

How to Design Battery Charger Applications that Require External Microcontrollers and Related System-Level Issues

Notebook computers increasingly require complex battery charging algorithms and systems. This article provides information and background on lithium-ion (Li+), nickel-cadmium (NiCd), and nickel-metal-hydride (NiMH) batteries and related system-level switch-mode and linear battery chargers. These voltage regulators and current regulators are controlled by external microprocessors like the 8051 or Microchip PIC, and examples are provided with these controllers. An overview of requirements for charging common battery chemistries with Maxim battery charger ICs is provided, along with a discussion of system-level trade-offs and firmware design tips, and a list of World Wide Web engineering resources.

The previous issue of Maxim's Engineering Journal (Vol. 27) discussed new developments in stand-alone battery chargers. This second article of a two-part series explores the system-level issues in applying battery-charger ICs.

Over the past five years, market pressures on portable equipment have transformed the simple battery charger into a sophisticated switch-mode device capable of charging an advanced battery in 30 minutes. This development also marks a departure from the self-contained, stand-alone charger ICs of only a few years ago. Some of those ICs included considerable intelligence: enough to handle the complex task of fast charging advanced batteries.

Maxim still manufactures stand-alone charger ICs, but market demand has changed recently. Today's battery-charger subsystems regulate charging voltage and current using the intelligence of an external microcontroller (μC), usually available elsewhere in the system. This approach achieves low cost in high-volume applications and allows the greatest flexibility in tailoring the charger to a specific application.

All necessary intelligence once resided in the battery-charger controller IC itself, but now the system designer must implement a charging algorithm and write the associated firmware. This article provides the information and background necessary to implement charger systems based on Maxim's wide range of battery-charger ICs for all popular chemistries.

The following discussion presents an overview of the requirements for charging common battery chemistries with Maxim battery-charger ICs. It addresses system-level trade-offs and firmware design tips, and lists World Wide Web resources available to designers. The discussion closes with design examples based on two common μ Cs: the 8051 and the Microchip PIC. Either example can serve as a base for further development of custom charger circuitry.

Overview of Battery-Charging Techniques

Four rechargeable battery chemistries are in practical use today: Nickel Cadmium (NiCd), Nickel Metal Hydride (NiMH), Gelled Lead-Acid (PbSO₄), and Lithium-Ion (Li⁺). The trade-offs to be made among these chemistries are beyond the scope of this article, but the *References* section provides access to such information.

Caution: consult the battery manufacturer for specific recommendations. The information presented here is intended only as an overview of charging requirements for various cell chemistries.

This section describes general charging techniques and limitations for the four common chemistries. For additional details and background, see the Maxim data sheets and other reference material cited at the end of the article.

Fast battery charging has several phases, as explained in the text and by the state diagram for a generic charger (Figure 1).

