15×C

最大快速充电时间 t = 90 min，充电电流为 1C

Figure 14. NiMH 电池的 Trickle_charge() 函数

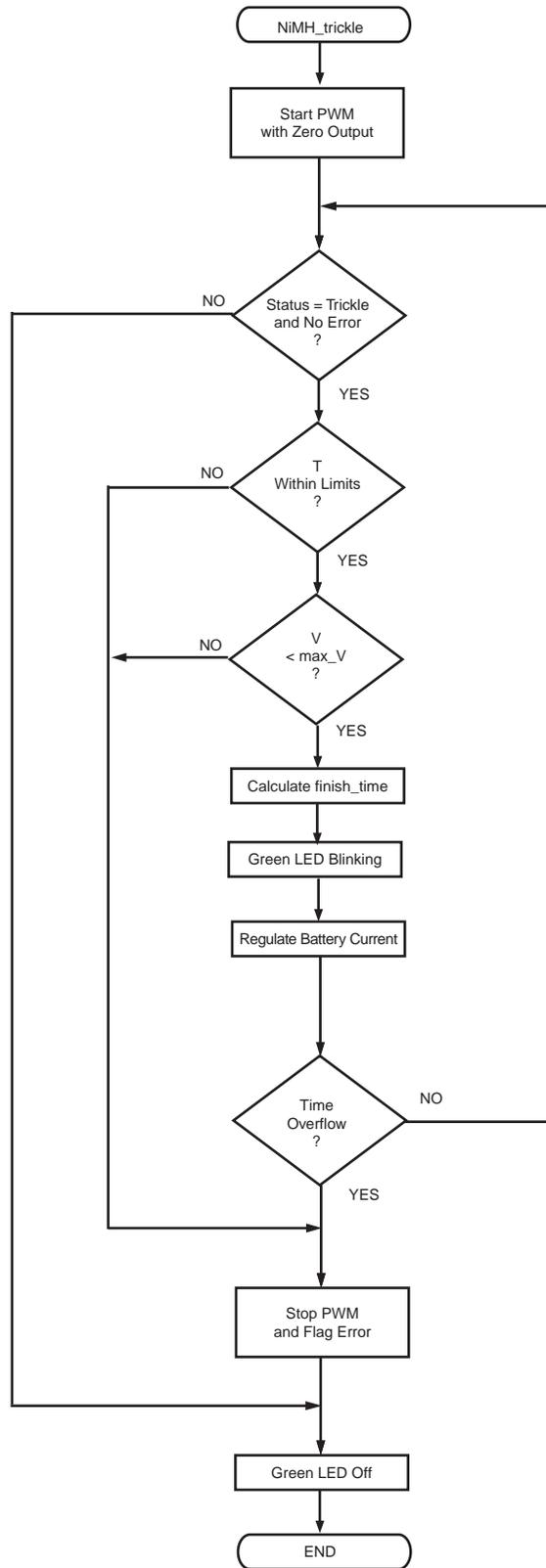


Figure 15. NiMH 电池的 Fast_charge() 函数，第一部分 (2)

