AN1467

High-Power CC/CV Battery Charger Using an Inverse SEPIC (Zeta) Topology

Authors: Mihnea Rosu-Hamzescu
Sergiu Oprea
Microchip Technology Inc.

INTRODUCTION

As the number of photovoltaic systems and electric vehicles increases, so does the demand for intelligent, high-power and high-efficiency battery chargers. Most systems on the market today use either lead-acid or lithium type batteries, requiring constant current/constant voltage charging algorithms.

This application note contains the necessary information to build a 100W inverse SEPIC (also called Zeta converter) battery charger. The novelty consists in driving this topology synchronously using Microchip components, essentially pushing the efficiency over 95% at 8A. The Zeta converter has many advantages, such as input to output DC insulation, buck-boost capability and continuous output current, but it is difficult to control.

The control scheme is also interesting, as it uses the Numerically Controlled Oscillator (NCO) peripheral to implement a form of fixed on-time, variable frequency control that allows 15 bits of resolution for the control system. This opens up quite a few new possibilities in low-cost software controlled power supplies.

Finally, since this implementation allows the control of the output voltage and current with a high resolution, it is quite easy to attach multi-chemistry battery charging algorithms to the basic output regulation loop, greatly increasing its usefulness.

The complete implementation of the regulator and charger library uses only 1k words of program space and 55 bytes of RAM.

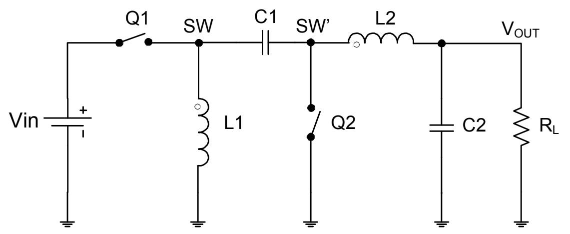
THE ZETA CONVERTER

Considered by many designers as an “exotic” topology, the ZETA converter (also known as the inverted SEPIC) offers certain advantages over the classical SEPIC. This topology has the same buck-boost functionality as the SEPIC, but the output current is continuous, providing a clean, low-ripple output voltage make. This low-noise output converter can be used to power certain types of loads, such as LEDs, which are sensitive to the voltage ripple. The ZETA converter offers the same DC isolation between the input and output as the SEPIC converter, and can be used in high-reliability systems.

This topology can also offer high efficiency, especially if the synchronous rectification is used. The synchronous rectification can be easily implemented here, because this topology, unlike the SEPIC converter, uses a low-side rectifier.

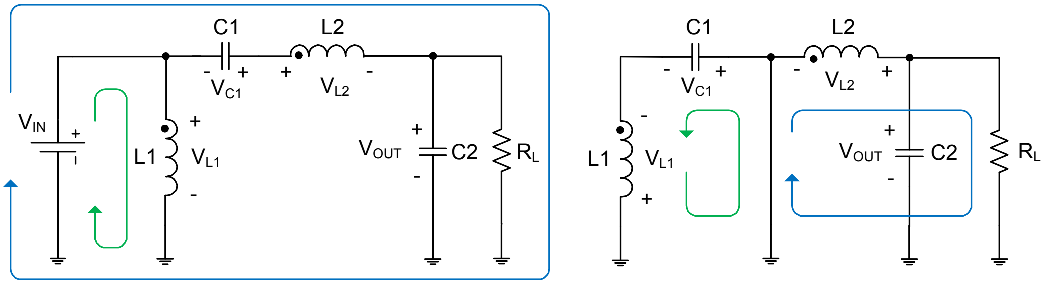
The ZETA converter power train is depicted in Figure 1.

FIGURE 1: THE ZETA CONVERTER POWER TRAIN



The two switches, Q1 and Q2, operate out of phase. As with the SEPIC converter, there are two switching cycles that are presented in Figure 2.