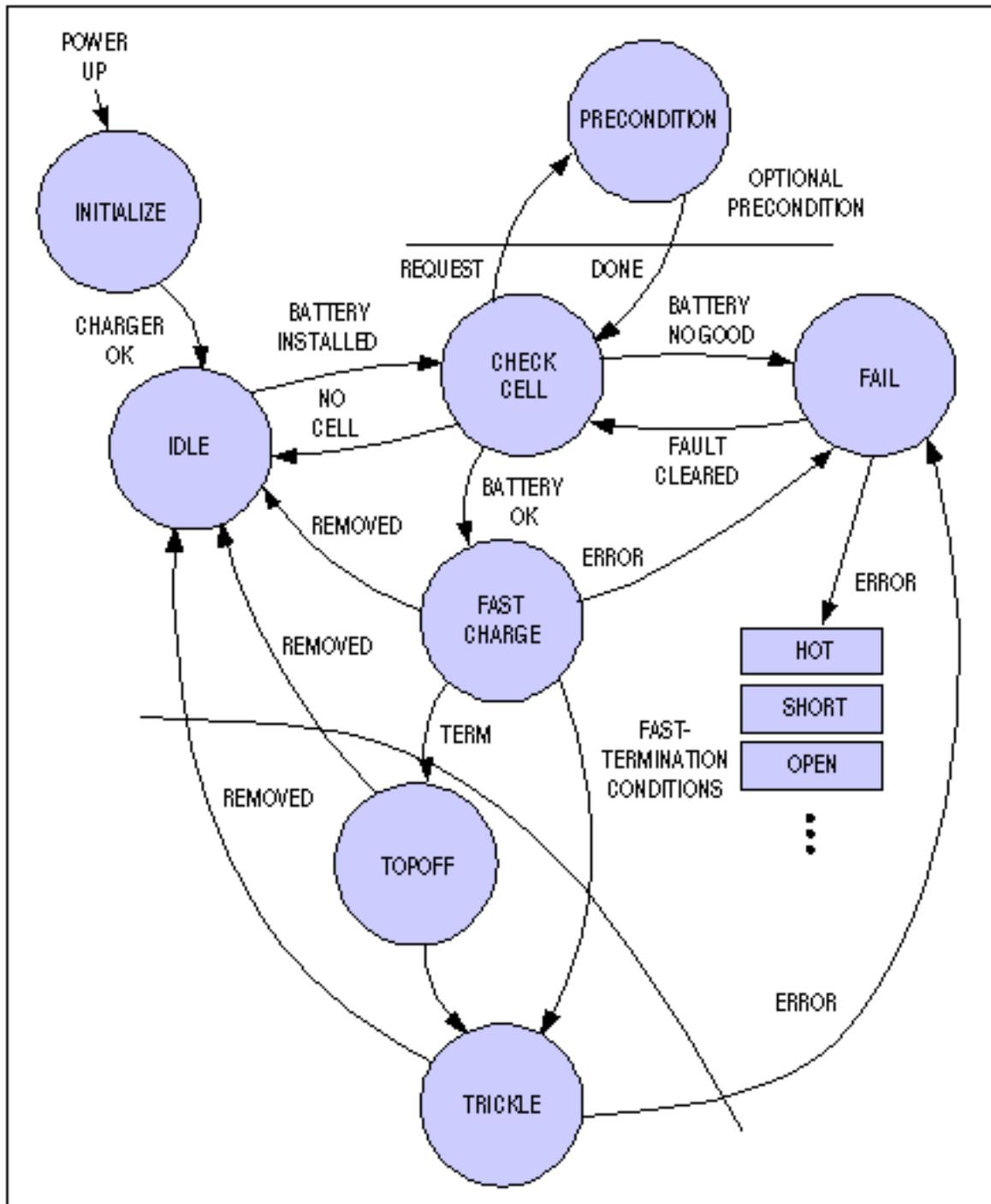


Figure 1. Generic charger-state diagram

Initialization

Though not a part of the actual charging procedure, initialization is an important stage in the process. The charger initializes itself and performs its own self-test. A charge can be interrupted by a power failure and consequent reinitialization. Without a smart battery or some type of time-stamped, nonvolatile storage, such events can occur unnoticed. Most chargers reinitialize themselves fully after a power failure. If overcharging is an issue, the charger can then execute a special self-test sequence to determine if the battery is already charged. A battery present on power-up, for example, should trigger such an action.

Several circumstances can allow this initialization to cause charging problems. A fixed-time charger, for example, applies charge to a battery for a fixed interval of four hours. If a power failure occurs three hours and 59 minutes into the charge, the charger starts another four-hour charge, giving the battery a four-hour overcharge. This treatment can damage the battery, and it is one reason fixed-time charging is seldom used. The example also shows why the charger should monitor battery temperature or use other termination methods as a backup measure.

Cell Qualification

This phase of the charging procedure detects when a battery is installed and whether it can be charged. Cell detection is usually accomplished by looking for voltage on the charger terminals while the charger source is off, but that method can pose a problem if the cells have been deeply cycled and are producing little voltage. As an alternative, the charger often looks for a thermistor or shorting jumper rather than the cell itself. The presence of this hardware can also serve to identify the battery pack. Smart batteries, on the other hand, conduct a rich exchange of serial data with the battery pack, usually providing all the necessary charging parameters over a specialized I²C™like protocol called the System Management Bus (SMBus™).

Once the charger determines that a cell is installed, it must determine if the cell is good. During this subphase (qualification), the cell is checked for basic functioning: open, shorted, hot, or cold. To test whether or not a cell is chargeable, some chargers-lead-acid types especially-apply a light charging current (about one-fifth of the fast rate) and allow the cell a fixed amount of time to reach a specified voltage. This technique avoids the problem of false rejects for deeply cycled PbSO₄ batteries, and with the battery manufacturer's approval, it can be used for other chemistries as well.

A check of the ambient and cell temperatures is also a part of the qualification phase. When a charger detects high or low temperature, it usually waits a predetermined interval for the temperature to return to nominal. If this doesn't happen within the allotted time, the charger reduces the charging current. This action in turn reduces battery temperature, which increases efficiency. Finally, the cells are checked for opens and shorts. Open cells are easily detected, but a shorted-cell indication requires confirmation in order to avoid false failure indications. If all of these checks are satisfactory, the cell can be charged, and the state is advanced as shown in Figure 1.

Preconditioning Phase (optional)

Some chargers (primarily those for NiCd batteries) include an optional preconditioning phase in which the battery is fully discharged before recharging. Full discharge reduces each battery's voltage level to 1V per cell and eliminates dendritic formations in the electrolyte, which cause what is often falsely labeled the memory effect. This so-called memory effect refers to the presence of dendritic formations that can reduce the run life of a cell, but a complete charge and discharge cycle sometimes eliminates the problem.

Preconditioning can be accomplished before each charge, or it can follow an indication (by load test or other operation) that more than half of the cell's charge remains. Preconditioning can last from one to ten hours. Discharging a battery in less than one hour is not generally recommended. Fast preconditioning raises the practical problem of what to do with heat dissipated by the load resistor. Nor is preconditioning for longer than ten hours usually recommended unless it can be initiated manually upon detection of reduced capacity. Confusion and misunderstanding surround the NiCd "memory effect," so the designer should avoid putting a button on the charger to counteract it.

Fast-Charge Phase and Termination

Fast-charge and termination methods used depend on cell chemistry and other design factors. The following discussion covers fast-charging techniques widely used for today's common battery chemistries. For specific guidelines and recommendations, consult the battery manufacturer's applications department.

NiCd and NiMH Cells

Fast-charging procedures for NiCd and NiMH batteries are very similar; they differ primarily in the termination method used. In each case, the charger applies a constant current while monitoring battery voltage and other variables to determine when to terminate the charge. Fast-charge rates in excess of 2C are possible, but the most common rate is about C/2. Because charging efficiency is somewhat less than 100%, a full charge at the C/2 rate requires slightly more than two hours.

While constant current is applied, the cell voltage rises slowly and eventually reaches a peak (a point of zero slope). NiMH charging should be terminated at this peak (the $0\Delta V$ point). NiCd charging, on the other hand, should terminate at a point past the peak: when the battery voltage first shows a slight decline ($-\Delta V$) (Figure 2). Cell damage can result if fast charge continues past either battery's termination point.

