

电池快速充电指南—— 第2部分

Franco Contadini, 主管工程师
Alessandro Leonardi, 现场销售客户经理

“[电池快速充电指南——第1部分](#)”介绍了有关快速充电电池系统设计的一些挑战。通过在电池包中实现电量计功能，原始设备制造商(OEM)可以设计智能快速充电器，从而提高系统灵活性，更大限度地降低功耗，确保安全充电/放电，并改善整体用户体验。在第2部分中，我们将详细探讨如何使用评估套件和树莓派板实现电池并联的快速充电系统。

评估1S2P架构

评估简单充电系统并测试其功能，通常可以使用评估套件来完成。这些套件包括配置充电系统所需的所有硬件和软件应用，以及基于图形用户界面(GUI)的工具和API。

但相应地，包含多个单元的复杂系统的相关评估工作也更加繁琐。复杂系统中可能有多个器件需要进行表征。开发人员将需要编写一些软件代码来读取系统不同部分生成的信号，对其进行分析，并采取行动。[MAX17330](#)可帮助管理包含两节锂离子电池的并联电池快速充电系统。如数据手册所述，[MAX17330](#)可用于同时对两节锂离子电池进行充电和控制。该系统需要两个[MAX17330](#) IC，每个IC管理一节锂离子电池，以及一个能够即时调整输出电压的降压转换器（如[MAX20743](#)）。

该系统还需要使用一个微控制器来配置和管理电池充电，以及处理两个IC之间的通信。本文选择的树莓派板是系统测试中普遍使用的平台，此外我们选用Python作为编程语言。树莓派通过I²C管理通信，并记录有助于评估和调试的重要系统参数，包括充电电流、电池电压和电池荷电状态(SOC)。这些数值均存储在Excel文件中，方便进行离线分析。

测试1S2P架构

本节将介绍如何测试充电器和电量计([MAX17330](#))。本节还会说明并联充电可达到的实际性能。为了获得更大的灵活性和可控性，该器件由微控制器通过I²C进行编程。

图1显示了1S2P系统架构以及评估两节并联电池充电所需的连接。树莓派控制三个EVKIT：一个[MAX20743](#)EVKIT（降压转换器）和两个[MAX17330](#)EVKIT（充电器+电量计）。数据记录在Excel文件中。

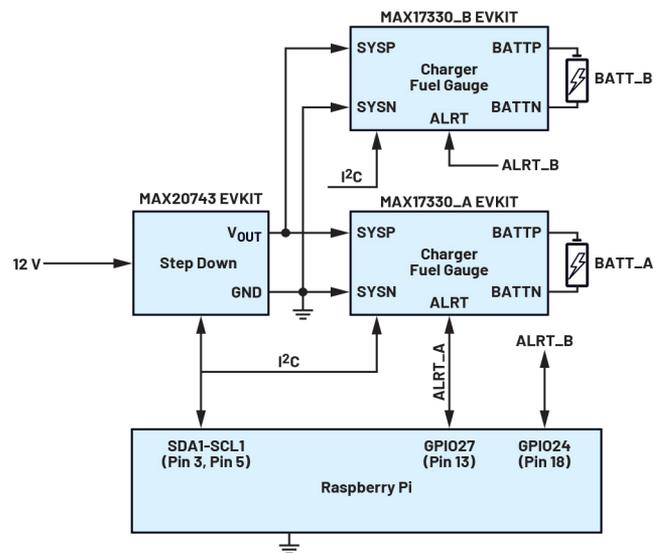


图1. 使用树莓派的1S2P充电系统评估架构

可从[MAX17330](#)产品页面的“工具和仿真”选项卡中下载并使用基于GUI的[MAX17330](#)评估套件软件。使用配置向导（从“器件”选项卡中选择）可为[MAX17330](#)生成初始化文件(.INI)。INI文件中包含寄存器地址/寄存器值格式的器件寄存器初始化信息。微控制器需使用该文件来逐个配置[MAX17330](#)中的寄存器。