FIGURE 2: SWITCHING CYCLES OF THE ZETA CONVERTER



Cycle 1: Q1 closed, Q2 open

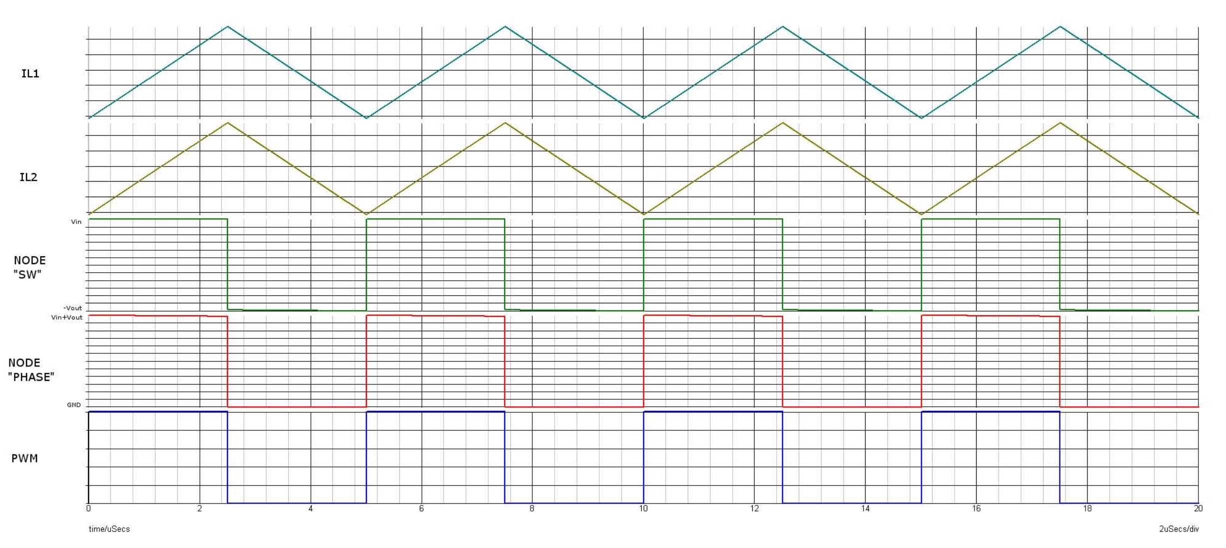
Cycle 2: Q1 open, Q2 closed

In the first cycle, Q1 is closed and the current begins to flow in the primary inductor L1 and through the load via the coupling capacitor C1 and inductor L2. In the second cycle, Q2 is closed and the energy stored in the L2 inductor is delivered to the load. The energy stored in the main inductor L1 will be reset to its initial value through the coupling capacitor C1.

The typical waveforms of the ZETA converter are presented in Figure 3. The continuous current flow on the load is maintained by the output inductor L2. The voltage across the main switch (Q1) is the sum of the input and output voltages as is the case with the SEPIC converter. The voltage stress across the main switch is higher and can increase the switching losses of Q1.

The two inductors can be magnetically coupled, sharing the same magnetic core. This can greatly reduce the current ripple, as the mutual inductance will double the apparent value of the inductors.

FIGURE 3: TYPICAL WAVEFORMS FOR THE ZETA CONVERTER



If the converter operates in Continuous Current Mode (CCM) and reaches the steady state, the volt-second balance principle can be applied to determine the DC transfer function (transformation ratio).

EQUATION 1: STEADY STATE ANALYZE

"On State"

$$V_{L1} = V_{IN}$$

$$V_{L2} = V_{IN} + V_{C1} - V_{OUT}$$

"Off State"

$$V_{L1} = -V_{C1}$$

$$V_{L2} = -V_{OUT}$$

EQUATION 2: VOLT-SECOND BALANCE

$$D * \mathrm{V}_{IN} - (1 - D) * \mathrm{V}_{C1} = 0$$

$$D * (\mathrm{V}_{IN} + V_{C1} - VOUT) - (1 - D) * \mathrm{V}_{OUT} = 0$$

The DC transfer function can be found by solving this system of equations.

EQUATION 3: DC TRANSFER FUNCTION

$$\mathrm{V}_{OUT} = \frac{D}{1 - D} * \mathrm{V}_{IN}$$

For a duty cycle (D) lower than 50%, the ZETA performs as a buck converter, and for duty cycle higher than 50%, as the boost converter. As this converter requires high and low side switches, it can be implemented using drivers developed for the synchronous buck converter, like the MCP14628. However, some technical challenges must be solved before using the MCP14628 synchronous buck driver. As can be seen from the waveforms, the main switching node (SW) goes below ground. The typical application for the MCP14628 synchronous buck converter must be modified in order to avoid damage of the chip when the SW node goes below ground.

Figure 4 shows the typical application for the MCP14628 synchronous buck driver, while Figure 5 shows the modified schematic to drive the synchronous ZETA converter.

FIGURE 4: TYPICAL APPLICATION FOR THE MCP14628 SYNCHRONOUS BUCK DRIVER

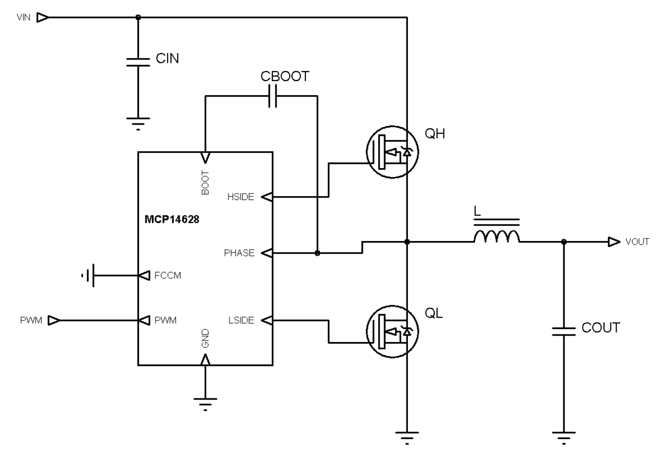
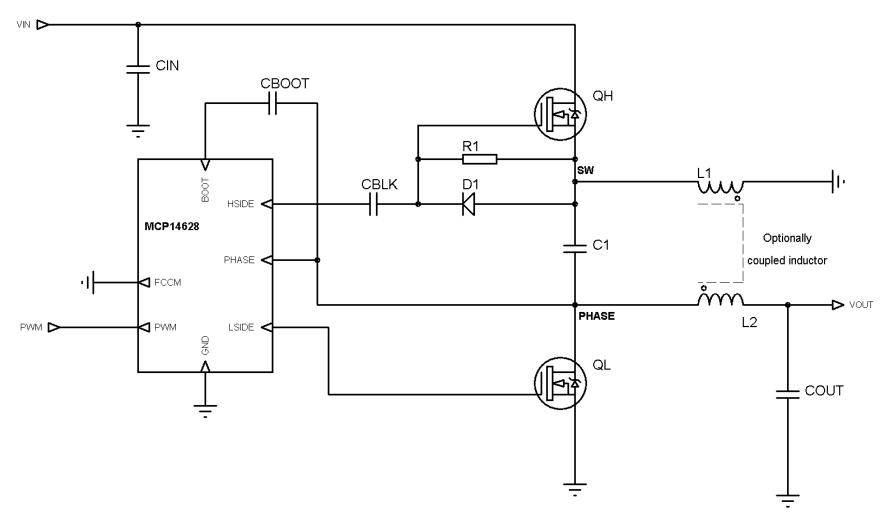