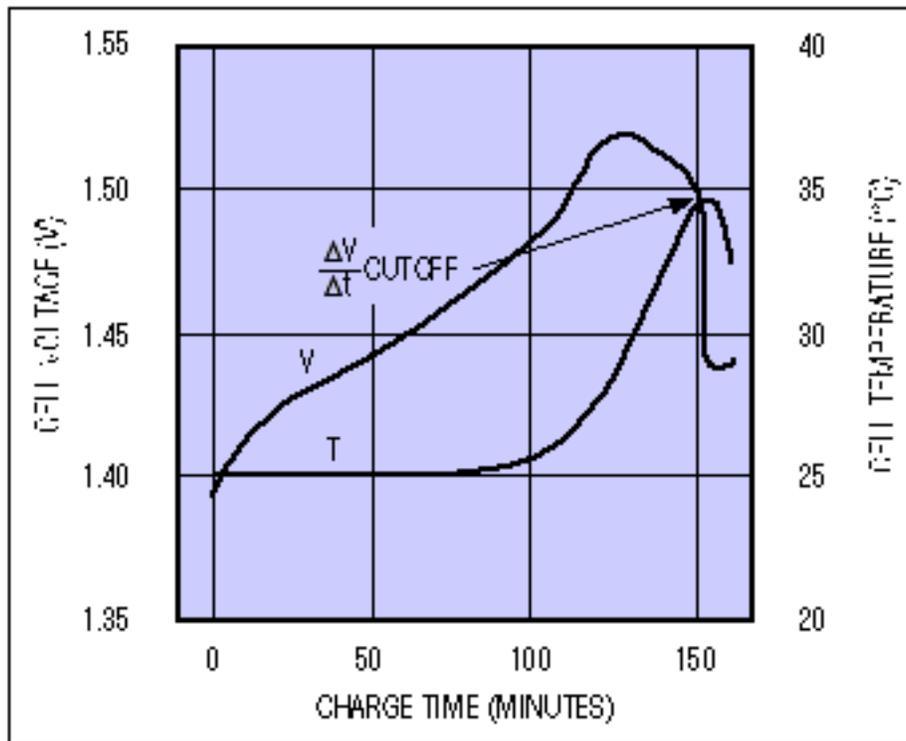


Figure 2. NiCd battery-charging characteristics at C/2 rate

At rates exceeding C/2 (resulting in a charge time of no more than two hours), the charger also monitors the cell's temperature and voltage. Because cell temperature rises rapidly when a cell reaches full charge, the temperature monitor enables another termination technique.

Termination on this positive temperature slope is called ΔT termination. Other factors that can trigger termination include charging time and maximum cell voltage. Well-designed chargers rely on a combination of these factors.

Note: Because certain effects that appear when a cell first begins charging can imitate termination conditions, chargers usually introduce a delay of one to five minutes before activating slope-detection termination modes. Also, charge-termination conditions are difficult to detect for rates below C/8, because the voltage and temperature slopes of interest ($\Delta V/\Delta t$ and $\Delta T/\Delta t$) are small and comparable to other system effects. For safety during a fast charge, the hardware and software in these systems should always err on the side of *early* termination.

Lithium-Ion Cells

Li⁺ battery charging differs from the nickel-chemistry charging schemes. A top-off charge can follow to ensure maximum energy storage in a safe manner. Li⁺ chargers regulate their charging voltage to an accuracy better than 0.75%, and their maximum charging rate is set with a current limit, much like that of a bench power supply (Figure 3). When fast charging begins, the cell voltage is low, and charging current assumes the current-limit value.

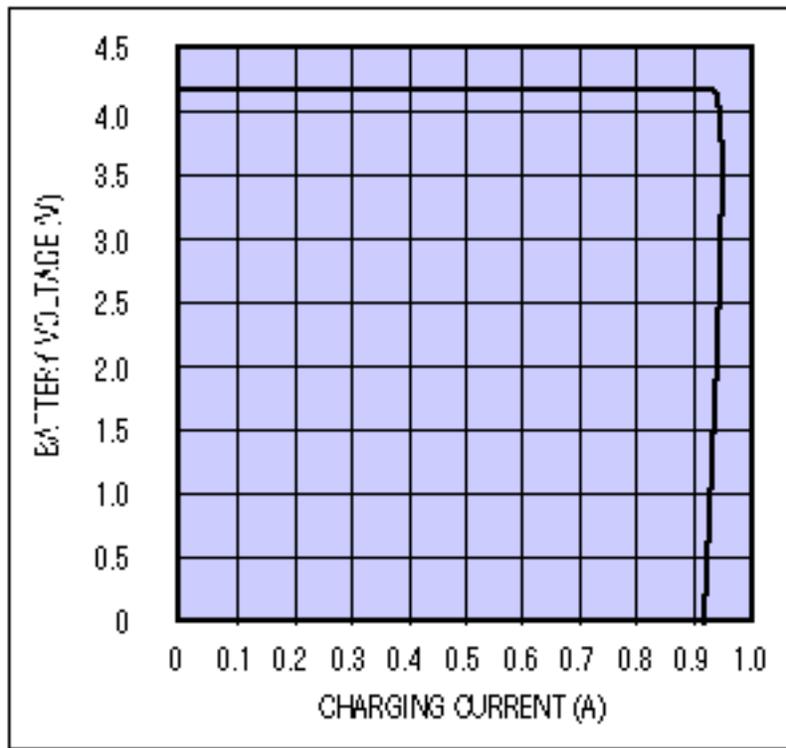


Figure 3. Li+ battery voltage vs. charging current

Battery voltage rises slowly during the charge. Eventually, the current tapers down, and the voltage rises to a float-voltage level of 4.2V per cell (Figure 4).