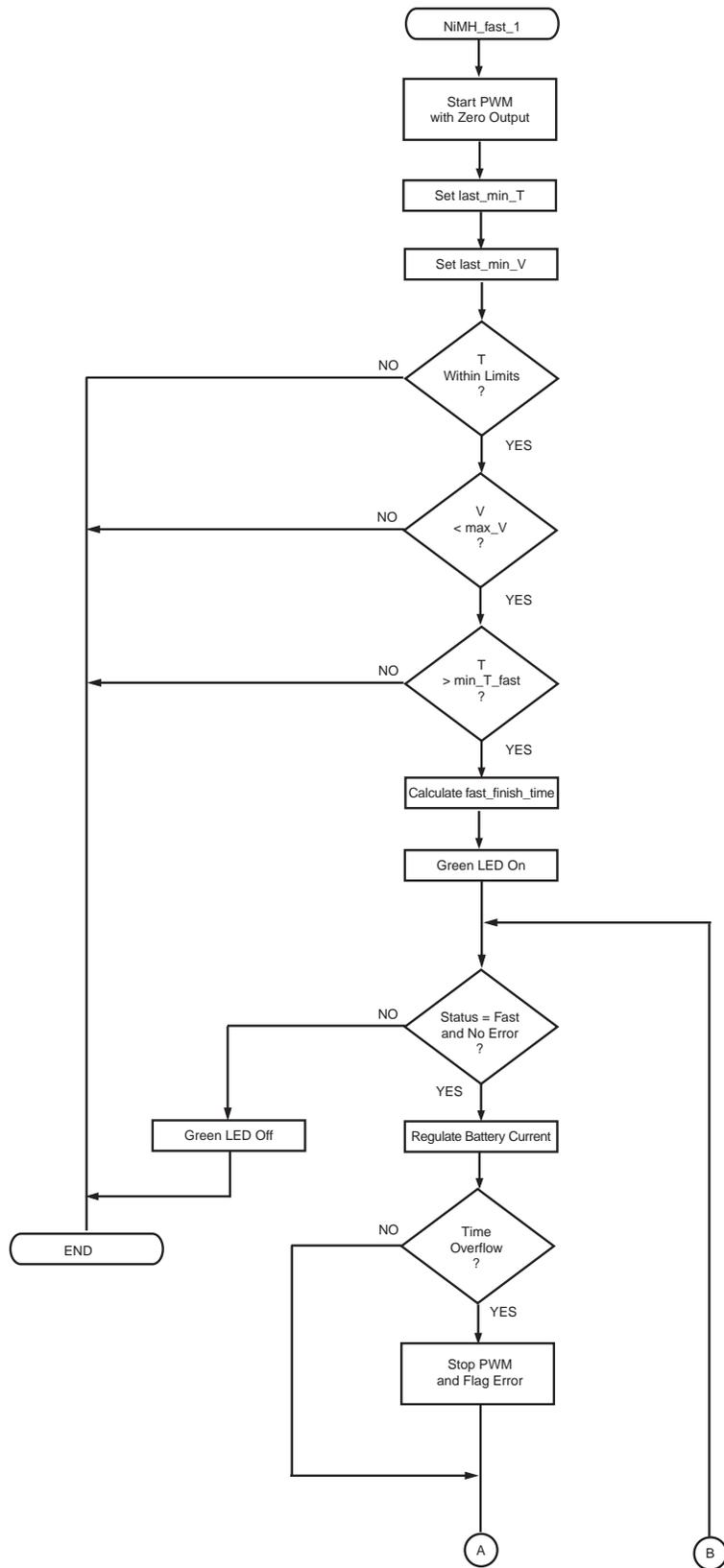
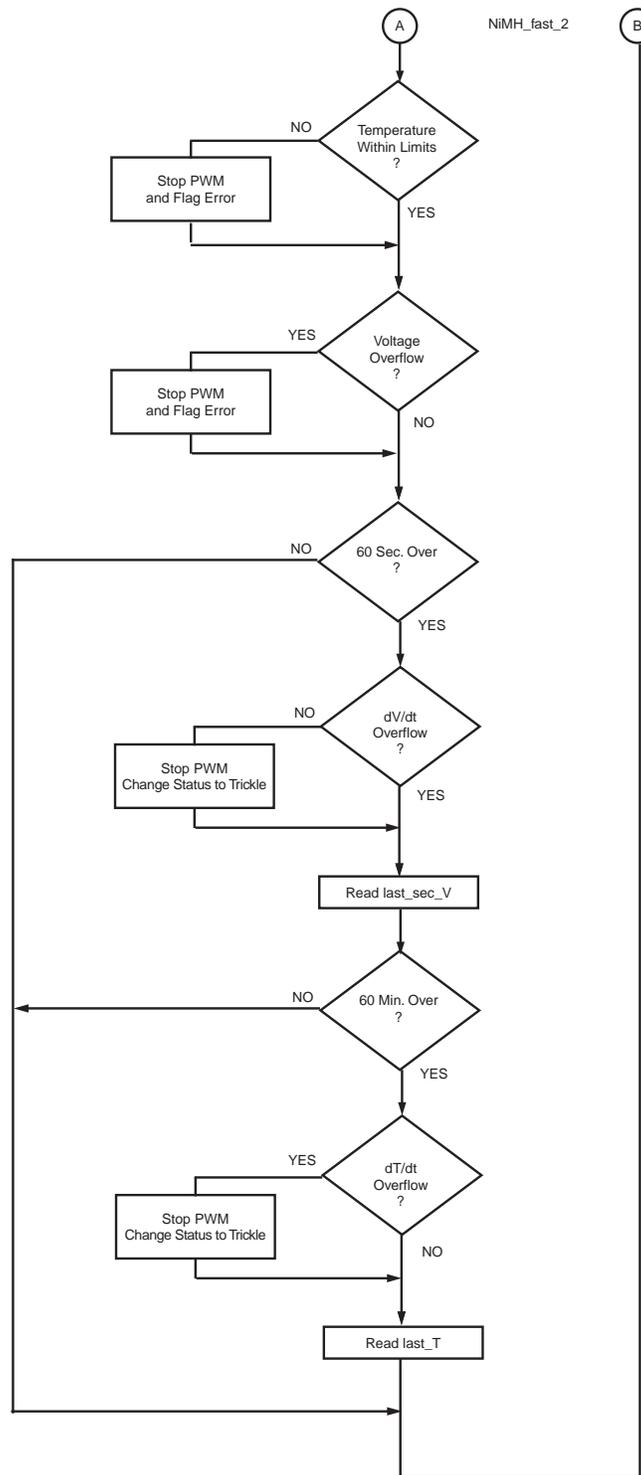


Figure 16. NiMH 电池的 Fast_charge() 函数，第二部分 (2)



Lilon.C

充电方法

Li-Ion 电池是需要充电的最先进的电池。快速充电首先以恒定充电电流 1C 开始，直到电池单元电压达到 4.1 或 $4.2V \pm 50\text{ mV}$ 。然后以恒定电压法充电，直至充电电流下降到 I_{min} 。

