Network-Enabled High Performance Triple Conversion UPS

Venkat Anant

We wish to thank Diego Garay, Jorge Zambada, and Luis Reynoso for their contributions that comprised a significant portion upon which this application note was generated.

Contents

1. Introduction

The main purpose of an Uninterruptible Power Supply (UPS) is to provide clean and stable power to a load, regardless of power grid conditions such as black-outs. Uninterruptible Power Supplies have been widely used for office equipment, computers, communication systems, medical/life support and many other critical systems. Recent market requirements include a target expectation for UPS reliability of 99.999% power availability, performance demands of zero switch over time, and complex network connectivity and control methods, such as Simple Network Management Protocol (SNMP).

Presently, a UPS is often implemented with an MCU. But MCUs have significant price/performance limitations that can be rectified by implementing the UPS using a hybrid MCU with efficient digital signal processing capability. This has not been possible until recently, due to performance and the cost of the processors required to do the job. The 56F8300 series of digital signal controllers has the required performance, peripherals, and price targets to enable UPS designs to implement advanced features.

2. Freescale Semiconductor's Solution

The digital 56800/E network-enabled high-performance UPS reduces system component count, increases system reliability, and enables advanced functions while reducing cost. Key advantages of this digitally-controlled UPS include:

- Single-device solution which combines MCU functionality and DSP processing power
- Bidirectional AC/DC conversion
- High input power factor with Direct PFC and lower power pollution to the power grid
- Extended battery life and lower maintenance costs
- Power source and load conditioning can be monitored in real time
- Network communication for remote control and monitoring
- Expedites time-to-market using out-of-the-box software components

The functional blocks of the on-line triple conversion UPS are shown in **Figure 2-1**.

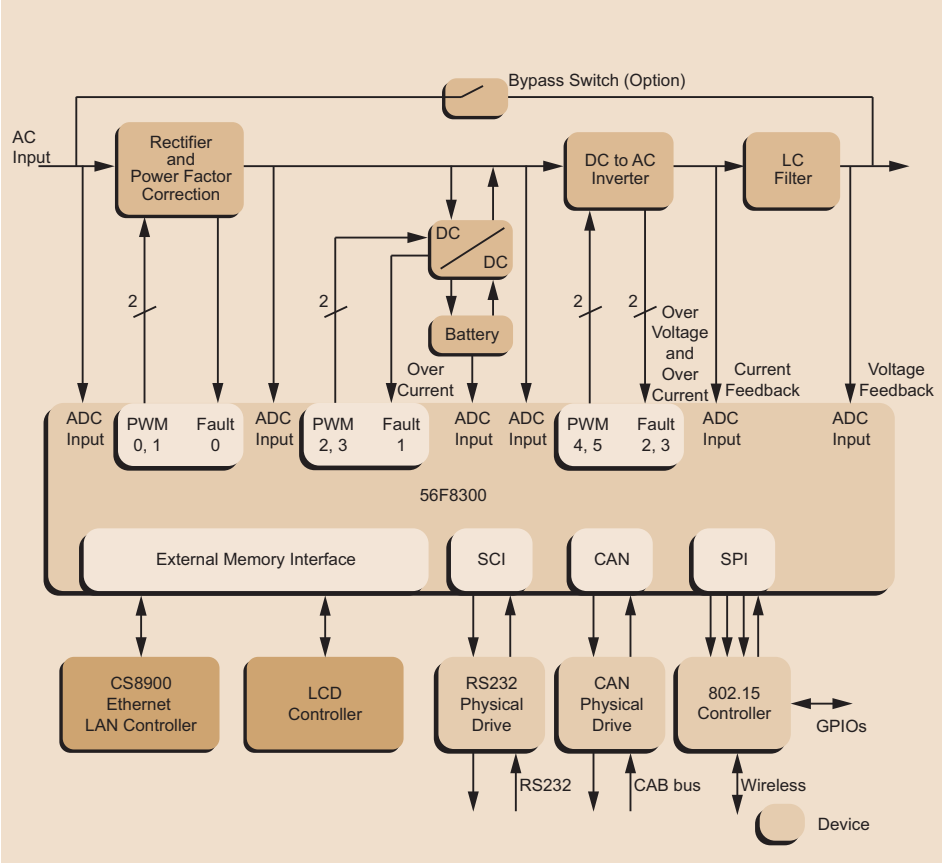


Figure 2-1. On-line Triple Conversion UPS Block Diagram

During normal operation, the AC input voltage is rectified by an AC/DC converter that rectifies the input voltage and regulates the input power factor. The output of the AC/DC converter is a DCBus voltage that is used as the source for both the battery charger and DC/AC inverter. The battery charger is a boost/buck DC/DC converter. When the system is charging the batteries, the DC/DC converter works in buck mode, which steps the high DCBus voltage down to the batteries' acceptable voltage level and charges the batteries. When the battery pack is fully charged, the converter switches to stand-by mode. If an input power failure occurs, the DC/DC converter works in boost mode, which supplies power to the DCBus from the batteries. The DC/AC inverter is used to convert the DC voltage to approximated sinusoidal output voltage pulses. The pulse string is input to an LC filter, which generates a true sinusoidal output voltage that is supplied to the load. The user can select the frequency of the sinusoidal output voltage and the frequency can either be synchronized to the input voltage frequency or any other desired independent stable frequency. When any failures or faults are generated or maintenance is needed in the UPS system, the bypass switch will be engaged, which turns the UPS system off and connects the load directly to the input power source. All necessary control functions are implemented within the controller, such as:

- Power on/off control
- DCBus voltage regulation
- Input power factor correction
- Battery management
- AC output voltage regulation
- Frequency synchronization of input and output
- Power source monitoring
- System self diagnostics and self protection
- Emergency event processing
- Real-time multi-tasking system operation
- Communication protocols

3. Advantages and Features of Freescale's Controller

The Freescale 56F8300 (56800E core) family is well suited for UPS design, combining the DSP's calculation capability with an MCU's controller features on a single chip. These controllers offer many dedicated peripherals, including Pulse Width Modulation (PWM) units, Analog-to-Digital Converters (ADC), timers, communication peripherals (SCI, SPI, CAN), on-board Flash and RAM. Generally, all the family members are appropriate for use in UPS design.

The following section uses a specific device to describe the family's features.

3.1 56F8346, 56800E Core Family

The 56F8346 provides the following peripheral blocks:

- Two Pulse Width Modulator units (PWMA and PWMB), each with six PWM outputs, three Current Sense inputs, and three Fault inputs for PWMA/PWMB; fault-tolerant design with dead time insertion, supporting both center-aligned and edge-aligned modes
- Two 12-bit Analog-to-Digital Converters (ADCs), supporting two simultaneous conversions with dual 4-pin multiplexed inputs; the ADC can be synchronized by PWM modules

- Two Quadrature Decoders (Quad Dec0 and Quad Dec1), each with four inputs, or two additional Quad Timers, A & B
- Two dedicated general purpose Quad Timers totaling three pins: Timer C, with one pin and Timer D, with two pins
- CAN 2.0 B-compatible module with 2-pin ports used to transmit and receive
- Two Serial Communication Interfaces (SCI0 and SCI1), each with two pins, or four additional GPIO lines
- Serial Peripheral Interface (SPI), with a configurable 4-pin port, or four additional GPIO lines
- Computer Operating Properly (COP) / Watchdog timer
- Two dedicated external interrupt pins
- 61 multiplexed General Purpose I/O (GPIO) pins
- External reset pin for hardware reset
- JTAG/On-Chip Emulation (OnCE)
- Software-programmable, Phase Lock Loop-based frequency synthesizer for the controller core clock
- Temperature Sensor system

3.2 Peripheral Description

PWM modules are the controller's key features enabling UPS control. Each PWM is double-buffered and includes interrupt controls. The PWM module provides a reference output to synchronize the Analog-to-Digital Converters. The PWM has the following features:

- Three complementary PWM signal pairs, or six independent PWM signals
- Features of complementary channel operation
- Dead time insertion
- Separate top and bottom pulse width correction via current status inputs or software
- Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM signals
- 15-bit resolution
- Half-cycle reload capability
- Integral reload rates from 1 to 16
- Individual software-controlled PWM outputs
- Mask and swap of PWM outputs
- Programmable fault protection
- Polarity control
- 20mA current sink capability on PWM pins
- Write-protectable registers

The UPS utilizes the PWM block set in the complementary PWM mode, permitting generation of control signals for all switches of the power stage with dead time insertion.