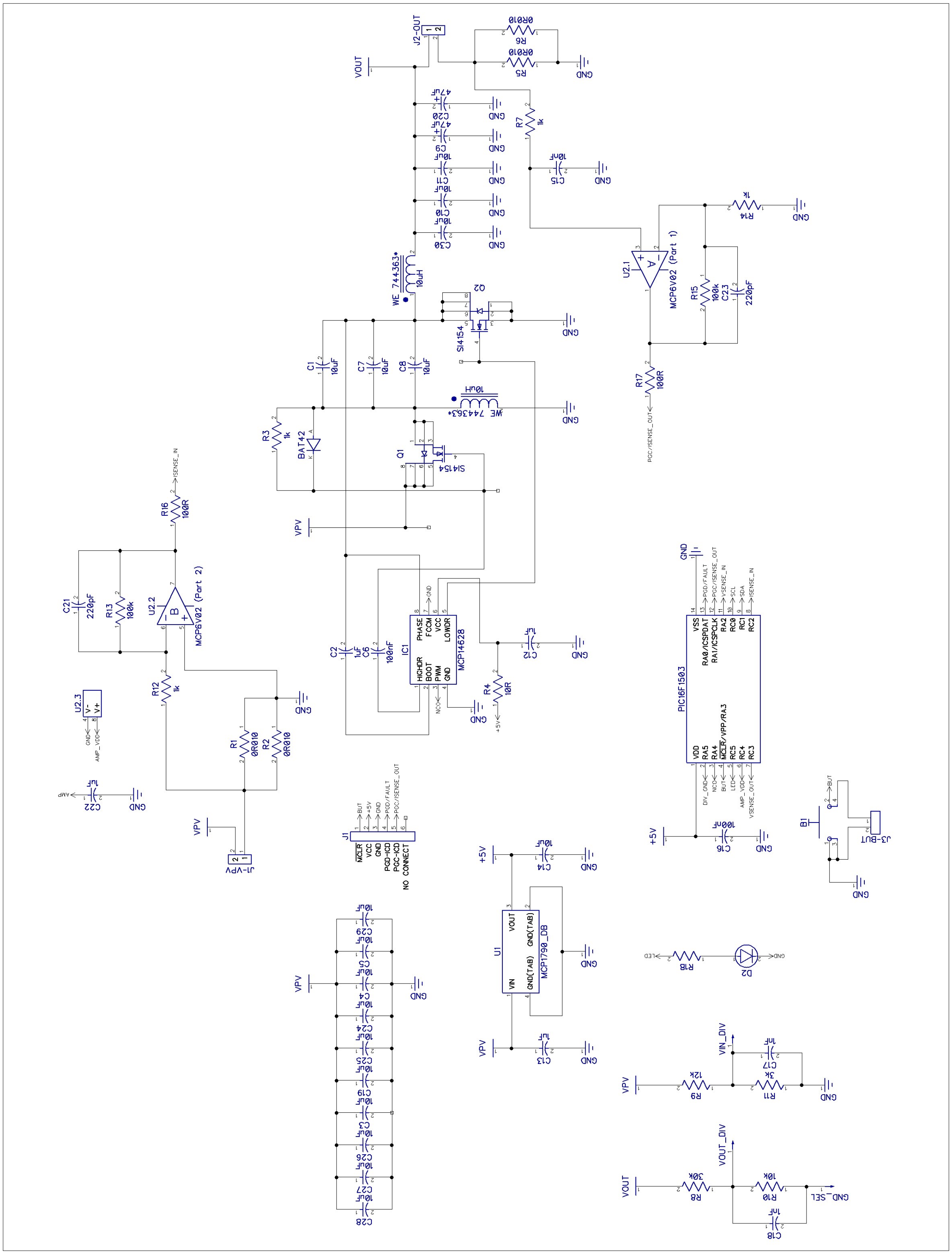
FIGURE 5: MCP14628 DRIVING THE SYNCHRONOUS ZETA CONVERTER



The PHASE pin remains connected to the drain of the synchronous rectifier MOSFET (QL), allowing the bootstrap capacitor (C_{BOOT}) to be charged during the synchronous rectification state. The gate of the main transistor MOSFET (QH) is now driven through the isolation capacitor CBLK. The gate and source of the main MOSFET can now safely swing below the ground level, without affecting the functionality of the MOSFET driver. Diode D1 and resistor R1 form a DC component restorer circuit, that will center again the drive signal to 2.5V. The loop for the driving current of the high-side MOSFET includes now the ZETA coupling capacitor, C1.