为了精确测量电池电压（而不是充电电压）在测量时要关闭 PWM。由于充电算法要由恒定电流法切换到恒定电压法，因此还需要监控充电电压。这也就是为什么需要测量两个电压的原因，一个是在关闭 PWM 时测得的，一个是在 PWM 工作时测得的。

涓流充电在原理上与快速充电是一样的。只是电流要小得多，恒定电压阶段则通过定时器确定何时结束。

充电参数简介

充电条件：

快速充电：

绝对最大充电电压 $V = \text{电池单元数} * \text{单元电压}$

电压容限 = 电池单元数 * 50 mV

快速充电电流 = 1C

最小电流门限 = 50 mA/ 电池单元

涓流充电：

涓流充电电流 = 0.025C

最大涓流充电时间 $t = 90\text{ min}$ ，充电电流为 0.025C

一般充电过程结束的判别标准：

绝对最低温度 $T = 5 \times C$

绝对最高温度 $T = 40 \times C$

快速充电过程结束的判别标准：

参见充电条件

快速充电过程的错误：

快速充电时的最低温度 $T = 10 \times C$

最大快速充电时间 $t = 90\text{ min}$ ，充电电流为 1C

Figure 17. 锂电池的 Trickle_charge() 函数

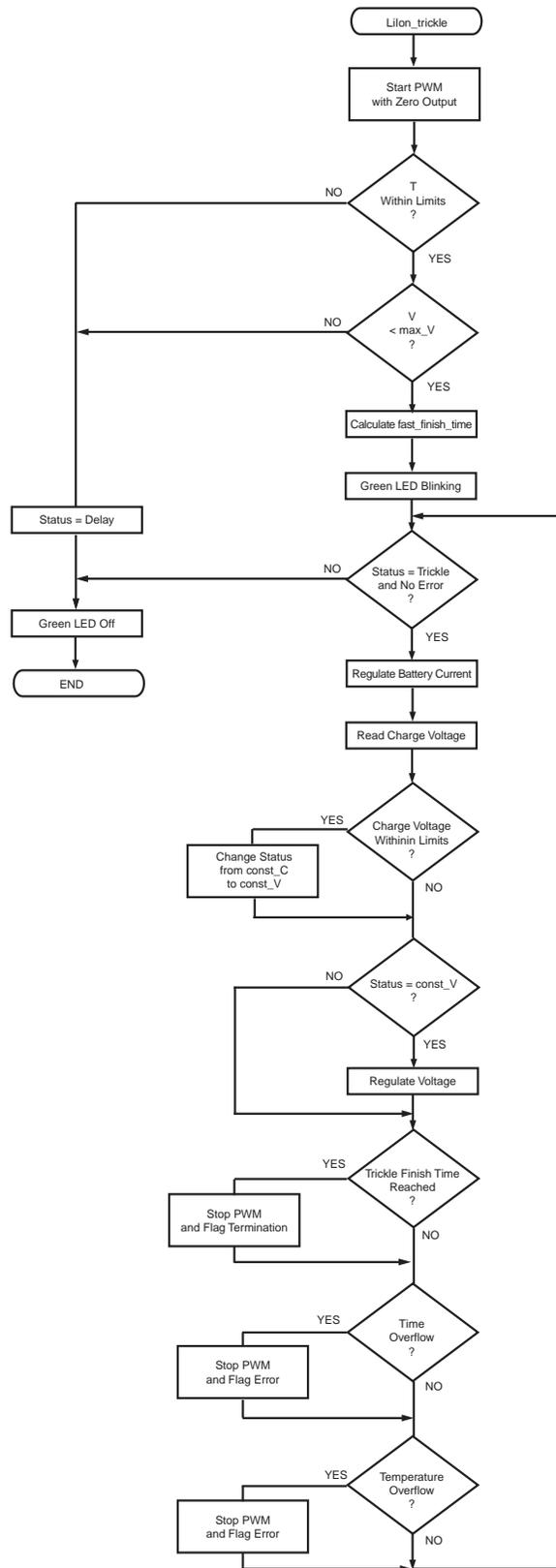


Figure 18. 锂电池的 Fast_charge() 函数，第一部分 (2)