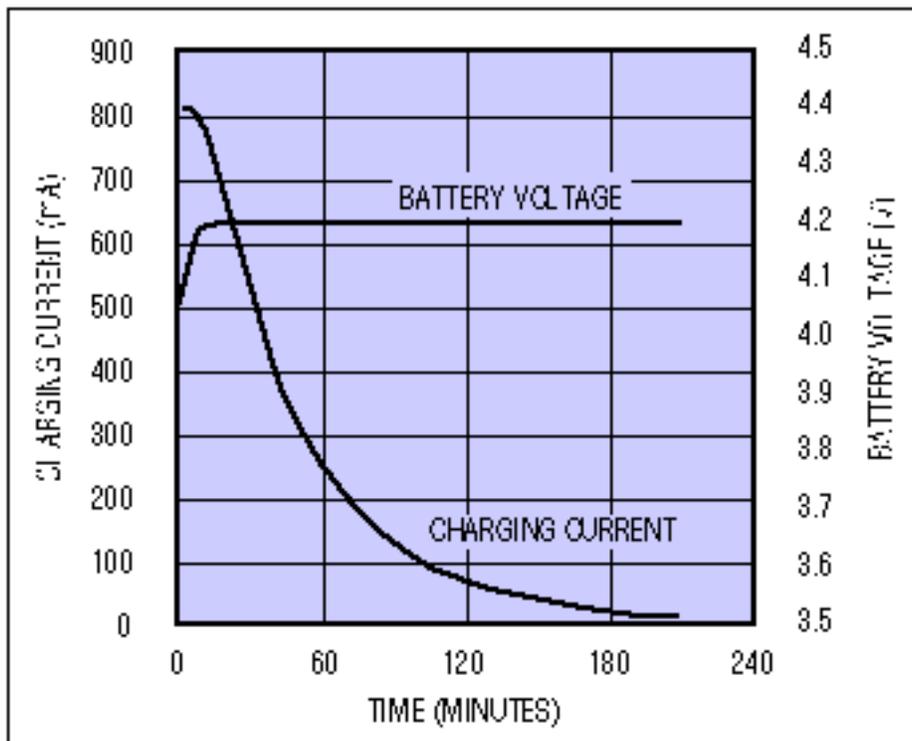


Figure 4. Li+ battery-charging profile

The charger can terminate charging when the battery reaches its float voltage, but that approach neglects the topping-off operation. One variation is to start a timer when float voltage is reached, and then terminate charging after a fixed delay. Another method is to monitor the charging current, and terminate at a low level (typically 5% of the limit value; some manufacturers recommend a higher minimum of 100mA). A top-off cycle often follows this technique, as well.

The past few years have yielded improvements in Li+ batteries, the chargers, and our understanding of this battery chemistry. The earliest Li+ batteries for consumer applications had shortcomings that affected safety, but those problems cannot occur in today's well-designed systems. Manufacturers' recommendations are neither static nor totally consistent, and Li+ batteries continue to evolve.

Lead-Acid Cells

PbSO₄ batteries are usually charged either by the current-limited method or by the more common and generally simpler voltage-limited method. The voltage-limited charging method is similar to that used for Li+ cells, but high precision isn't as critical. It requires a current-limited voltage source set at a level somewhat higher than the cell's float voltage (about 2.45V).

After a preconditioning operation that ensures that the battery will take a charge, the charger begins the fast charge and continues until it reaches a minimum charging current. (This procedure is similar to that of a Li+ charger). Fast charge is then terminated, and the charger applies a maintenance charge of V_{FLOAT} (usually about 2.2V). PbSO₄ cells allow this float-voltage maintenance for indefinite periods (Figure 5).

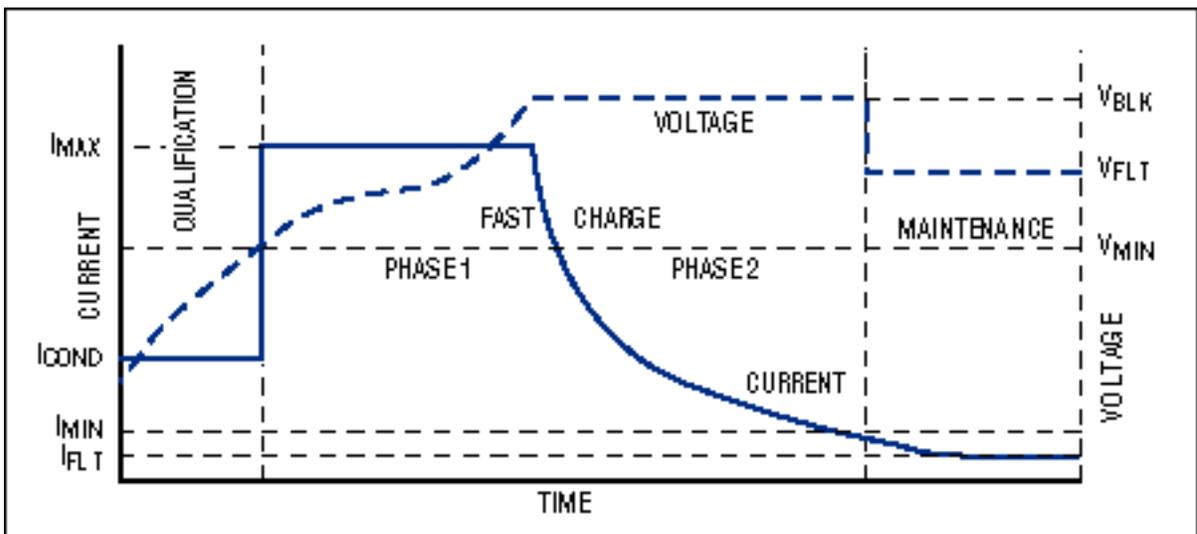


Figure 5. PbSO₄ battery-charging profile

At higher temperatures, the fast-charge current for PbSO₄ batteries should be reduced according to the typical temperature coefficient of 0.3% per degree centigrade. The maximum temperature recommended for fast charging is about 50° C, but maintenance charging can generally proceed above that temperature.

Optional Top-Off Charge (All chemistries)

Chargers for all chemistries often include an optional top-off phase. This phase occurs after fast-charge termination and applies a moderate charging current that boosts the battery up to its full-charge level. (The operation is analogous to topping off a car's gas tank after the pump has stopped automatically.) The top-off charge is terminated on reaching a limit with respect to cell voltage, temperature, or time. In some cases, top-off charge can provide a run life of 5% or even 10% above that of a standard fast charge. Extra care is advisable here: the battery is at or near full charge and is therefore subject to damage from overcharging.

Optional Trickle Charge (All chemistries except Li+)

Chargers for all chemistries often include an optional trickle-charge phase. This phase compensates for self-discharge in a battery. PbSO₄ batteries have the highest rate of self-discharge (a few percent per day), and Li⁺ cells have the lowest. The Li⁺ rate is so low that trickle charging is not required or recommended. NiCds, however, can usually accept a C/16 trickle charge indefinitely. For NiMH cells, a safe continuous current is usually around C/50, but trickle charging for NiMH cells is not universally recommended.

Pulsed trickle is a variation in which the charger provides brief pulses of approximately C/8 magnitude, with a low duty cycle that provides a typical average trickle current of C/512. Because pulsed-trickle charging applies to both nickel chemistries and lends itself well to the on/off type of microprocessor (μ P) control, it is used almost universally.