The Analog-to-Digital Converter (ADC) consists of a digital control module and two analog Sample and Hold (S/H) circuits. The ADC features:

- 12-bit resolution
- Maximum ADC clock frequency is 5MHz with 200ns period
- Single conversion time of 8.5 ADC clock cycles (8.5 x 200ns = 1.7μs)
- Additional conversion time of 6 ADC clock cycles (6 x 200ns = 1.2μs)
- Eight conversions in 26.5 ADC clock cycles (26.5 x 200ns = 5.3μs) using simultaneous mode
- ADC can be synchronized to the PWM via the sync signal
- Simultaneous or sequential sampling
- Internal multiplexer to select two of eight inputs
- Ability to sequentially scan and store up to eight measurements
- Ability to simultaneously sample and hold two inputs
- Optional interrupts at end of scan, if an out-of-range limit is exceeded, or at zero crossing
- Optional sample correction by subtracting a preprogrammed offset value
- Signed or unsigned result
- Single-ended or differential inputs

The Quad Timer is an extremely flexible module, providing all required services relating to time events. It has the following features:

- Each timer module consists of four 16-bit counters/timers
- Counts up/down
- Counters are cascadable
- Programmable count modulo
- Maximum count rate equals peripheral clock/2 when counting external events
- Maximum count rate equals peripheral clock when using internal clocks
- Counts once or repeatedly
- Counters are preloadable
- Counters can share available input pins
- Each counter has a separate prescaler
- Each counter has capture and compare capability

4. Online UPS Theory and Description

4.1 Introduction

Uninterruptible Power Supplies (UPS) are electronic devices designed to provide power to critical mission systems. An Online UPS (OUPS) provides continuous power to the load during power outage or glitches caused by power source switching.

4.1.1 The Concept of an Online UPS

The minimum components needed to design an Online UPS are the rectifier, the battery bank and the inverter. The rectifier converts the distribution line's AC (Alternating Current) power to DC (Direct Current) power, the form of current suitable to store energy in a battery bank. At all times, this DC is also fed to an inverter, which reconverts the DC power to an AC waveform connected to any equipment utilizing AC that a user considers as mission critical. If the AC supply fails for any reason, the inverter will continue to draw power from the batteries.

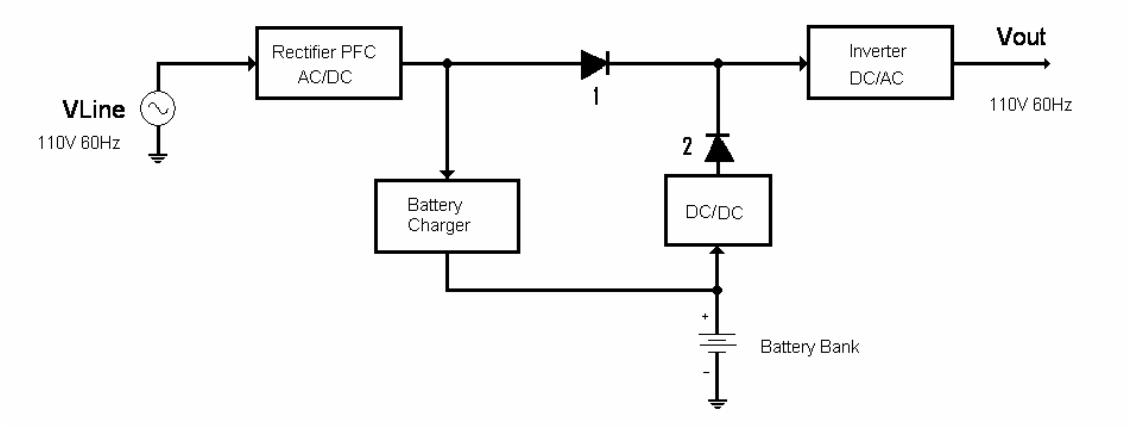


Figure 4-1. A Basic Online UPS

4.1.2 Input Power Factor Control (PFC)

When a sinusoidal input signal is connected to a full wave rectifier, conduction will occur only during the peaks of the signal. This causes a two-fold inconvenience to the electricity distribution line:

- Insertion of harmonics to the lines
- High current peaks, which imply greater losses on the distribution

These effects are aggravated by the long distances the electric distribution networks usually span.

From the electrical utility's point of view, the best possible load is the pure resistive: The current waveform should be a pure sinusoidal waveform identical to the voltage waveform and of the same frequency and phase.

In order to show a resistive load to the utility lines, the input current to the UPS is controlled (i.e., modulated) to make it match a set point. This set point depends on the input voltage waveform, and its amplitude is dependent on the equipment's power consumption.

4.1.3 DC/DC Converters

If a rectifier is connected to the AC line supply, then the DC voltage will be equal to the peak voltage of the line. (i.e., in a 120 V_{RMS} line, the peak will be 120√2, 170V). If the battery bank is configured for 12 or 24 V_{DC}, the UPS works by using DC/DC converters.

For an online UPS, two power DC/DC converters are required. One converter operates as the battery charger, and the other boosts the battery voltage in the absence of line input and generates the appropriate DC required by the inverter.

4.1.4 Phase Locked Loop (PLL)

This UPS can operate in the Free Running mode or in the Locked-to-Line mode. If the AC main line frequency is at the nominal value of 50Hz or 60Hz ± 5%, the PLL locks the inverter output to the line. If the AC main line frequency runs out of limits for any reason, the UPS will automatically switch to run locked to the internal frequency reference.

The UPS will also work in the Free Running mode if commanded to operate as a frequency converter. For example, it can connect to a 60Hz AC main line frequency and output a signal of 50Hz frequency, and vice versa.

The purpose of phase locking the inverter to the line input is to enable the automatic bypass feature and to avoid signal “mixing” at the rails. These two features are detailed in the following sections.

4.1.5 Bypass Operation

In order to allow a UPS bypass without loss of power at the load, two conditions must be met:

- The inverter output must be locked to the frequency and phase of the AC main line
- The inverter output and the AC main line’s RMS voltages must be within 10% of one another

When the bypass conditions are met, the bypass switch can transfer the load to the AC main line in the event of a UPS failure or when commanded by the operator during routine maintenance. It can also switch the load back to the inverter after any maintenance.

4.1.6 Rail Ripple

The energy to support the load is stored in the rail capacitors. These capacitors are current-fed by the PFC circuitry in the Online mode or by the battery booster in the Battery Back-up mode.