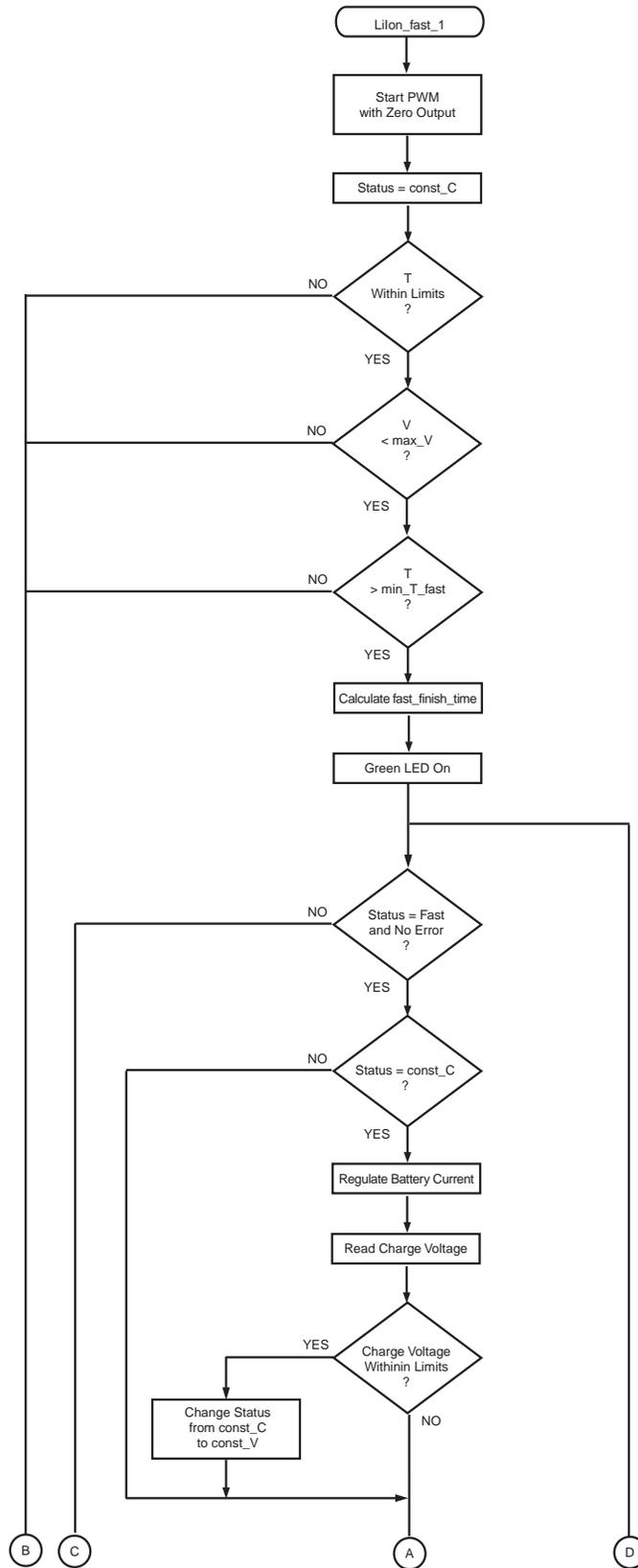
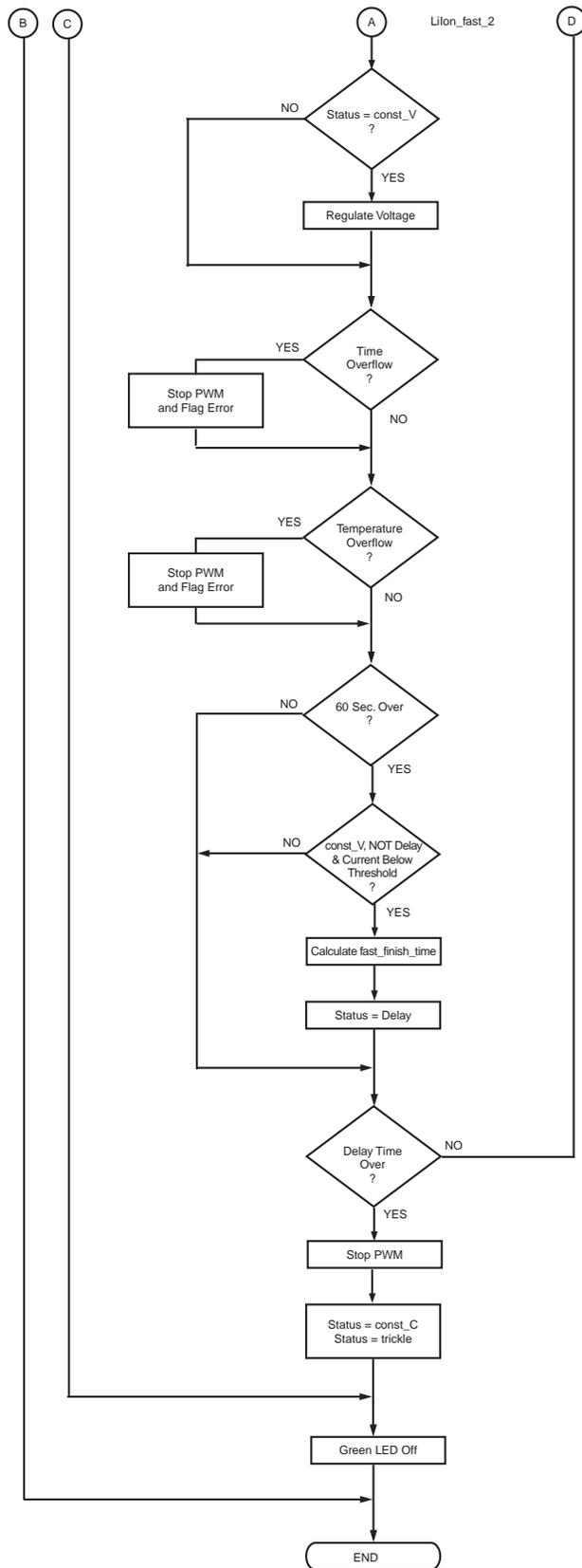


Figure 19. 锂电池的 Fast_charge() 函数，第二部分 (2)