Generic Charging System

Before looking at specific circuit implementations, designers should become familiar with generic blocks and features (Figure 6). All fast chargers should include these block functions in some form. The bulk power source provides raw dc power, usually from a wall cube or brick. The current and voltage controls regulate current and voltage applied to the battery. For less-expensive chargers, the regulator is usually a power transistor or other linear-pass element that dissipates power as heat. It can also be a buck switching supply that includes a standard freewheeling diode for average efficiency or a synchronous rectifier for highest efficiency.

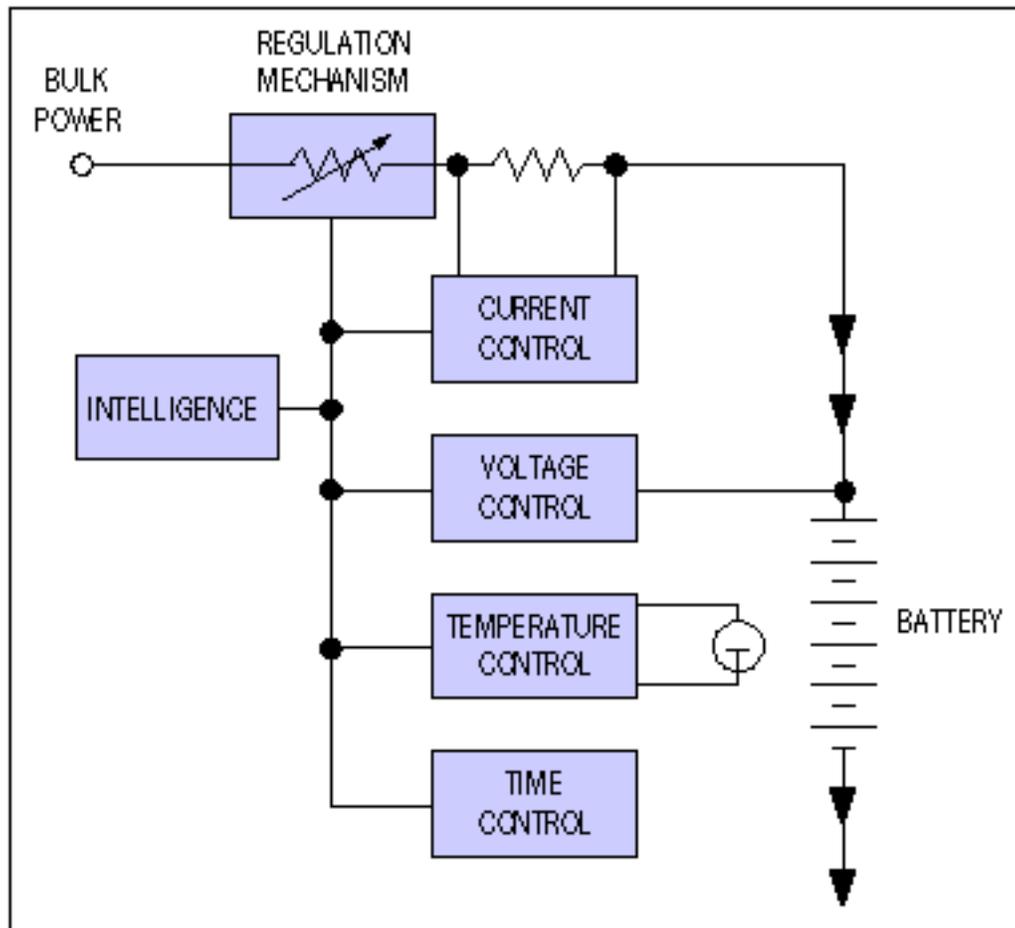


Figure 6. Generic charging-system block diagram

The blocks on the right in Figure 6 represent various measurement and control functions. An analog current-control loop limits the maximum current delivered to the battery, and a voltage loop maintains a constant voltage on the cell. (Note that Li+ cells require a high level of precision in the applied charging voltage.)

A charger's current-voltage (I-V) characteristic can be fully programmable, or it can be programmable in current only, with a voltage limit (or vice versa). Cell temperature is always measured, and charge termination can be based either on the level or the slope of this measurement. Chargers also measure charging time, usually as a calculation in the intelligence block.

This block provides intelligence for the system and implements the state machine previously described. It knows how and when to terminate a fast charge. Intelligence is internal to the chip in stand-alone charger ICs. Otherwise, it resides in a host μC , and the other hardware blocks reside in the charger IC. As mentioned previously, this latter architecture is the one preferred today.

Overview of Maxim's Charger Offerings

Maxim manufactures a broad selection of stand-alone and controller-type battery-charger ICs. The variety enables a system designer to make tradeoffs in performance, features, and cost. Table 1 lists these ICs by the battery chemistry supported, in their order of introduction, with the most recent models at the top.

Table 1. Overview of Maxim's Battery-Charger ICs

Part	Control Method	Standard Regulation Mode**	Features	Chemistry	Charge Rate	Charge Termination Method
MAX1547	µC control, SMBus	Synchronous switching	Smart-battery system, level 2 compliant, smart-battery charger with SMBus, Li+, independent I-V control	All	Programmed	Programmed
MAX1548	User	Synchronous switching	Analog-controlled version of MAX1647, high-accuracy switching, I/V source: Li+	All	Programmed	Programmed
MAX745	DAC or stand-alone	Synchronous switching	Advanced, low-cost, switch-mode Li+ charger, stand-alone, Li+ only	Li+	Constant voltage, Li+	Li+ float
MAX846A	DAC or stand-alone	Linear	Low-cost, universal charger, accurate reference for Li+, external CPU support, reset and regulator	All	Constant voltage, Li+, programmed	Li+ float or programmed

MAX1540	DAC or stand-alone	Synchronous switching	Analog-controlled, switch-mode current source, Li+ or universal	Li+, NiCd, NiMH	Fast, trickle, pulse-trickle, top-off	Programmed or Li+ stand-alone
MAX712	Stand-alone	Linear	Complete, low-cost NiMH with termination modes, max times, LED outputs. No Li+.	NiMH	Fast, trickle	0ΔV, max voltage, max temperature, max time
MAX713	Stand-alone	Linear	Complete, low-cost NiCd with termination modes, max times, LED outputs. No Li+.	NiCd	Fast, trickle	0ΔV, max voltage, max temperature, max time

* The use of a DAC and μC is also possible with the DAC-input types.