In the Online mode, a ripple with the phase and twice the frequency of the AC main line will be present at the rails, superimposed with a ripple with the phase and twice the frequency of the inverter current. In this situation, if a frequency offset Δf is present, lower-order components can appear. Locked operation is preferred to minimize the effects of frequency mixing.

4.1.7 Pulse Width Modulation (PWM)

High-power control requires switchable electronic devices, precluding their use in the active region, where power dissipation in the device is very high. For this reason, control is made by pulse width modulation, where the duty cycle of a signal is modified, then a linear filtering device passes the desired signal value to the analog components. PWM is then used to implement inverters, PFCs, and DC/DC converters.

4.1.8 A Controller Solution to Control a UPS

The control system for a UPS must accomplish the following functions:

- Control strategies for inverter, PFC, PLL, and DC/DC converters. Every control loop starts at an Analog-to-Digital Converter (ADC) in order to sense the signals, and ends at a Pulse Width Modulator as an actuator.
- Deciding when to activate or deactivate a component
- Detecting failure conditions and implementing any required action
- Enabling Monitor and Control (M & C) communications

Compared to traditional analog controls, today's low-cost and high-performance controllers provide a better solution in performance and cost. A single MCU includes a powerful processor core and such peripherals as PWMs, Timers, and Analog-to-Digital Converters. A single 56800E device is able to assume the monitoring and real-time control required by an Online UPS.

4.2 System Overview

Figure 4-2 depicts a simplified UPS system.

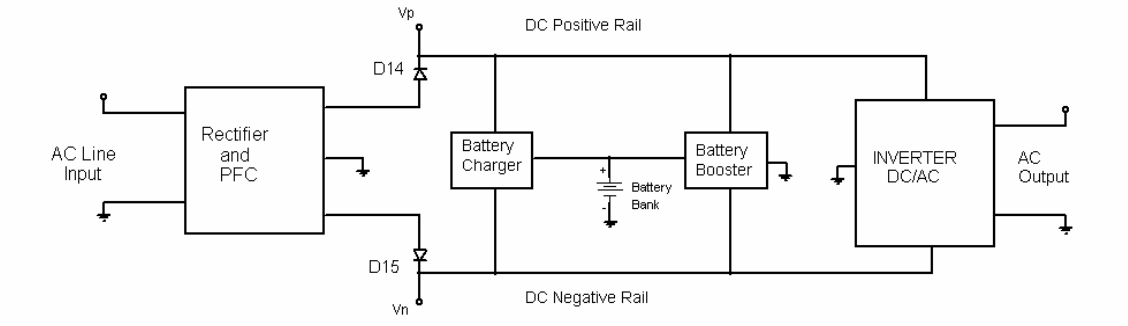


Figure 4-2. Simplified UPS Schematic

Figure 4-3 shows a photo of a completed UPS prototype. The system's power electronics and ferromagnetic components are detailed on the left side of the the figure.

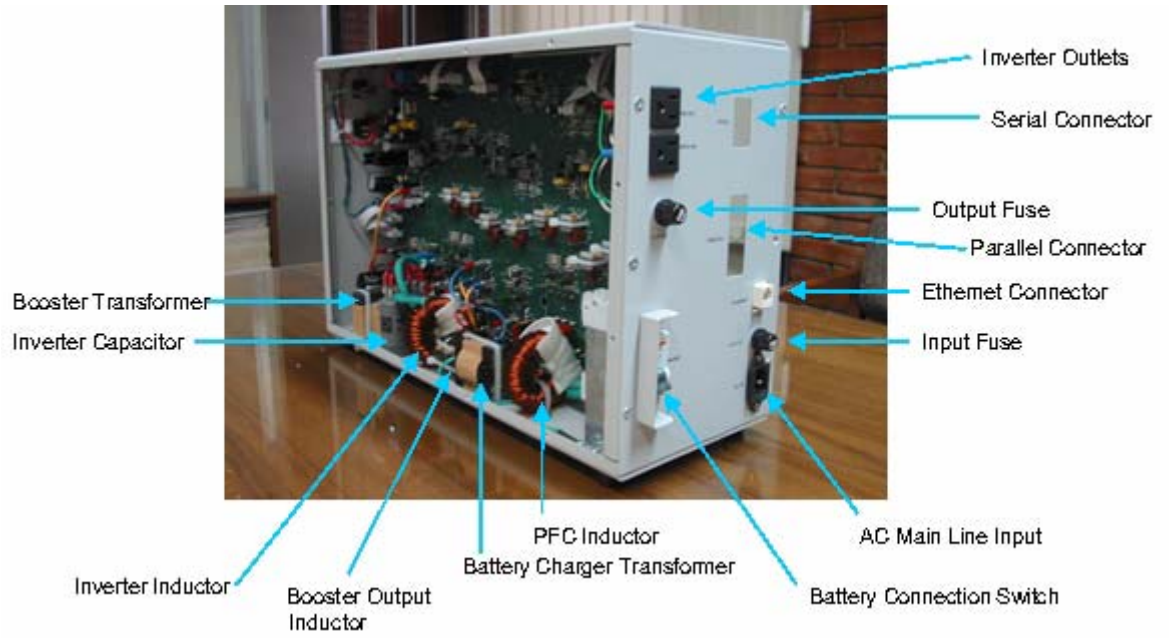


Figure 4-3. Prototype UPS

4.3 System Actuators

A simplified schematic of the controller's relationship with actuators is shown in **Figure 4-4**, where all switches represent MOSFETs or IGBTs.