[MAX17330](#)EVKIT数据手册详细说明了生成初始化文件所需的各个步骤。图2所示的配置用于启动并联充电。接下来可按图3中的配置启用步进充电。图4显示了基于图3配置步进充电后的预期步进充电曲线。

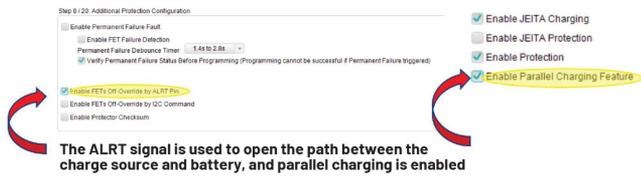


图2. 配置MAX17330进行并联充电



图3. 启用步进充电

MAX20734降压转换器可在需要时提高两个MAX17330EVKIT上的电压。MAX20734降压转换器根据地址0x21处的内部寄存器值改变输出电压。降压转换器可以通过I²C控制，已编写一个Python类来执行此操作。

最后，如图5所示，MAX20743EVKIT输出分压器被修改，输出范围为3 V至4.6 V（使用的值为R6 = 4K7和R9 = 1K3）。

表1. MAX20743基于寄存器0x21的转换输出电压

0x21寄存器值	电压
0x014E	3 V
0x0150	3.05 V
0x0158	3.1 V
0x015C	3.15 V
0x0162	3.2 V
0x0166	3.25 V
0x016E	3.3 V
0x0172	3.35 V
0x0178	3.4 V
0x017C	3.45 V
0x0182	3.5 V
0x0188	3.55 V
0x018E	3.6 V
0x0192	3.65 V
0x019E	3.7 V
0x01A4	3.75 V
0x01A9	3.8 V
0x01AE	3.85 V

表1. (续)

0x21寄存器值	电压
0x01B4	3.9 V
0x01BA	3.95 V
0x01BF	4 V
0x01C4	4.05 V
0x01CB	4.1 V
0x01D1	4.15 V
0x01D6	4.2 V
0x01DC	4.25 V
0x01E2	4.3 V
0x01E8	4.35 V
0x01ED	4.4 V
0x01F3	4.45 V
0x01F8	4.5 V
0x01FE	4.55 V
0x0204	4.6 V

从表1可以得出如下曲线：

$$Register = 0 \times 014e + \left(\frac{x - 3}{0.1 \times 11} \right)$$

其中，x为要在输出端施加的电压。虽然这种方法会有轻微误差，但也是根据电压估算所需寄存器值的好方法。

上电与初始化

当MAX17330首次连接电池时，默认寄存器值设置强制IC进入关断状态。要唤醒器件，请按PKWK按钮。这将使临时保护MOSFET短路，从而唤醒两个MAX17330EVKIT。

接下来，树莓派需要通过I²C与所有三个器件通信。小心地初始化I²C硬件，避免器件地址冲突。默认情况下，两个MAX17330EVKIT使用相同I²C地址。第一步是更改两个电量计之一的地址。

MAX17330兼有易失性和非易失性寄存器，非易失性寄存器以“n”前缀标识。这也导致产生一对节点地址：6Ch（易失性寄存器）和16h（NV寄存器）。

改变MAX17330器件节点地址的方法有两种：

- ▶ 使用I²C Sid字段设置nPackCfg NV寄存器。此更改可以利用配置向导设置。参见表3。
- ▶ I²C Cmd寄存器支持动态更改I²C总线。参见表4。

为了便于使用，我们采用第二种方法来改变地址，这样可以使用同一INI文件来初始化两个器件。生成两个器件的通用设置可以简化器件配置，并消除有关手动输入地址的用户错误。