建议的改进

NiCd 电池具有记忆效应 – 充几次电之后就无法充满了。为了复位记忆，可以增加一个泄放电阻，允许 MCU 在充电之前先对其进行放电处理。

附录 1: 原理图

Figure 20. 主模块的方框图

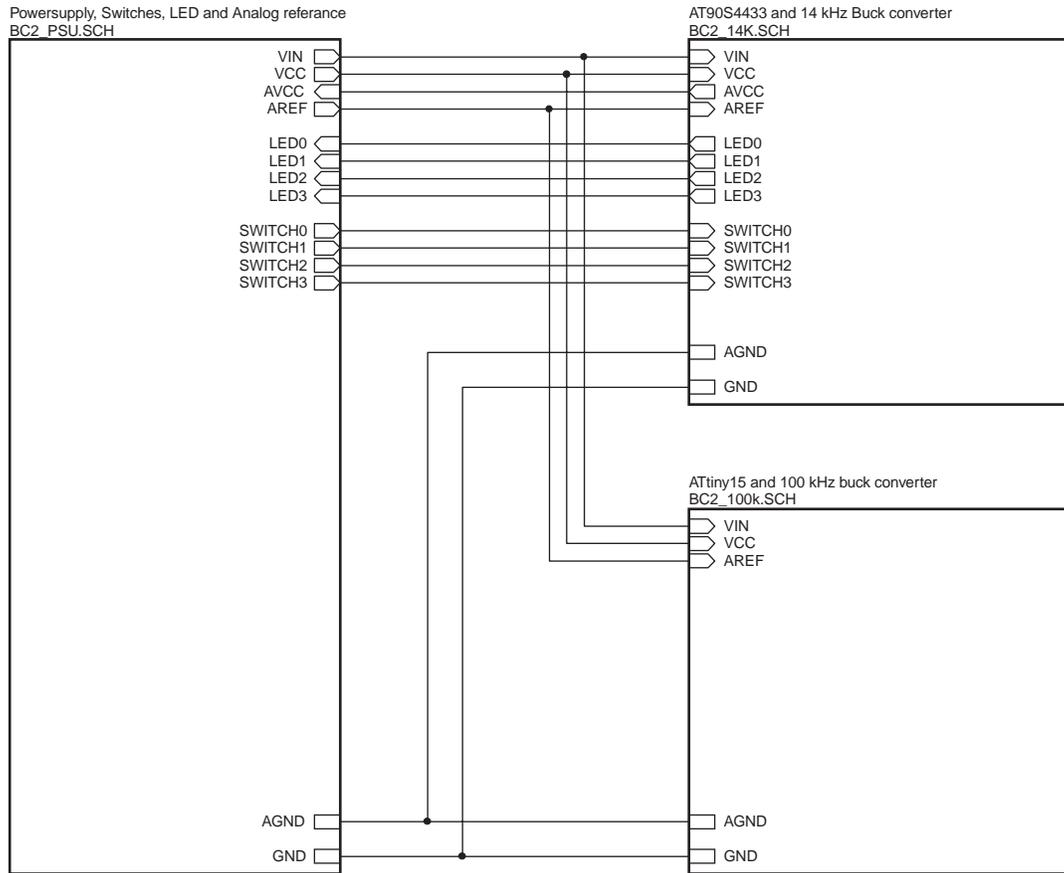
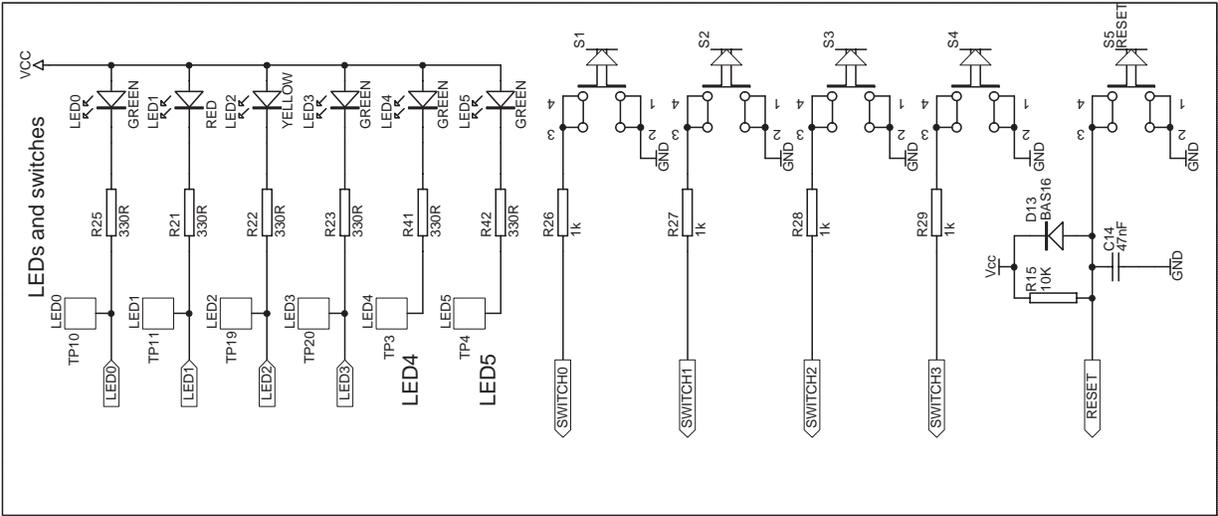


Figure 21. 电源和基准电压原理图



Testpoint TP3, TP4, TP10, TP11, TP19 and TP20 have no marking in the silkscreen. They are placed close to their respective resistors making it easy to (if desired) cut the track and patch the LED to an other function.