** All linear types can be used in a hysteretic switching mode for higher efficiency.

The choice between linear and switch-mode regulation constitutes a major design decision. Linear mode is less costly, but it dissipates power and gets hot. Heat may not be a problem in large desktop chargers, but it can be unacceptable in smaller systems such as a notebook PC. Synchronous switching regulators offer the highest efficiency (in the mid-90% range), which makes them suitable for the smallest systems, including cell phones. Some of the nonsynchronous switch-mode circuits listed also offer reasonable efficiency. In addition, most of the linear parts can be used in a moderately efficient hysteretic switching mode. (For details, consult the appropriate data sheet.)

The charger's level of autonomy poses another design decision. Stand-alone chargers, for example, are completely self-contained. The MAX712/MAX713 and have LED-control outputs for the user's end equipment as well.

Other devices can stand alone or can operate with a digital-to-analog converter (DAC) and μP . They include the MAX1640/MAX1641, MAX846A, and MAX745. The MAX1640, a voltage-limited current source intended primarily for charging nickel-chemistry batteries, includes a charge timer and pulse-trickle circuitry. It has stand-alone features and operates with a high-efficiency synchronous switching regulator or (for lower cost applications) a standard switcher.

The MAX846A and MAX745 are both capable of stand-alone operation in charging Li+ batteries, and they include the high-accuracy reference and independent voltage and current control necessary for universal controllers. The MAX846A is a linear type, and the MAX745 is a synchronous-switching type. Though either can stand alone, they usually operate with a μC that provides limited control of the charging process. LED illumination and fast-charge termination are usually initiated by the software. The MAX846A includes a linear regulator and a CPU-reset output for the μC .

The least autonomous and most flexible devices are the MAX1647 and MAX1648. They are similar, except the MAX1647 has built-in DACs and an SMBus serial port, and the MAX1648 has analog inputs for voltage and current control. The MAX1647 is a complete, serially controlled dc power supply with independent voltage and current registers. Capable of SMBus communications with a smart battery, it provides Level 2 compliance with the Intel/Duracell smart-battery specification.

μC Design Tips

These charger ICs typically operate with a low-cost 8-bit controller such as the 8051, PIC, 68HC11, or 68HC05. The firmware can be written in assembly language or in C, either of which feature ready availability, low cost, and free tools. Third parties and manufacturers of these devices have assembled an impressive array of compilers, assemblers, emulators, and code libraries. Much of this source code is available on the World Wide Web, especially the toolbox routines for assembly language. The Tips for charger program structure section provides further information on these resources.

All common 8-bit μC s are suitable, but the selection of a specific μC is beyond the scope of this article. Peripherals such as analog-to-digital converters (ADCs), DACs, and the SMBus serial interface are available in these μC s, and simpler μC versions that require external ADCs or DACs are also useful. Often, simpler μC versions that require external ADCs or DACs are more flexible and ultimately more useful.

The ROM and RAM requirements for charger applications are modest. In general, you can implement a single-chemistry charger in less than 0.5kbytes of code and 32 bytes of RAM (simple requirements for even a low-end PIC). With some ingenuity, you can implement a multi-chemistry charger with about 50% more code.

The simplest way to develop μC code is to start with a skeleton or a piece of similar code, and modify it to suit your needs. This approach gets a prototype working quickly by overcoming a lot of the blank-page, compiler/assembler- syntax problems. Unfortunately, only a limited amount of battery-charger firmware exists on the Web and in standard application notes. However, two design examples in the Hardware and Software Examples section provide a starting point. See the Resources and references section for more information on some of the more difficult toolbox routines, such as SMBus communications and math routines, and for examples of program designs that illustrate approaches to these designs.

Tips for Charger-Program Structure

Writing battery-charger software is straightforward and best implemented with a state machine. Define a state variable or series of flags that represents the current state. The code then tends to be a large case statement that acts according to this state variable. The code modules modify the state variable according to the current conditions. Disallowed and undecodable states pose the only potential problems. All case statements must have a default case that picks up these disallowed or "impossible" states and corrects them. Always include a mechanism that detects these conditions and then takes intelligent action, such as stopping the charger.

Keep the code simple: avoid multiple interrupts and complex multitasking or queuing structures where possible. Using a single timer-tick interrupt is a very effective way to keep time. If the CPU has a timer with an interrupt, use it to maintain system-timer flags. This powerful technique is an exception to the no-interrupts rule. If no timer interrupt is available (as in the PIC16C5x), use the system timer (RTC) and poll it. Design the code so the timer cannot overflow between polls.

Avoid hardware interrupts. Instead, poll the hardware inputs at regular intervals set by the timer tick. Code execution takes place in real time, but it doesn't have to react immediately to stimuli. The 100ms required to determine whether the battery is installed is acceptable, considering that battery charging takes an hour. Typical performance for stand-alone chargers is usually one calculation per minute for termination.

A simple and workable structure for these programs is a paced loop. The main program is a loop that looks at timer flags set by a timer interrupt-service routine or the loop itself, and calls subroutines that perform the multiple tasks required. Some routines run on each pass, and others run on every "nth" loop or tick. The basic tick time might be 100ms, for example. A blinking-light subroutine with a half-second period would be called to complement the LED every five ticks, and the temperature-limit detector would be checked on each pass through the loop. The result is a very robust structure.

For controllers that lack a timer interrupt, the paced loop can be implemented by the routines themselves, using their own execution times to maintain system timing. This technique is implemented in the next section by the code example for an 8-pin PIC controller. A simple flow chart of this structure (Figure 7) is described in greater detail in Reference 7.