HARDWARE IMPLEMENTATION

FIGURE 6: ZETA CONVERTER BOARD SCHEMATIC



THE SOFTWARE CONTROL LOOP

EXAMPLE 1: THE VOLTAGE/CURRENT CONTROL LOOP

```
void main()
{
    Initialize_Hardware();

    while(1)
    {
        if(T0IF)
        {
            T0IF = 0;
            if(second) second--;

            read_ADC();

            cc_cv_mode();

            if(!cmode) pid(vout, vref); else
            pid(iout, iref);
        }

        if(!second)
        {
            second = SECOND_COUNT;
            Battery_State_Machine();
        }
    }
}
```

PIC16F1503 is one of the enhanced core devices, which benefit from the recent line of peripherals, such as the NCO (Numerically Controlled Oscillator), CWG (Complementary Wave Generator) or CLC (Configurable Logic Cell). To implement a software-controlled regulation loop, the user needs:

- 2 ADC inputs (10-bit or better recommended) for monitoring output voltage and current.
- 1 NCO output for the converter control signal
- 1 Timer for the main timer tick
- (optional) LED signaling, serial interface for logging and button

The main control loop is based on TIMER0, which is configured to overflow 4MHz/256/16 = 976.56 times per second, using the 1:16 prescaler. This is the timer tick used for the PI loop.

Every timer period, output voltage (V_{OUT}) and current (I_{OUT}) is read (4 samples each) and stored. Since there is only one regulation loop, the firmware must decide if it needs to regulate the current or voltage at a certain time. The decision is simple enough: if the read voltage is over the set voltage reference, the converter will limit the voltage; if the read current is over the set current reference, then the converter will limit the current.

The function `cc_cv_mode()` takes care of the transitions between the two working modes, and the variable `cmode` shows whether the current or voltage is currently regulated. This variable is also important for the topping stage of the battery charging algorithms. The function also uses a fast debouncing counter, that prevents erratic jumping between the constant voltage and Constant Current mode in boundary conditions.

Regulating the voltage or current is very simple. Depending on the value of `cmode`, the PI function is called with different parameters:

- `pid(vout, vref)` for voltage or
- `pid(iout, iref)` for current

The variables `vref` and `iref` are the output voltage and current limits. The special macros `SET_VOLTAGE(x)` and `SET_CURRENT(x)`, which modify these variables, allow the state machine to change the charging parameters transparently.

The PI function operates on the NCO increment, practically varying the duty cycle by varying the frequency. The NCO operates in Pulse Frequency mode configured for a 2 µsec pulse. This means the converter operates using a fixed on-time, the duty cycle raises proportionally with the frequency, and the on-time value allows 15 bits of frequency resolution. At 500 kHz, we have 100% duty cycle, so the maximum increment value is clamped to 29500 (out of 32768), which gives 450 kHz or 90% duty.

THE CC/CV CHARGER LIBRARY

Except for Ni-MH (and Ni-Cd), all the popular battery chemistries on the market today use a form of constant current, followed by a constant voltage charging (or constant voltage with current limit) algorithm. Since the hardware presented in this application note is capable of regulating output voltage and current, using it for this purpose comes only naturally. Also, one of the biggest problems in synchronous chargers, the battery reversal current, is solved by the driver diode emulation feature.

The converter's maximum current is 8A, so it is somewhat impractical to use it for very small batteries (current shunt amplifier output for 50 mA or less is close to the noise floor of the ADC). Also, because this implementation has no burst mode at very low duty cycles, the output ripple increases. It is best used with lead-acid and lithium type batteries with capacities over 4 Ah (also probably not useful on batteries bigger than 80 Ah).

MULTI-STEP CHARGING

Properly charging the batteries requires multiple steps and specific mechanisms for each chemistry.

Lead-acid batteries have generally a more sluggish behavior when charging, and the process will take longer. It is not recommended to charge using currents over C/5, or the topping charge portion may not be complete when the current threshold triggers the end of the charge. Also, lead-acid batteries may be placed in floating charge for extended periods of time.

Li-Ion chemistries have different charging requirements, in the sense that they will not absorb the overcharge, so the current flow must be cut as soon as the battery is full. The optimum charging current is C/2, for a good balance between the charge time and the battery life cycle. Deeply depleted cells have to be trickle-charged until the cut-off threshold is reached. Some Li-Ion chemistries are more resilient and forgiving to abuse than others. LiFePO4 is one such example.

Ni-Zn cells are similar to Li-Ion. No float charge should be applied, and deeply depleted cells need to be trickled back to life.

All these particularities need to be implemented in the charger state machine and a set of charging parameters written for each chemistry type.