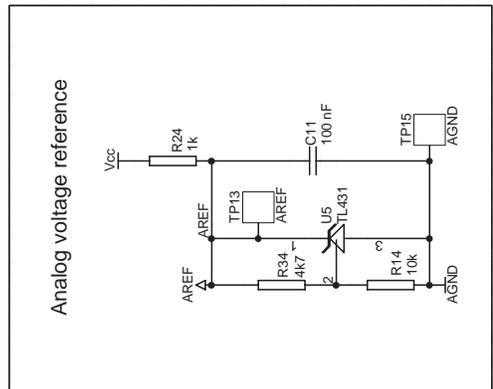
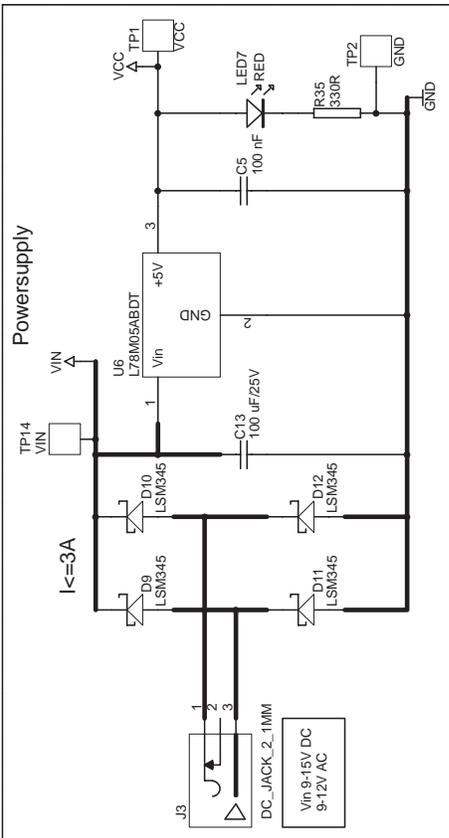