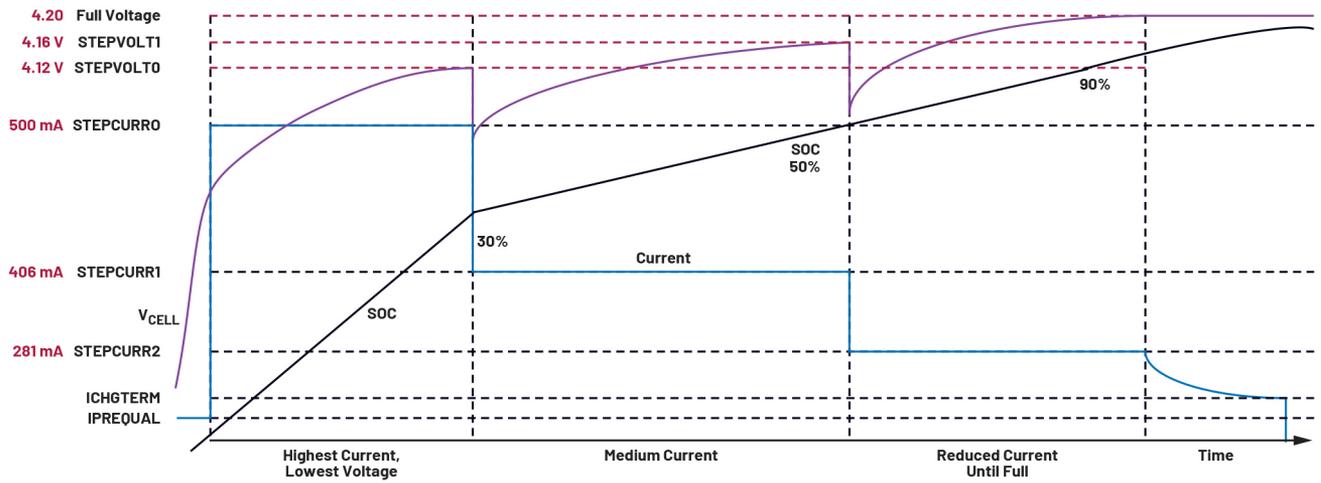


图4. 基于图3来配置步进充电的预期步进充电曲线

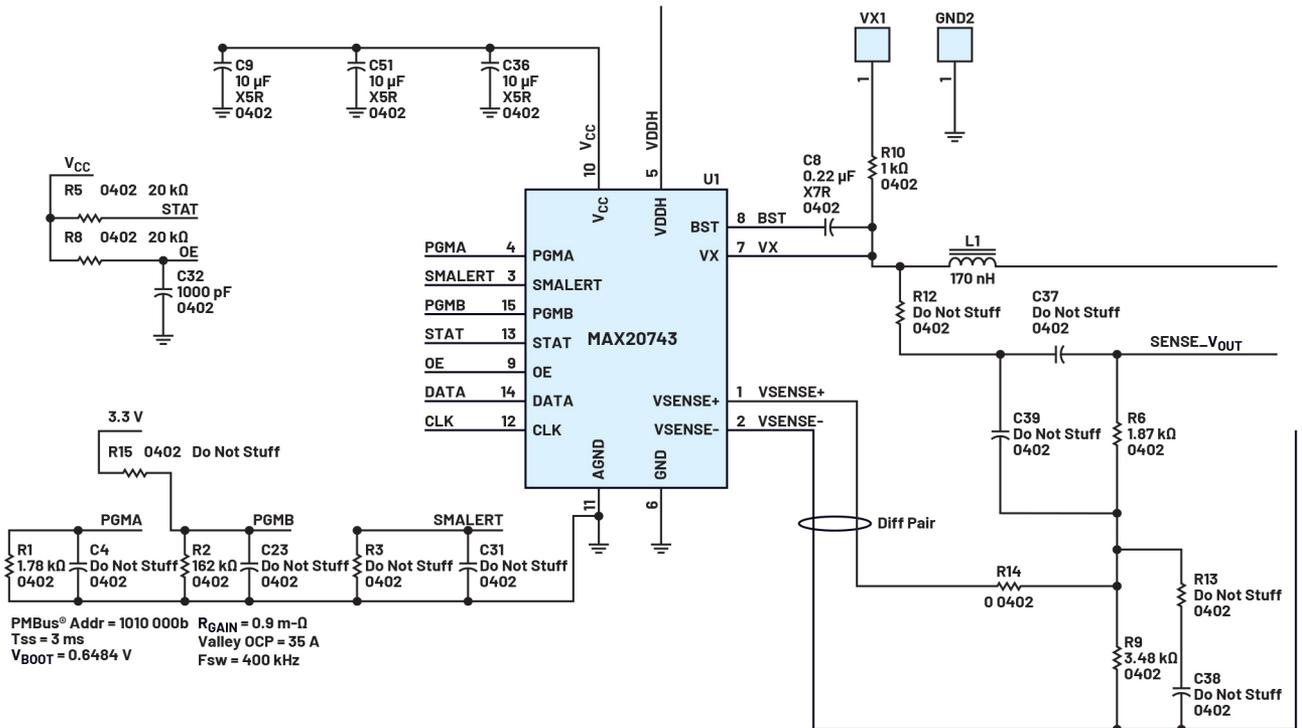


图5. 输出分压器已被修改，输出范围为3V至4.6V (R6 = 4 K7且R9 = 1 K3)

表2. MAX17330寄存器

寄存器页	锁	说明	2-Wire节点地址	2-Wire协议	2-Wire外部地址范围
00 h		Modelgauge M5 EZ数据块	6通道	I ² C	00 h - 4 Fh
01 h - 04 h	锁2				
05 h - 0Ah		保留			
0 Bh	锁2	Modelgauge M5 EZ数据块 (续)	6通道	I ² C	80 h - BFh
0 Ch	SHA	SHA存储器	6通道	I ² C	C0h - CFh
0 Dh	锁2	Modelgauge M5 EZ数据块 (续)	6通道	I ² C	D0h - DFh
0 Eh - 0 Fh		保留			
10 h - 17 h		SBS数据块	16通道	SBS	00 h - 7 Fh
18 h - 19 h	锁3	Modelgauge M5 EZ非易失性存储器模块	16通道	I ² C	80 h - EFh
1 Ah - 1 Bh	锁1	寿命记录和配置非易失性存储器模块			
1 Ch	锁4	配置非易失性存储器模块			

表3. nPackCfg (1B5h)寄存器格式

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	S_Hib	THCfg	THType	000			0	ParEn	I ² Csid			0001			

表4. I²CCmd (12Bh)寄存器格式

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0										GoToSID		0		IncSID	