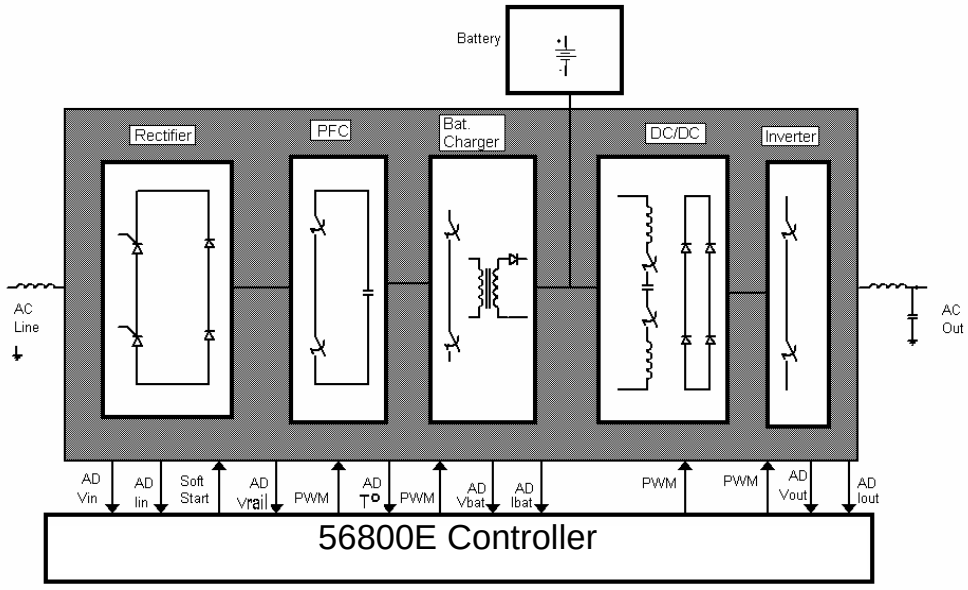


Figure 4-4. Relationship between a 56800E and Power Actuators

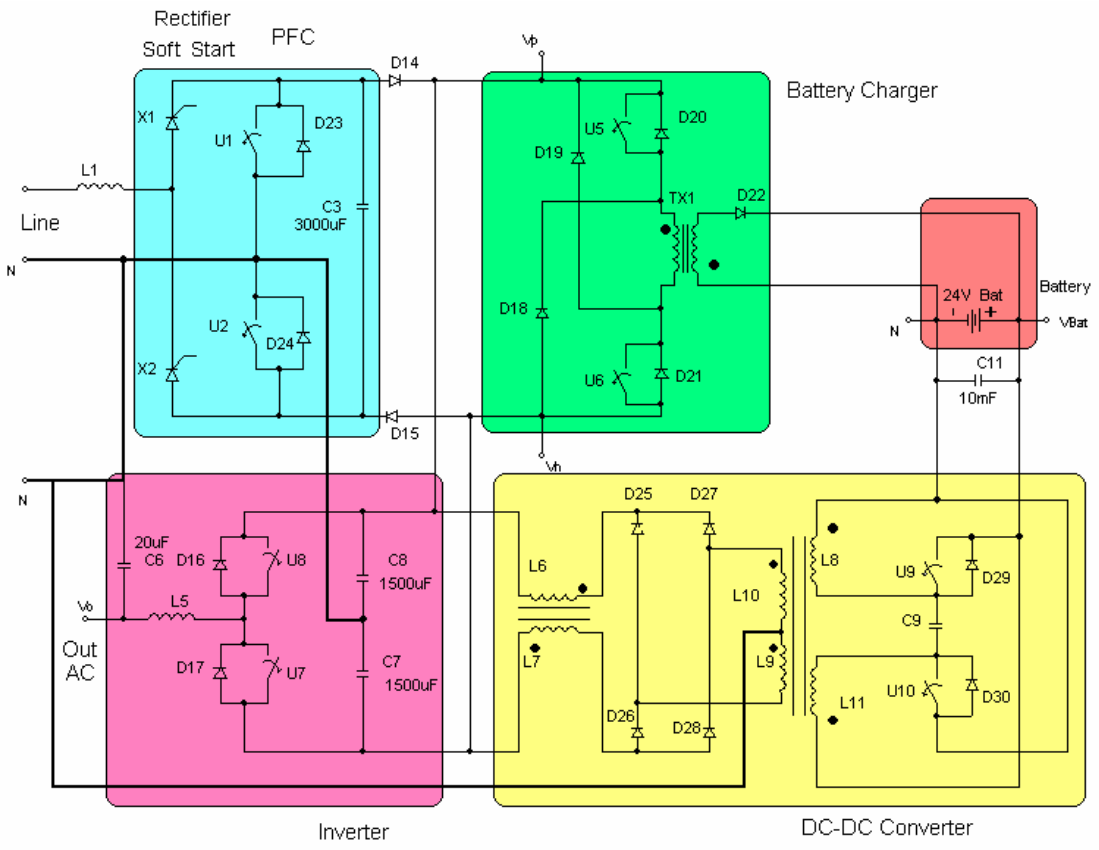


Figure 4-5. Simplified Schematic Diagram of the UPS

4.4 Input Rectifier Theory of Operation

The input rectifier is implemented as a four-diode bridge (X1, X2, D24, and D23). A soft start system implemented with SCRs can prevent a huge in-rush current when the system starts, while the system's internal capacitors get charged to the line's peak voltage.

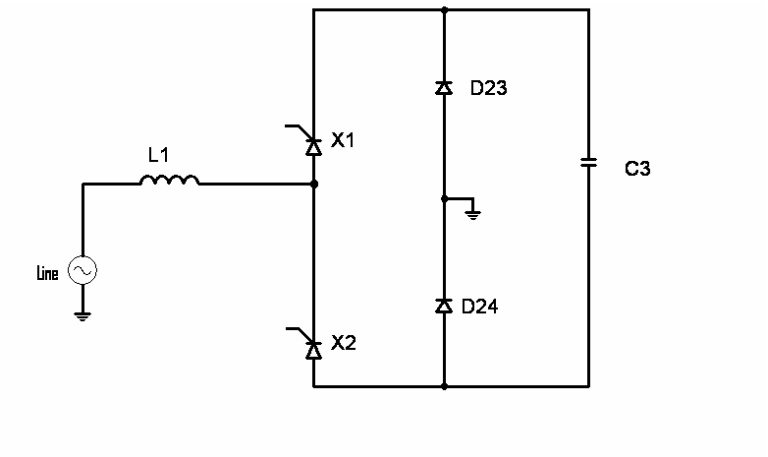


Figure 4-6. Input Rectifier

4.4.1 Rectifier Soft Start

If a high voltage is applied to a discharged capacitor, its low impedance will result in a very high inrush current across the circuit, reducing the components' longevity.

A soft start circuit is designed to avoid that circumstance. **Figure 4-7** shows a full wave-controlled rectifier bridge. If the trigger angle of X1 and X2 is gradually decreased from the zero crossing towards the peak voltage, as demonstrated in **Figure 4-8**, the capacitor voltage will then increase slowly. As the current on a capacitor equals the capacitance value times the voltage derivative with respect to time, the input current will be proportional to the slope of the voltage applied to the capacitor.