Figure 22. ATtiny15 及 100 kHz Buck 变换器原理图

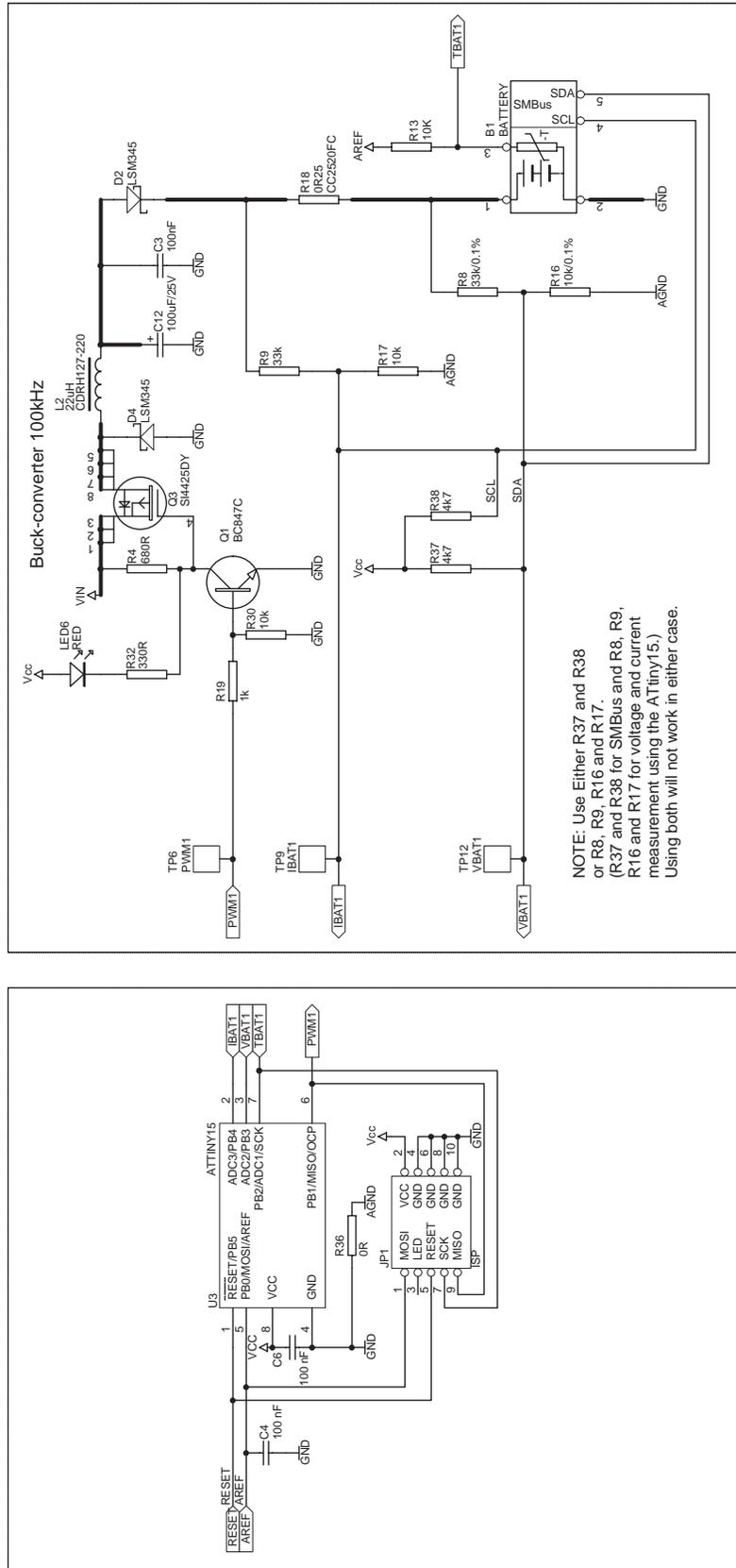
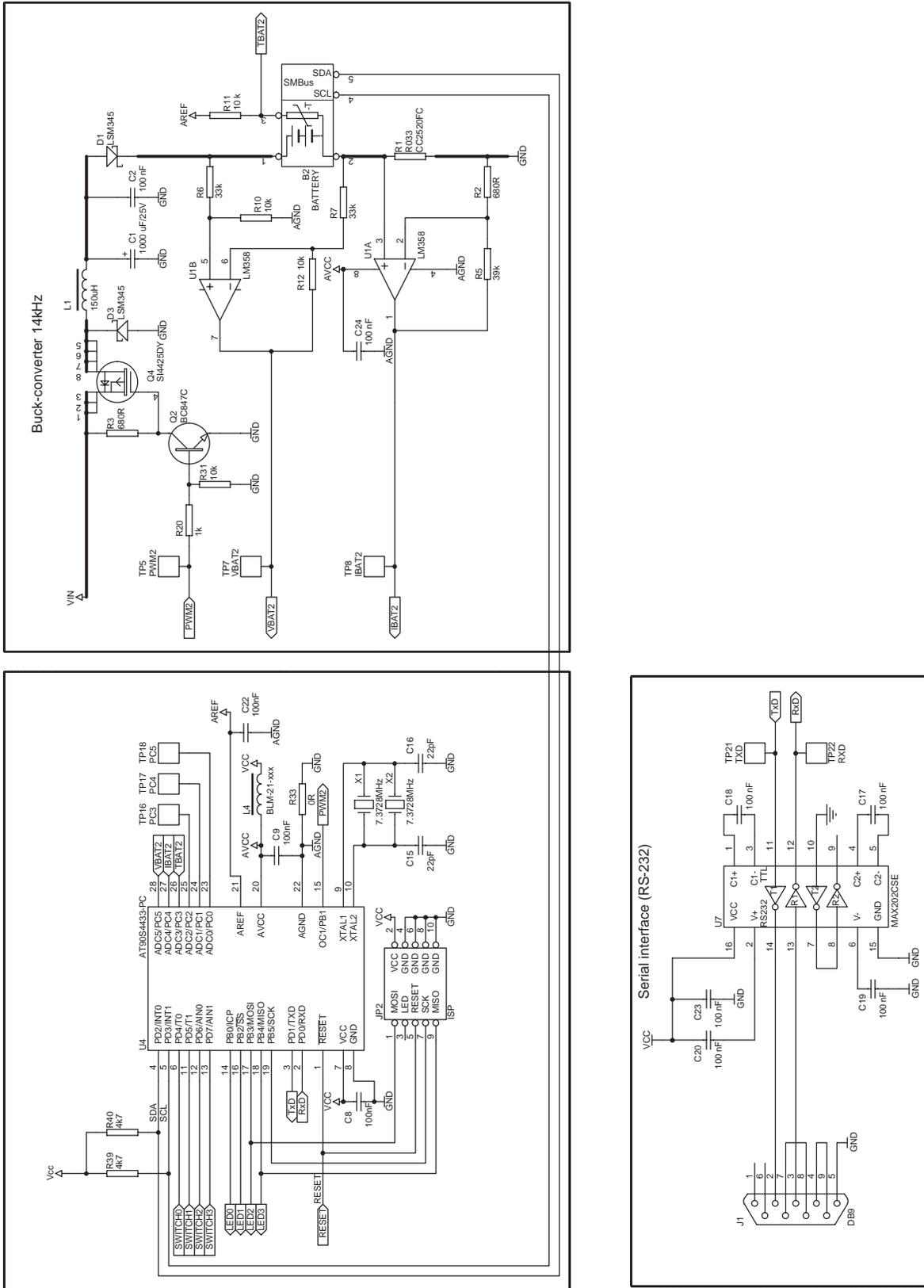