TABLE 1: BATTERY CHEMISTRIES

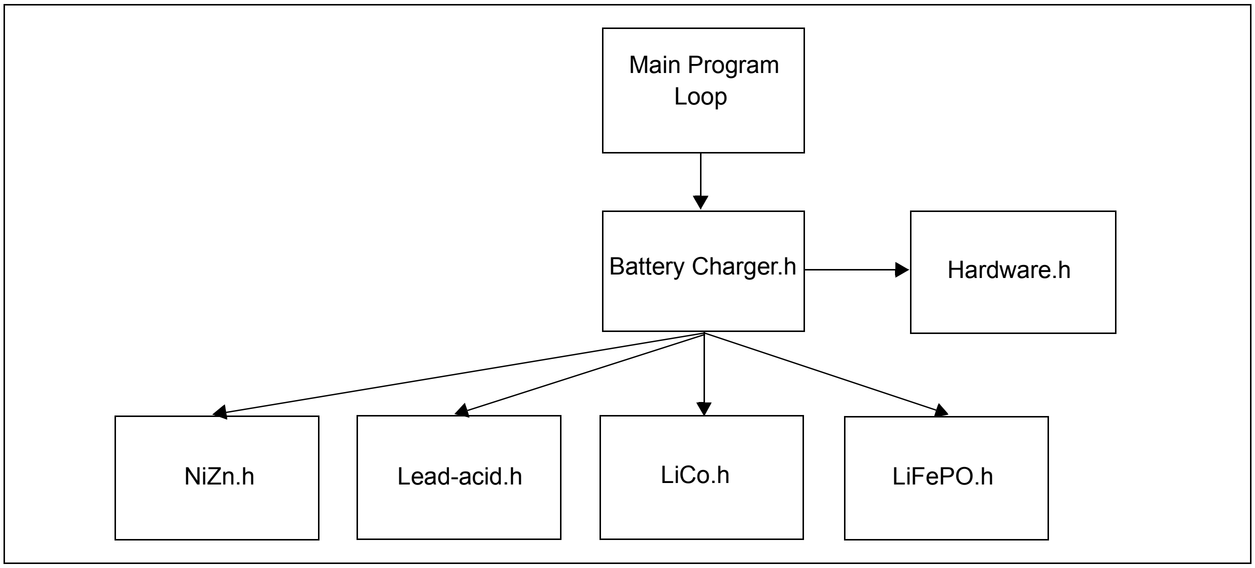
Chemistry	Pre-charge	Charge	Float
Lead-acid	No, if under 1.75V per cell for extended periods replace it.	2.4V/cell to C/40 current	Yes, 2.25V/cell
Li-Co	Yes. C/10 to 3.0V	4.2V/cell to C/33 (3%)	No
LiFePO4	Yes. C/10 to 2.7V (2.5V for some variants)	3.65V/cell to C/33 (3%)	No
Ni-Zn	Yes. C/10 to 1.3V	1.9V/cell to C/33 (3%)	No

FIRMWARE STRUCTURE

Because all the voltage and current regulating functions are in the main program loop, the battery charging state machine is only called every second and it only needs to make decisions based on the output voltage and current values. Depending on the current charging state and chemistry type, it will set the voltage and current limit.

Big (to huge) batteries or cells take a long time to relax from the charging voltage to the open circuit voltage, so it may be unfeasible to stop charging to read the OCV (open circuit voltage) from time to time. For example, even a small 3-cell 4Ah lead-acid battery takes 2-3 seconds to relax from 7.2V to 6.75V, when switching to float charge.

FIGURE 7: CC/CV CHARGER LIBRARY FILE STRUCTURE



THE MAIN LOOP AND *HARDWARE.H*

All the output regulating functions should be placed here. Because the charger state machine itself is largely hardware independent, it does not matter how the output regulation is done (in this case, a simple PI software loop).

There are still some requirements that need to be met for the battery state machine to function properly.

Besides the obvious requirement to call the state machine every second, different parameters and macros need to be available to the state machine code. For this implementation, all values are 12 bits (4 x 10-bit ADC readings). Using a different type of ADC or more samples requires changing all the related parameters and each battery chemistry header file.

- `VSENSE` and `ISENSE` will contain updated values of the output voltage and current. They can be defined as macros or return functions.

```
#define VSENSE vout
#define ISENSE iout
```

- `SET_VOLTAGE(x)` and `SET_CURRENT(x)` will set the converter maximum output voltage and current. They can be defined as macros or functions.

```
#define SET_VOLTAGE(x) { vref = x; }
#define SET_CURRENT(x) { iref = x; }
```

- `SET_LED_BLINK(x)` sets the LED state and blink rate to show the current battery charging state.

```
#define SET_LED_BLINK(x) { led_state = x; }
```

`CONSTANT_VOLTAGE` should show whether the converter is regulating the output voltage or not. This is important for the state machine, because the minimum current and flat current charge termination should only be initiated in Constant Voltage mode.

```
#define CONSTANT_VOLTAGE (!cmode)
```