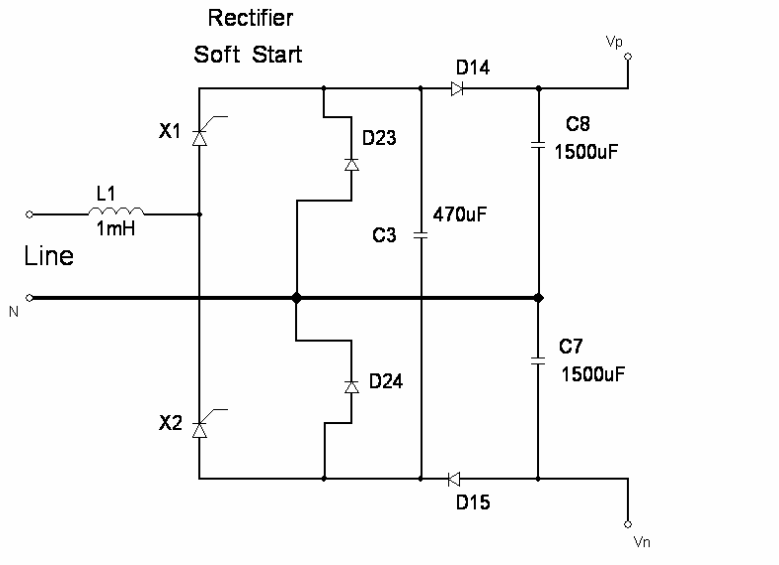


Figure 4-7. Full Wave-Controlled Rectifier Bridge

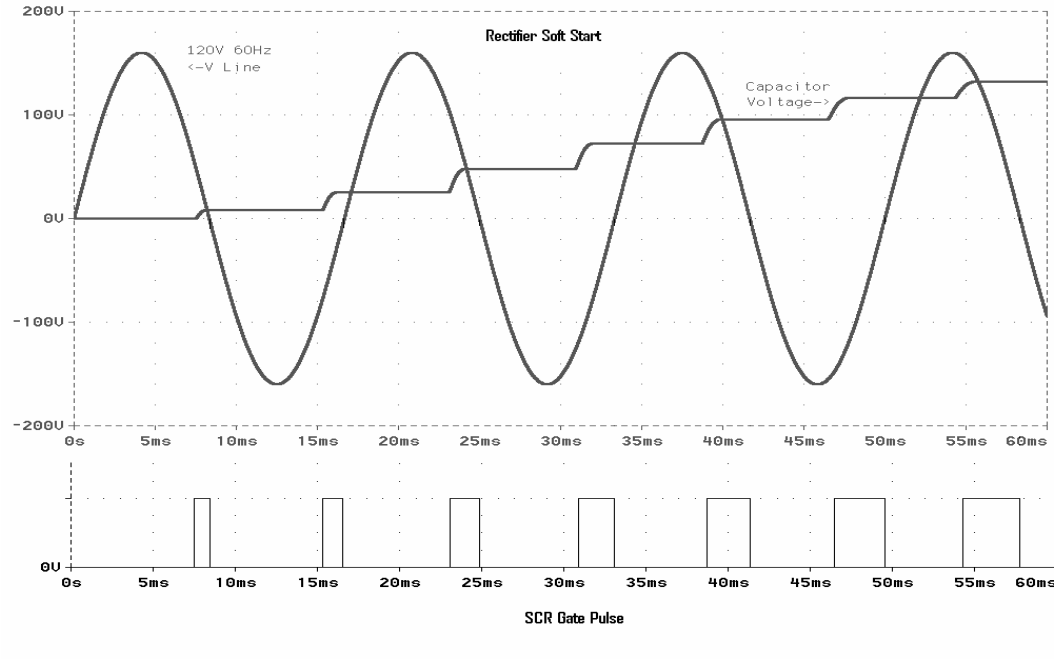


Figure 4-8. Relationship between Rectifier Soft Start Operation / Actuator Signals and the AC Main Line Voltage

4.5 Power Factor Corrector (PFC) Theory of Operation

After the rectifier soft start finishes, X1 and X2 must act as diodes with continuous triggers. When no PFC is implemented, the line current will be similar to that shown in **Figure 4-9**, due to the diode–capacitor nature of a rectifier.

The objective of the PFC circuit is to simulate a resistive load to the power line; in other words, to obtain a unity power factor and low harmonic content in the current waveform. A fast control must be implemented in order to make the current waveform follow the AC voltage, while elevating and controlling the rail voltage and supplying the average power required to the load. **Figure 4-10** shows how the PFC works, illustrating a current signal waveform similar in form to the voltage waveform. The ripple in the figure is a consequence of the IGBT high frequency switching.

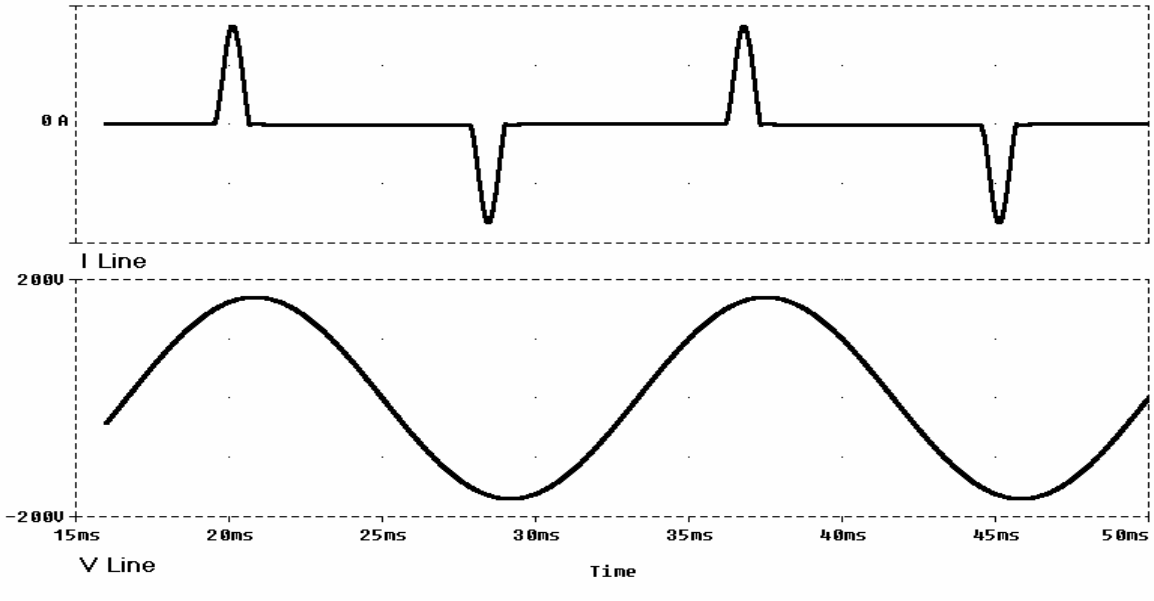


Figure 4-9. Typical Rectifier Current vs. AC Line Voltage

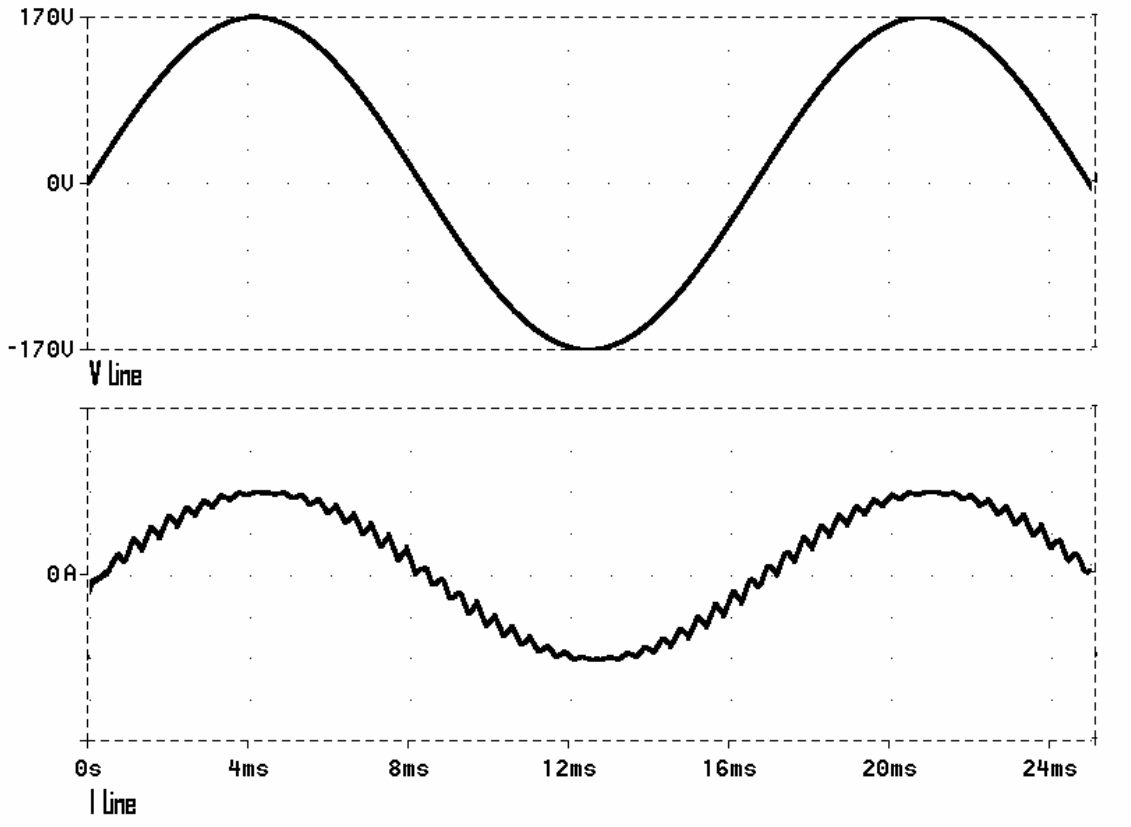


Figure 4-10. PFC Current and Voltage Waveforms

Once the voltage on capacitors C_3, C_7, and C_8, shown in **Figure 4-11**, reach the AC main supply peak after the rectifier soft start, the PFC is turned on, correcting the power factor presented to the line and generating the rail DC voltages V_P and V_N at a value higher than the line peak.