由于两个MAX17330器件共用同一I²C总线，因此该程序要求将一个器件的ALRT信号设置为低电平，并将另一个设置为高电平。

表5. I²C ALRT设置

GoToSID	警报高	警报低
	主要/次要地址	主要/次要地址
0b00	ECh/96h	6Ch/16h
0b01	64h/1Eh	ECh/96h
0b10	E4h/9Eh	64h/1Eh
0b11	6Ch/16h	E4h/9Eh

表4中的数据来自MAX17330数据手册，显示了I²CCmd寄存器如何根据ALERT GPIO引脚值动态更改器件地址。在这种情况下，可使用GoToSID和IncSID字段更改I²C地址：

- ▶ Set ALRT_A logic low
- ▶ Set ALRT_B logic high
- ▶ Write I²CCmd = 0 × 0001 → MAX17330_A address remains at 6Ch/16h
→ MAX17330_B address set to ECh/96h

每个器件都分配有唯一的地址后，整个系统便可以由单个微控制器控制。

下面是微控制器完成I²C配置的脚本。这将是系统初始化的一部分。

- ▶ Load .INI file
- ▶ Assert ALRT_A and ALRT_B to keep the path between SYSP and BATT open
- ▶ Read V_{BATT-A} and V_{BATT-B}
- ▶ V_{MAX} = max (V_{BATT-A}, V_{BATT-B})
- ▶ Set V_{OUT} = V_{MAX} + 50 mV
- ▶ Release ALRT_A and ALRT_B
- ▶ Set nProtCfg.OvrEn = 0 to use ALRT as Output