Figure 23. AT90S4433 及 14 kHz Buck 变换器原理图



附录 2: 电源

下面的原理图为电源部分，它为电池充电器提供 +15V，为 AVR 处理器提供 +5V。

充电器的电源以 Power Integration 公司的 TOP224 为架构。由于采用了 flyback 技术，电源更加紧凑和有效。输入电压为 $85 V_{AC} \sim 265 V_{AC}$ (50 - 60 Hz)。

Figure 24. 电源原理图

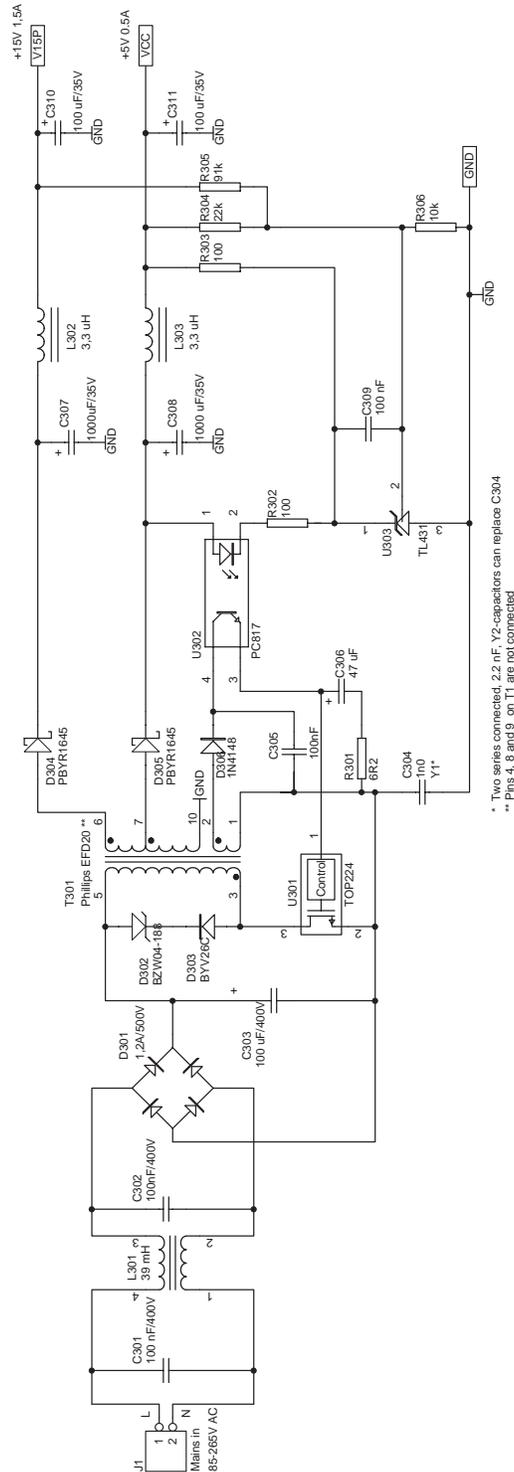


Table 5. 电源部分的器件列表

器件	类型	说明
R301	6,2Ω	C306 的串联电阻 (U301 的电源)
R302	100Ω	光耦的的串联电阻
R303	100Ω	电压基准的的串联电阻
R306	10 kΩ	反馈电阻
R304	22 kΩ	反馈电阻 (5V)
R305	91 kΩ	反馈电阻 (15V)
C304	1n0/Y1	Y1 电容 (可以用 2 * 2.2 nF Y2 电容代替)
C305	100 nF	
C309	100 nF	
C301	100 nF/400V	X 电容
C302	100 nF/400V	X 电容
C310	100 μF/35V	后级 LC 滤波器
C311	100 μF/35V	后级 LC 滤波器
C303	100 μF/400V	原边电容
C307	1000 μF/35V	
C308	1000 μF/35V	
L302	3.3 μH	后级 LC 滤波器
L303	3,3 μH	后级 LC 滤波器
L301	33 mH	输入噪声抑制器
D301	1.2A/500V	整流桥
D302	P6KE200	钳位 Zener 二极管
D303	BYV26C	钳位二极管的阻隔二极管
D304	PBYR1645	整流二极管 - 15V
D305	PBYR1645	整流二极管 - 5V
D306	1N4148	偏置电压的整流二极管 /U301 的电源
U301	TOP224	
U302	TL431	基准电压源
U303	PC817	光耦
T301	Phillips EFD20	变压器

变压器 T301 以 Philips 的 EFD20 骨架构成。原边绕组和偏置绕组使用 AWG26 (0.40 mm) 的线；次级绕组使用 AWG20 (0.80 mm) 的线。初级绕组和偏置绕组与两个次级绕组用绝缘带隔离开。5V 绕组为 15V 绕组的一部分。绕制方向必须遵照原理图给出的方向。

Table 6. 电源变压器绕组

绕组	圈数	线型
初级绕组	86	AWG26
偏置绕组	8	AWG26
次级绕组 (5V)	4	AWG20
次级绕组 (15V)	8 (4+4)	AWG20



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe

Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory

Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Microcontrollers

Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Atmel Smart Card ICs
Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Atmel Heilbronn
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Data- com

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

© Atmel Corporation 2002.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.