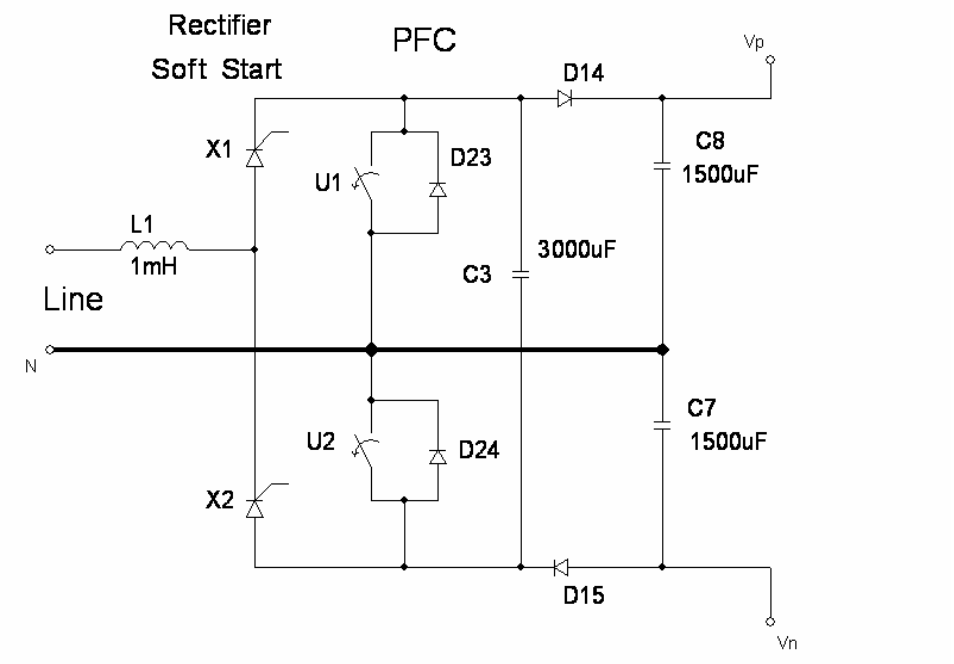


Figure 4-11. PFC Schematic

In order to work as a PFC, U_1 and U_2 in **Figure 4-11** turn on and off in complementary mode. When U_1 is on, U_2 is off, and vice versa.

The switching frequency of operation is 20kHz. The line nominal frequencies are 50Hz or 60Hz, so it is a valid approximation to consider the line voltage as a constant during a switching period. For positive values of the line, when U_2 is on (closed) and U_1 is off (open), the circuit reduces to that shown in **Figure 4-12**, where C_3 is connected in parallel to C_8, causing the voltages on these capacitors to be the same, thus reducing the ripple voltage on C_8.

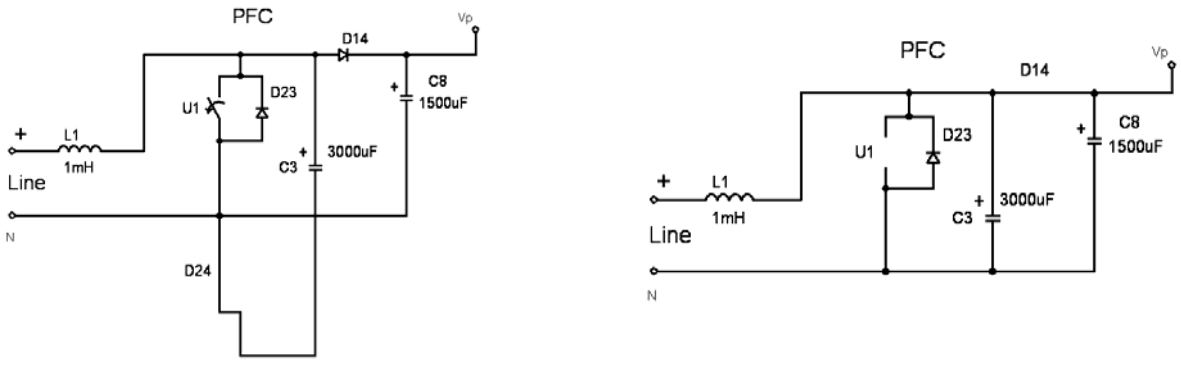


Figure 4-12. Partial PFC Schematic when U_1 Is Open and U_2 Is Closed

For positive values of the line, when U_1 is closed and U_2 is open, the circuit reduces as shown in **Figure 4-13**. C_3 is now connected in parallel to C_7, thus reducing its ripple voltage. The line is applied directly to L_1, increasing the current across it with a constant slope, because line voltage could be considered constant, and the inductor current, which corresponds to the current in the line (I_{LINE}), depends on the voltage across its terminals.

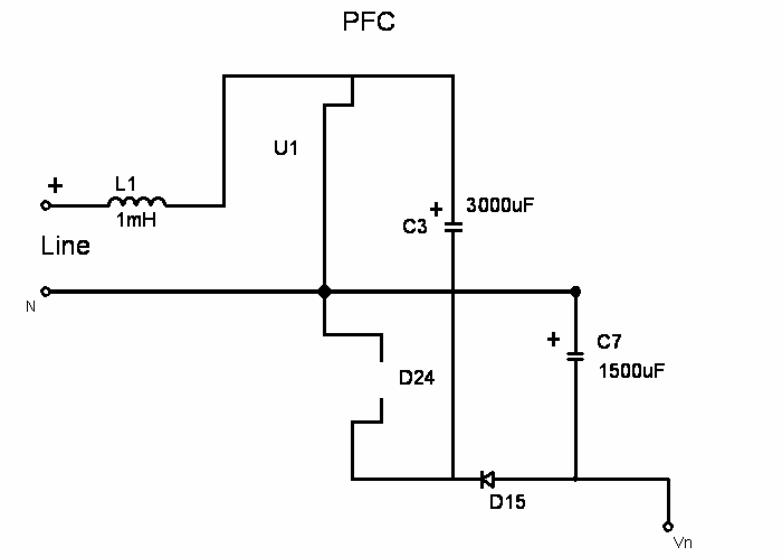


Figure 4-13. Partial PFC Schematic when U_1 Is Closed and U_2 Is Open

Inductor current is calculated by the equation:

$$\Delta I_{line} = \frac{1}{L}\int_{t_1}^{t_2} V_{L_1}\,dt$$

where:
V_{L1} is the voltage across the inductor terminals

The peak current across the inductor at time t_2 depends, among other factors, on the instant value of the line voltage and the time difference $t_2 - t_1$, which is the time that U_1 remains closed.