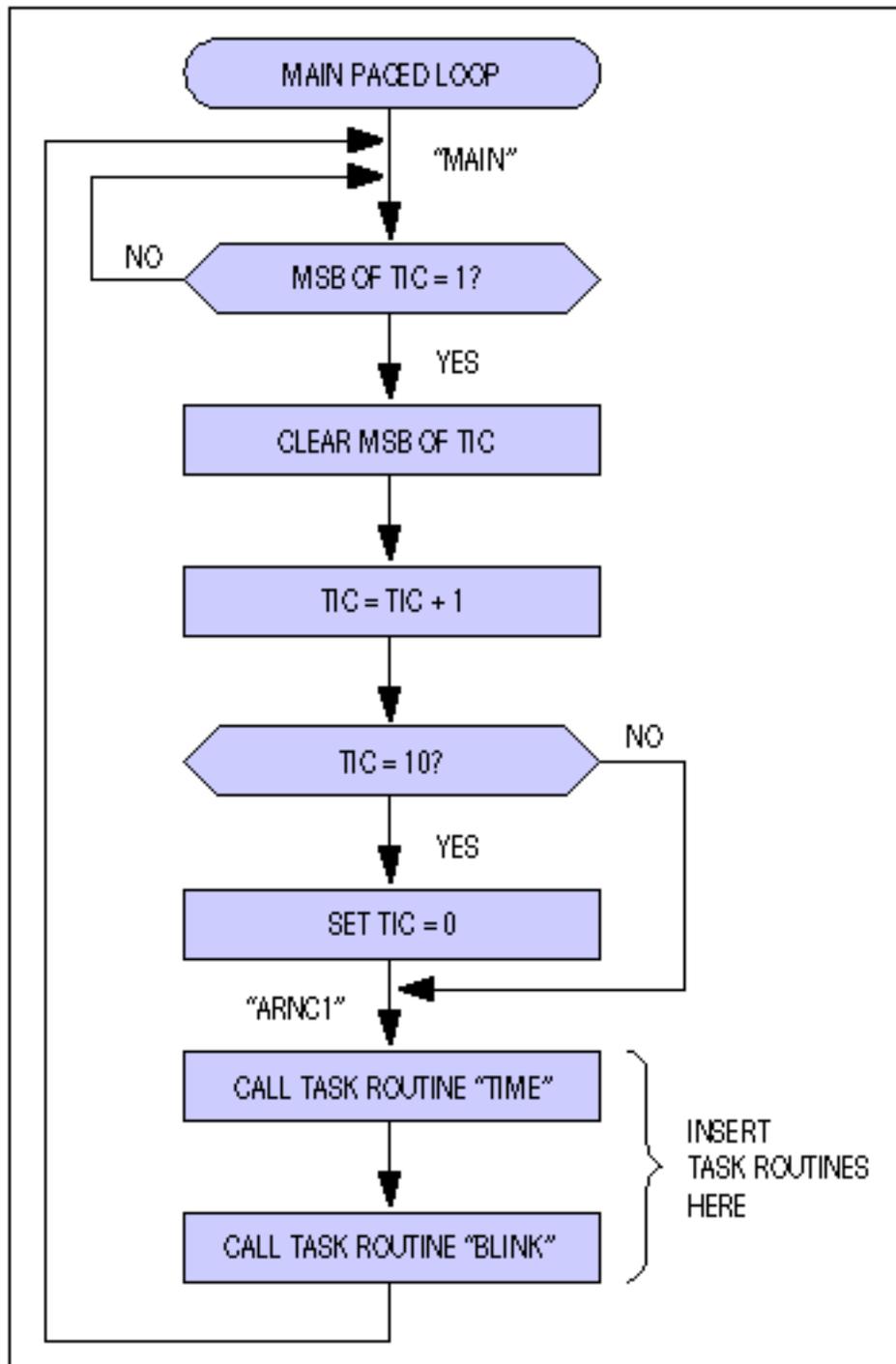


Figure 7. Main paced-loop flow chart

Hardware Fail-Safe Reminder

Before exploring some examples, one final recommendation is to consider the use of a μP supervisor with a watchdog timer and a hardware fail-safe system. The supervisor's reset function provides a clean system reset when the power comes up, and the watchdog timer can catch a stalled CPU or errant firmware stuck in a loop. Maxim also makes some simple temperature-measurement/control products. The MAX6301 temperature switches make an especially good backup system. They are SOT23 devices that change their output level when a

fixed temperature threshold is crossed.

Supervisors are especially important in charger applications, because the constant application and removal of power to the charger can confuse the CPU. If, for example, the processor stalls and fails to terminate a fast charge, the results can be catastrophic. The system should also include a temperature sensor or other hardware override that can end the fast charge without software intervention. Some of Maxim's SOT23-reset supervisors include a watchdog (see the MAX823).

Hardware and software examples

1) MAX846A Li+ charger with charge timer and LED-status outputs, controlled by an 8-pin PIC

In this example, a small external μP enhances the MAX846A, forming a complete desktop-charger system that includes user-interface functions such as the LEDs in Figure 8 (to indicate the charge process and status). The MAX846A is designed for this type of operation. Its auxiliary linear regulator and μP -reset circuit (to support the external μC) reduces the cost of a typical desktop-charger application.