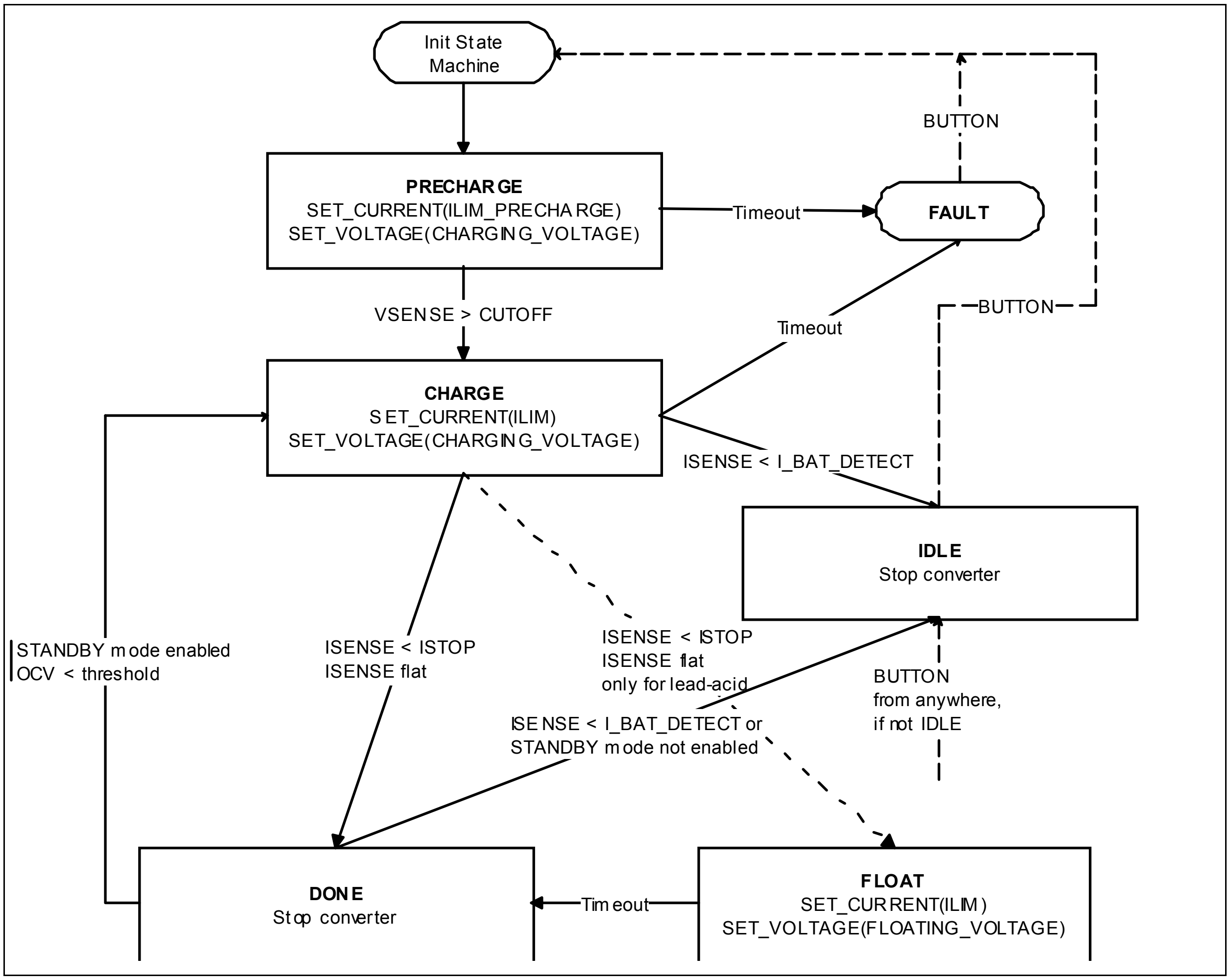
- `I_BAT_DETECT` is a minimum current reading that shows a missing battery/load. The value depends on the current shunt amplifier offset and amplification. It is useful for detecting that the battery has been removed during charge, since there is no OCV measurement.

```
#define I_BAT_DETECT 16
```

BATTERYCHARGER.H

The Charger State Machine

FIGURE 8: CHARGER STATE MACHINE



Please note that the state machine has been designed with basic functionality and simplicity in mind, and it may be further improved depending on the final application.

There are 6 states**:**

- **PRECHARGE** – will charge using a low current setting (usually C/10) until the voltage exceeds the defined cut-off voltage. This is necessary for Li-Ion chemistries and for Ni-Zn. Lead-acid type batteries should not need this, and the trickle current may be the same as the full charge current. In this case, the battery will switch to CHARGE after one state machine update (1 second).
- **CHARGE** – will charge using the defined full current setting. Once the converter goes into Constant Voltage mode, the state machine starts monitoring output current. When the minimum output current threshold has been reached or the current does not decrease for a certain time (flat current), the state machine switches to either FLOAT (for lead-acid) or DONE for the rest of the chemistries. If the stop current was below the battery detection threshold, it will go directly to IDLE (battery removed). This is useful to quickly stop charging if there is no battery. If a time out occurs before any of these conditions are triggered, then the state machine will switch to FAULT.

- **FLOAT** – works in Constant Voltage mode (typically 2.25V/cell). It is necessary for lead-acid batteries to maintain full charge (counter battery self-discharge). This state has a time out, which switches to DONE when it expires. After a defined battery relaxation period, it checks the battery current and, if it is below the battery detection threshold, it switches to IDLE. The “relaxation” period is necessary after switching from the charging voltage to the lower, floating voltage. Until the battery voltage relaxes, there will be no current flowing to the battery and the state machine might incorrectly detect that the battery has been removed. The bigger the battery (capacity), the slower the relaxation.
- **DONE** – is the final state for fully charged batteries. The converter is stopped, but, if a special `STANDBY_MODE` is enabled, then this state will monitor the OC voltage of the battery and jump back to CHARGE, when it drops too much. This way all types of batteries can be maintained near full charge for extended periods of time. This even works for Li-Ion, if the voltage threshold is chosen carefully. If `STANDBY_MODE` is not enabled, this state will switch automatically to IDLE.
- **IDLE** – stops the converter and does not do anything, except wait for user input. A button press will re-initialize the state machine and start the charging process. To automate the process, it is simple enough to add output voltage monitoring that will start charging when a battery is connected.
- **FAULT** – stops the converter and waits for the user input. A button press will change the state to IDLE.