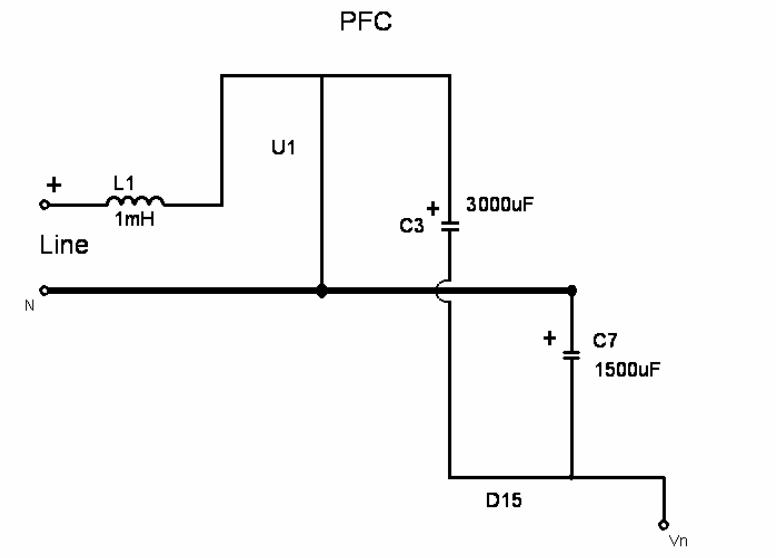


Figure 4-14. Resulting Parallel Connection between C3 and C7

The voltage-boosting characteristic of the PFC is accomplished by increasing the voltage across C_3 (and consequently, across C_7 and C_8). Given that a current is circulating in the inductor L_1, when U_1 opens, the voltage across L_1 adds to the line voltage as shown in **Figure 4-15**. This forces D_{24} into a conduction state and C_3 and C_8 to charge to the addition of the line and the inductor terminal voltage, V_{L1}. This is a typical boost configuration.

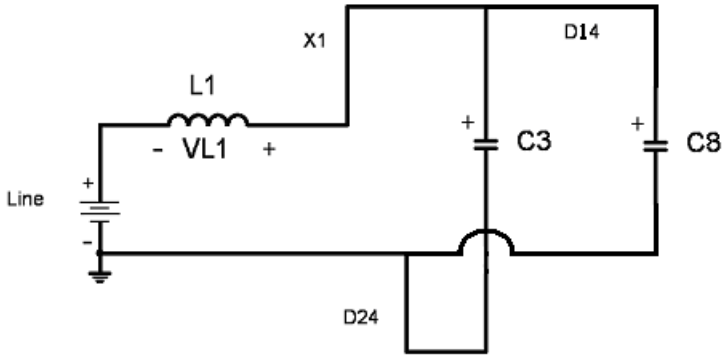


Figure 4-15. Voltage Boost across Capacitors C_3 and C_8

Due to symmetry, this circuit works in the same way for negative values in line voltage.

4.6 Battery Charger Theory of Operation

Figure 4-16 has been extracted from the UPS schematic shown in Figure 4-5. Using the high DC voltages V_P rail positive and V_N rail negative as its power sources, this circuit provides the charge conditions for a battery bank formed by two 12V batteries connected in series, which must be charged with constant current. When the float condition is reached, the charger must preserve a constant voltage while providing the battery bank's self-discharge current.

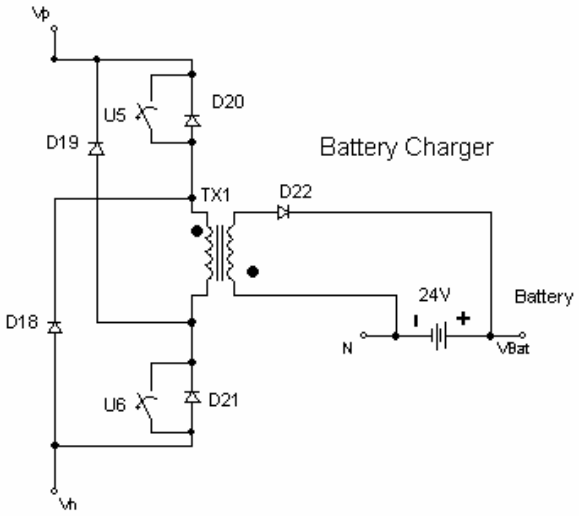


Figure 4-16. Battery Charger Schematic

The battery charger is an application of a two-transistor flyback configuration using a coupling inductor rather than a transformer. Because this operating mode implies no flow of current in the secondary when the primary has a non-zero current, and vice versa, it means that no current flows simultaneously in both windings.

Figure 4-16 shows a two-transistor version of a flyback converter, where U_5 and U_6 are simultaneously turned on and off. The advantage of such a topology over a single-transistor flyback converter is that the switches' voltage rating is V_P - V_N. Moreover, since a current path exists through the diodes D_{18} and D_{19}, which are connected to the primary winding, a dissipative snubber across the primary winding is not needed to dissipate the energy associated with the transformer primary-winding leakage inductance.

The design, calculations and construction of TX_1 are critical in order to prevent the reflected voltage from secondary to primary rising higher than V_P - V_N when D_{22} is on.

4.7 Battery Booster Theory of Operation

The battery booster is a DC/DC converter and transforms the battery voltage of $24V_{DC}$ to the required differential $440V_{DC}$ between rails. The rails are symmetric, implying $V_P = 220V_{DC}$ at the positive rail, and $V_N = -220V_{DC}$ at the negative rail, as shown in Figure 4-17.

Although the topology of Figure 4-17 is usually called "booster" because its output voltage is higher than input voltage, it is actually a push-pull converter with an arrangement of two forward converters working alternatively and a transformer to increase the output voltage.

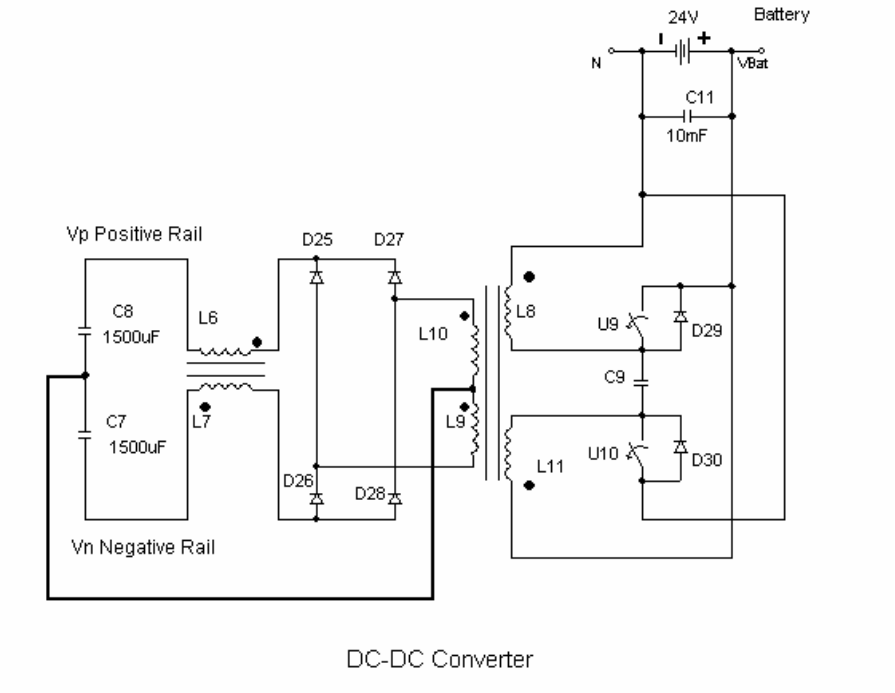


Figure 4-17. Battery Booster Schematic

The switches U_9 and U_{10} cannot be closed at the same time. Typical drive signals are illustrated in **Figure 4-18**.