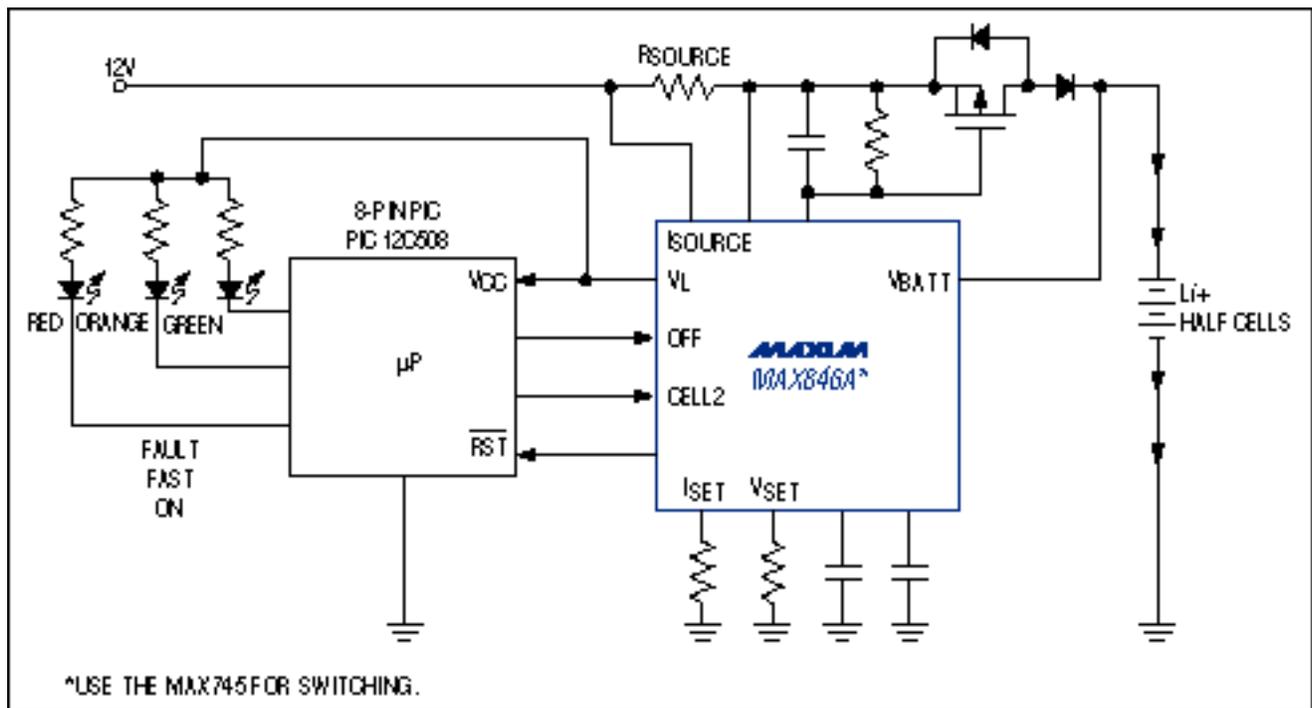


Figure 8. Li+ desk charger with LED status indicator

2) MAX1647-based, 2A Li+ charger with 8051 μC

The full-featured MAX1647 charger and 8051 μC form a full-featured Li+ charger (Figure 9). The Atmel 80C2051 controller shown (a nonexpandable 8051 in a small package) is typical of the

Figure 9. Full-featured Li+ charger

3) Software examples for the MAX1647 and MAX846A chargers

Software for the MAX1647 and MAX846A examples (Figure 9) is available at Maxim's web site. MAX846A software for the 8-pin PIC12C508 controller is written in Microchip PIC assembly language. It implements an LED user interface and a timer that terminates the fast charge five minutes after reaching the Li+ voltage limit. This simple example does not include the state machine or the complexities of a full charger, because much of that capability is available in the nearly stand-alone MAX846A.

The example does rely on the paced-loop structure without interrupts, as described earlier.

The MAX1647 example is written in 8051 assembly code for Atmel's ATM80C2051, a 20-pin version of the 8051. This code includes a general state-machine structure and SMBus-driver routines for communicating with the MAX1647 internal registers. It also incorporates a paced-loop structure, but employs the 80C2051's timer interrupt to create a timer-tick basis for all timing. For further details, see the source-code documents at Maxim's web site.

Resources and References

The following is a brief sampling of application notes and other resources available on the World Wide Web and from vendors. Most vendors publish their application notes on the web for easy access. Simply accessing the web and entering a μ C part number into the AltaVista search engine usually yields more than 50 documents.

8051-derivative Application Notes

Philips Semiconductors: Web site and CD-ROM

AN422: Using the 8XC751 Microcontroller as an I²C Bus Master

AN428: Using the ADC and PWM of the 83C752/87C752

AN439: 87C751 fast NiCd charger

EIE/AN92001: Low RF-Emission Applications with a P83CE654 Microcontroller

Intel Corp.: Web site and CD-ROM

Atmel Corp.: Web site and CD-ROM

A Digital Thermometer Using the AT89C2051 Microcontroller

Interfacing 24CXXX Serial EEPROMs with AT89CX051 MCU

68HC05 Application Notes

AN1263: Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers

AN1262: Simple Real-Time Kernels for HC05 MCUs

AN1256: Interfacing the HC05 MCU to a Multichannel D/A Converter

AN1241: Interfacing the HC05 MCU to Serial EEPROMs

AN1227: Using Serial EEPROMs with HC05 MCUs

AN477: Simple A/D Conversion for MCUs Without Built-In ADCs

PIC Application Notes

Microchip: Web site and CD-ROM

AN541: Using a PIC16C5X as a Smart I2C Peripheral

AN546: Using the A/D Converter in the PIC 16C73

AN554: Software Implementation of I2C Bus Master

AN577: PIC16C54A EMI Results

AN552: Implementing Wake-Up on Keystroke for the 16C54

AN585: A Real-Time Operating System for PIC16/17

AN606: Low-Power Design Using PIC16/17

AN520: A Comparison of Low-End 8-bit Microcontrollers

Parallax: Third-party web site and tools

References

1. *How to Implement an SMBus controller using the 80C51SL KBC*, Intel Corp. application note, November 1994.
2. *Handbook of Batteries*, by David Linden (Editor), 2nd Edition, McGraw Hill text, January 1995, ISBN: 0070379211
3. *The System Management Bus specification*, Versions 0.95a and 1.0, Intel Corp., February 1995.
4. *The Smart-Battery Data specification*, Version 1.0, Duracell Inc. and Intel Corp., February 1995.
5. *The SMBus BIOS specification*, Version 1.0, Intel Corp., February 1995.
6. *Smart-Battery Selector specification*, Version 0.9, Intel Corp., April 1995.
7. *Understanding Small Microcontrollers*, by James Sibigtroth. Published by Motorola Inc., CSIC Division, circa 1990.

SMBus is a trademark of Intel Corp.

I²C is a trademark of Philips Semiconductor Corp.

More Information

MAX1640: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX1926: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX712: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX713: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX745: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAX846A: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)