One LED is used to signal the current charger state to the user. The LED on/off states and blinking rates are defined in *Hardware.h* (blinking rate depends on the timer tick period). Each charger state sets the LED behavior:

- LED off – IDLE state
- LED blinking 0.5Hz – PRECHARGE and CHARGE states
- LED on – FLOAT and DONE states
- LED blinking 2 Hz – FAULT state

Charger Variables and Functions

The charger needs a few variables to keep track of the battery parameters. Most of the variables are internal, but some of them are also available to the main program loop.

- `battery_state` – holds the current charging state of the battery charger. Since the charger accepts only one battery, there is no reason to differentiate between the state of the charger and the state of the battery. This variable is available to the main program loop.

  ```
  enum charge_states { IDLE = 0, FAULT
  = 1, DONE = 2, PRECHARGE = 3, CHARGE
  = 4, FLOAT = 5 };
  ```

- `state_counter` – is used as a time-out counter for certain states:
  - PRECHARGE will switch to FAULT on time out
  - CHARGE will switch to FAULT on time out
  - FLOAT will switch to DONE on time out
- `imin` – is the minimum current value recorded during the constant voltage phase of the CHARGE state. When the value of this variable falls below the minimum current threshold (calculated as a fraction of the battery capacity), the topping charge is done.
- `imin_db` – is the minimum current debouncing counter. When the value of `ISENSE` is smaller than `imin` for `IMIN_UPDATE` times in a row, `imin` is updated with the value of `ISENSE`
- `iflat_db` – is the flat current debouncing counter. This counter is reset every time `imin` is updated. When it reaches zero, it triggers an end of charge condition.

The charger library has two functions:

1. `Init_State_Machine()` will initialize the state machine debouncing and time-out counters, set the charging voltage and current limits and start the converter. This function should be called when starting to charge from IDLE (basically, a new battery is inserted).
2. `Battery_State_Machine()` contains the code for each of the charger/battery state machine and will handle state transitions based on the measured current and voltage values. This function expects to be called every second, otherwise the time-out counters will measure a different interval.

THE BATTERY CHEMISTRY DEFINITION FILES

Lead-acid.h, *LiCo.h*, *LiFePO.h* and *NiZn.h* contain example definitions for charging these chemistries.

As mentioned before, the values in the definition files depend on the charger hardware implementation and the number of ADC samples taken on each measurement. In this case, we have a 10-bit ADC with a 5V reference, and 4 samples are taken for every measurement. The output current shunt is 5 mOhms, amplified 101 times. The output voltage divider is ¼.

EQUATION 4: VOLTAGE CALCULATION EXAMPLE FOR 7.2V

$$Voltage_{COUNTS} = \frac{Voltage}{Output_{divider} * \mathrm{ADC}_{reference}} * ADC_{max}$$

*Counts = 7.2V / 4 / 5V * 4096 = 1474.56*

EQUATION 5: CURRENT CALCULATION EXAMPLE FOR 8000 MA

$$Current_{COUNTS} = \frac{Current * Shunt_{value} * Amplification}{\mathrm{ADC}_{reference}} * ADC_{max}$$

*Counts = 6A * 0.005Ω * 101 / 5V * 4096 = 2482.17*

The lead-acid and LiFePO4 headers are shown as an example for charging parameter calculation. The first is an automotive type 60 Ah battery, and the second is a 20 Ah LiFePO4 cell.

For smaller batteries, the current shunt and amplification should be sized accordingly. A small 1800 mAh 18650-type cells has an end-of-charge current of just 55 mA. Because of the battery current shunt value, which is sized for the 8A output limit, it is hard to work with low currents. The 55 mA threshold translates to 23 ADC counts (oversampled 4x), and is dangerously close to the noise floor.

EXAMPLE 2: LEAD-ACID BATTERY SPECIFIC PARAMETERS

```
//Lead-acid battery specific parameters

#define PRESET1

//      1.75V per cell cutoff voltage
//      2.10V per cell charged OCV voltage
//      2.25V per cell floating voltage
//      2.40V per cell charging voltage

#ifdef PRESET1
    #define LA_3CELL
    #define CAPACITY              60000   //mAh
    #define CHARGING_VOLTAGE      1474    //7.20V
    #define FLOATING_VOLTAGE      1382    //6.75V
    #define TOPPING_VOLTAGE       1290    //6.30V
    #define CUTOFF_VOLTAGE        1075    //5.25V
    #define ILIM_PRECHARGE        2482    //only important for Li-Ion
    #define ILIM                  2482    //6A, C/10
    #define IFLOAT                496     //1.2A, C/50 (2%) minimum charging current
#endif

#define BATTERY_STANDBY_MODE

#define PRECHARGE_TIME        600
#define CHARGE_TIME           57600   //16 hours

#define FLOAT_TIME            43200
#define FLOAT_RELAX_TIME      (FLOAT_TIME - 60)

#define IFLAT_COUNT           600
```

Lead-acid batteries have a few parameters that are different:

- `LA_3CELL` or `LA6_CELL` is just a define that shows how many cells the battery has
- `FLOATING_VOLTAGE` is used for setting the float charge voltage
- `IFLOAT` is used instead of `ISTOP` to set the current threshold that will trigger the end of charge (and the switching to float charge). The value is in ADC counts (calculated using Equation 5)
- `FLOAT_TIME` is the floating charge state period in seconds.