参见表6。

非易失性空间中的某些寄存器需要重启固件才能使更改生效。因此，需要执行以下步骤：

- ▶ 置位Config2.POR_CMD以重启固件

参见表7。

接下来，我们需要启用充电器的中断：

- ▶ 设置 (Config.Aen和Config.Caen) = 1

参见表8。

现在器件已初始化。

表6. nProtCfg (1D7h)寄存器格式

D15	D14	D13	D12	D11	D10	D9	D8
ChgWDTEn	nChgAutoCtrl	FullEn	SCTest		CmOvrdEn	ChgTestEn	PrequalEn
D7	D6	D5	D4	D3	D2	D1	D0
保留	PFEEn	DeepShpEn	OvrdEn	UVRdy	FetPFEEn	BlockDisCEn	DeepShp2En

表7. Config2 (0ABh)寄存器格式

D15	D14	D13	D12	D11	D10	D9	D8
POR_CMD	0	AtRtEn	0	0	0	0	0
D7	D6	D5	D4	D3	D2	D1	D0
dSOCen	TAIrtEn	0	1	DRCfg		CPMode	BlockDis

表8. Config (00Bh)寄存器格式

D15	D14	D13	D12	D11	D10	D9	D8
0	SS	TS	VS	0	PBen	DisBlockRead	ChgAutoCtrl
D7	D6	D5	D4	D3	D2	D1	D0
SHIP	COMMSH	FastADCen	ETHRM	FTHRM	Aen	CAen	PAen

记录数据和中断

我们需要能够读取寄存器以记录数据，并检查ALERT GPIO线上是否已生成中断。我们可以使用如下脚本：

- ▶ 设置500 ms定时器
- ▶ $V_{MIN} = \min(V_{BATT-A}, V_{BATT-B})$
- ▶ $V_{sys_min} = nVEmpty[15:7]$
- ▶ `CrossCharge = False`
- ▶ `If ($V_{MIN} < V_{sys_min}$) → CrossCharge = True`
评估最小电池电压是否超过系统的最小工作电压
- ▶ `If FProtStat.IsDis = 0`
检测到充电信号
- ▶ `Clear Status.AllowChgB`
向所有电池表明充电器存在
- ▶ `If ($V_{BATT} > V_{MIN} + 400\text{ mV}$ and !Cross Charge)`
确定要阻止哪个电池以避免交叉充电

`Config2.BlockDis = 1`

`else`

`Config2.BlockDis = 0`

如果低电量电池远低于高电量电池，则允许放电

参见表9、10和11。

当MAX17330置位ALRT信号时，主机将执行以下操作：

`Read Status register data`

`If Status.CA is set`

`Read ChgStat register`

`If ChgStat.Dropout = 1 → increase V_{OUT}`

`If (ChgStat.CP or ChgStat.CT) = 1 → decrease V_{OUT}`

`Clear Status.CA`

参见表12和13。

图6显示了从Excel文件的记录数据提取的并联充电曲线。请注意该曲线随步进充电曲线的变化情况。

作者简介

Franco Contadini拥有超过35年的电子行业从业经验。在从事电路板和ASIC设计工作10年之后，他成为现场应用工程师，为工业、电信和医疗客户提供支持，主要负责电源和电池管理、信号链、加密系统和微控制器。Franco撰写了多篇关于信号链和电源的应用笔记和技术文章。他在意大利热那亚ITIS主修电子学。

Alessandro Leonardi是ADI米兰分公司的客户经理。他拥有米兰理工大学的电子工程学士和硕士学位。毕业后，他参加了ADI公司的现场应用培训生项目。

在线支持社区



访问ADI在线支持社区，中文技术论坛

与ADI技术专家互动。提出您的棘手设计问题、浏览常见问题解答，或参与讨论。

请访问ez.analog.com/cn



如需了解区域总部、销售和分销商，或联系客户服务和技术支持，请访问analog.com/cn/contact。

向我们的ADI技术专家提出棘手问题、浏览常见问题解答，或参与EngineerZone在线支持社区讨论。请访问ez.analog.com/cn。

©2023 Analog Devices, Inc. 保留所有权利。商标和注册商标属各自所有人所有。

“超越一切可能”是ADI公司的商标。

TA24441sc-5/23



请访问analog.com/cn