- `FLOAT_RELAX_TIME` is a blanking time (in seconds) at the beginning of the floating charge state needed for the battery to relax from the previous, higher, voltage used in the charge state. Without this, the charger detects no current flowing through the battery and switches to IDLE immediately.

EXAMPLE 3: LiFePO$_4$ BATTERY PARAMETERS

```
// LiFePO4 battery parameters

#define PRESET1

//      2.7V per cell cutoff voltage (2.5V under heavy load)
//      3.4V+ per cell charged OCV voltage
//      3.65V per cell charging voltage

#ifdef PRESET1
    #define CAPACITY                20000
    #define CHARGING_VOLTAGE        748     //3.65V
    #define TOPPING_VOLTAGE         696     //3.40V
    #define CUTOFF_VOLTAGE          553     //2.70V

    #define ILIM_PRECHARGE          827     //2A, C/10
    #define ILIM                    3310    //8A, C/2.5, hardware limit
    #define ISTOP                   251     //606mA, C/33 (3%) minimum charging current

#endif

#define BATTERY_STANDBY_MODE

#define PRECHARGE_TIME  600
#define CHARGE_TIME     14400

#define IFLAT_COUNT     600
```

Lithium type cells and Ni-Zn will not accept overcharge, so the floating charge-related parameters are removed. Also, `IFLOAT` is replaced by `ISTOP` as the end of charge minimum current threshold.

The rest of the parameters are common to all types of cells/batteries.

- `CAPACITY` is the battery capacity in mAh. It is useful to calculate the charge current limit and end-of-charge current limit
- `CHARGING_VOLTAGE` is the charging voltage limit in ADC counts, calculated with the formula in Equation 4.
- `TOPPING_VOLTAGE` is the open circuit voltage below which the charger will start charging again if `BATTERY_STANDBY_MODE` is defined. The value is the number of ADC counts (the same as the charging voltage)
- `CUTOFF_VOLTAGE` is the low voltage limit of the battery. If battery is below this value when connected to charge, it will be trickle-charged with a low current setting.
- `ILIM` is the charging current limit in ADC counts, calculated with the formula in Equation 5.
- `ILIM_PRECHARGE` is the trickle charge current for `PRECHARGE` (calculated the same way as the charging current)
- `IFLOAT/ISTOP` is the minimum current limit for the charging phase. It is also calculated using Equation 5.
- `BATTERY_STANDBY_MODE` is a parameter that alters the behavior of the charger state machine after the charging process is finished. Normally, at the end of the floating charge phase (for lead-acid) and or charge phase (for lithium and Ni-Zn), the charger goes to IDLE. If Standby mode is defined, then the charger stays in the DONE state and monitors the OC voltage of the battery. If the voltage drops below `TOPOFF_VOLTAGE`, it starts charging again from the CHARGE state.
- `PRECHARGE_TIME` is the maximum pre-charge time in seconds. When the timer expires, the charge goes into Fault mode.

  - `CHARGE_TIME` is the maximum charging time in seconds. When the timer expires, the charger goes into Fault mode. Both timers should be approximated using the battery capacity, charge current and specific chemistry behavior.
  - `IFLAT_COUNT` is a flat current counter used to end charge in case the battery has higher leakage than normal, and the charging current will not go below the normal end-of-charge current limit. The counter is re-initialized every time a new current minimum is recorded, otherwise it is decremented. When the counter reaches zero, it means that, for a number of seconds equal to the initialization value, no new minimum has been recorded. Basically, the battery charge current has not decreased for a long time.

Please note that, even though the charging algorithm is correct for many types of batteries, it is impossible to account for all variants and particularities. While this application note, hardware and firmware provide a nice starting point for high-power, intelligent charger applications, it is the user's responsibility to work with the battery manufacturer to get the optimum charging algorithms and parameters.

CONCLUSION

The SEPIC/inverse SEPIC (Zeta) converters are very attractive because of the input to output DC insulation, and the ability to generate output voltages, either lower or higher than the input. Unfortunately, it is quite complicated to drive these topologies synchronously, generally limiting their use in high-power applications.

Using the MCP14628 and a few extra components, a 100W synchronous Zeta converter was designed and built with an achieved efficiency of 95%. The phase lag between the input and output denies the use of pulse-by-pulse control techniques, but the new NCO peripheral available on the PIC16 family has opened some new options, previously available only to DSP type controllers.

The CC/CV charger library makes use of this hardware to easily implement robust charging algorithms for a number of popular battery types, such as lead-acid, lithium-cobalt, lithium-manganese, lithium iron phosphate, and nickel-zinc, thus making it a must-have in all battery operated devices.

REFERENCES

- PIC16(L)F1503 14-Pin Flash, 8-Bit MCU Data Sheet: http://ww1.microchip.com/downloads/en/DeviceDoc/41607A.pdf
- 2A Synchronous Buck Power MOSFET Driver: http://ww1.microchip.com/downloads/en/DeviceDoc/22083a.pdf
- MCP6V01/2/3 Data Sheet: http://ww1.microchip.com/downloads/en/DeviceDoc/22058c.pdf
- MCP1790/MCP1791 Data Sheet: http://ww1.microchip.com/downloads/en/DeviceDoc/22075b.pdf

APPENDIX A: REVISION HISTORY

Revision A (10/2012)

Initial Release.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

═ ISO/TS 16949 ═

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620766088

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/11