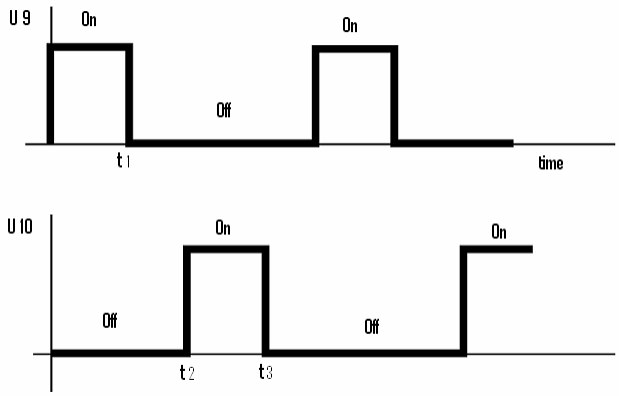


Figure 4-18. Drive Signals for Battery Booster Switches

Using a control signal like the one shown in **Figure 4-18**, the waveforms at the transformer secondary over L_{10} and L_9 are illustrated with positive and negative values of voltage in **Figure 4-19**.

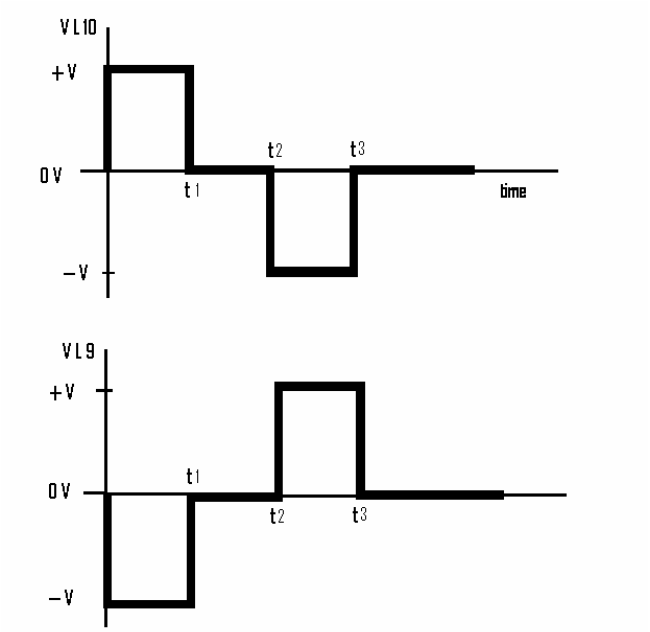


Figure 4-19. Signals at Transformer Secondary Windings L_{10} and L_9

C_9 remains charged to V_{BAT} and is added because the coupling factor is not equal to one, implying that a leakage inductor must be considered. C_9 acts as a snubber, creating a current path to avoid an uncontrolled voltage peak at the primary windings of the transformer.

The four diodes D25, D26, D27 and D28 form a conventional full-wave rectifier bridge and generate only positive values in the cathodes of D_{25} and D_{27} and negative values at the anodes of D_{26} and D_{28}, as shown in **Figure 4-20**.

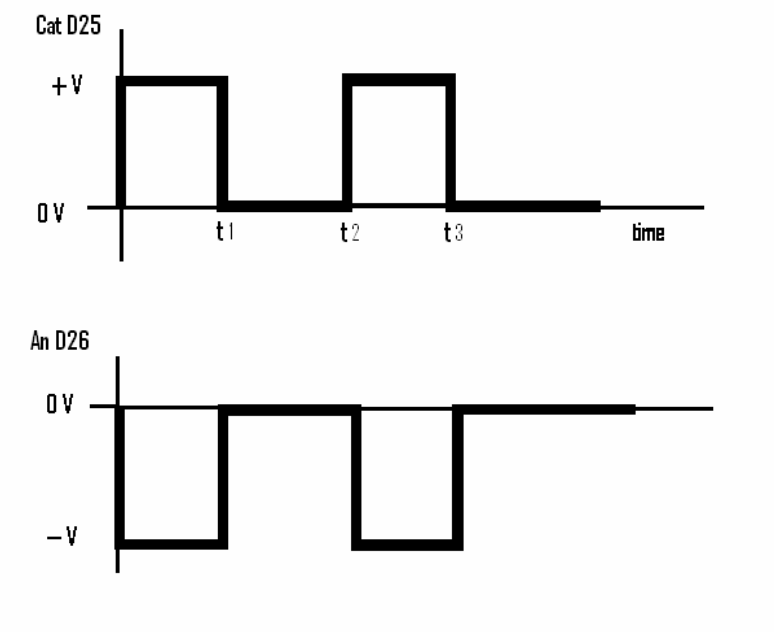


Figure 4-20. Signals at the Cathode of D_{25} and Anode of D_{26}

The objective of DC/DC converters is to generate a DC voltage. In order to filter any undesirable AC components, two LC filters are added:

- L_7 - C_7 for the negative rail
- L_6 - C_8 for the positive rail

4.8 Inverter Theory of Operation

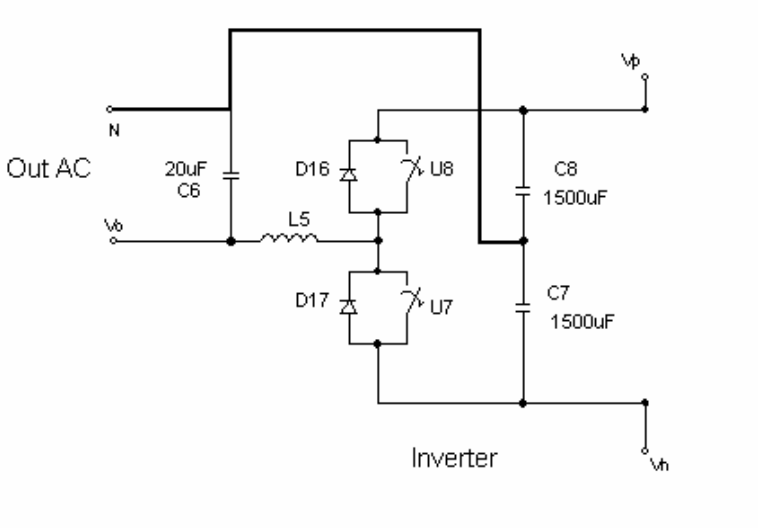


Figure 4-21. Inverter Schematic Diagram

The chosen inverter configuration is a half-bridge monophasic circuit. It must generate a low-distortion sinusoidal waveform in the output terminals from the V_P and V_N DC rail voltages. To produce such a signal, V_P and V_N must be higher than the AC positive and negative peak voltages, respectively.

Consider a switching period of T seconds. Assume that U_8 is closed during T_P seconds, while U_7 is open. During the remaining $T - T_P$ seconds, U_7 is closed and U_8 open. If the cutoff frequency of the low-pass filter composed by inductor L_5 and C_6 is low enough to reject the 1/T Hz switching frequency, then the output approximates to:

$$V_o = V_P \frac{t_p}{T} + V_N\left(1 - \frac{t_p}{T}\right)$$

$$\frac{t_p}{T} < 1, \quad \frac{1}{T} = 20\text{kHz}$$

where:

tp/T is the duty cycle of the control signal

The expression simplifies to :

$$V_o = (V_P - V_N)\frac{t_p}{T} + V_N$$

As 1/T = 20kHz is much higher than 50Hz or 60Hz, then the output voltage will result proportional to T_P, and the sinusoidal signal can be considered constant during T seconds.

4.9 Pulse Width Modulation

If a sinusoidal reference signal is compared to a symmetric triangle wave, the resulting signal is pulse width modulated, as shown in **Figure 4-22**. The triangle waveform frequency corresponds to the PWM switching frequency.

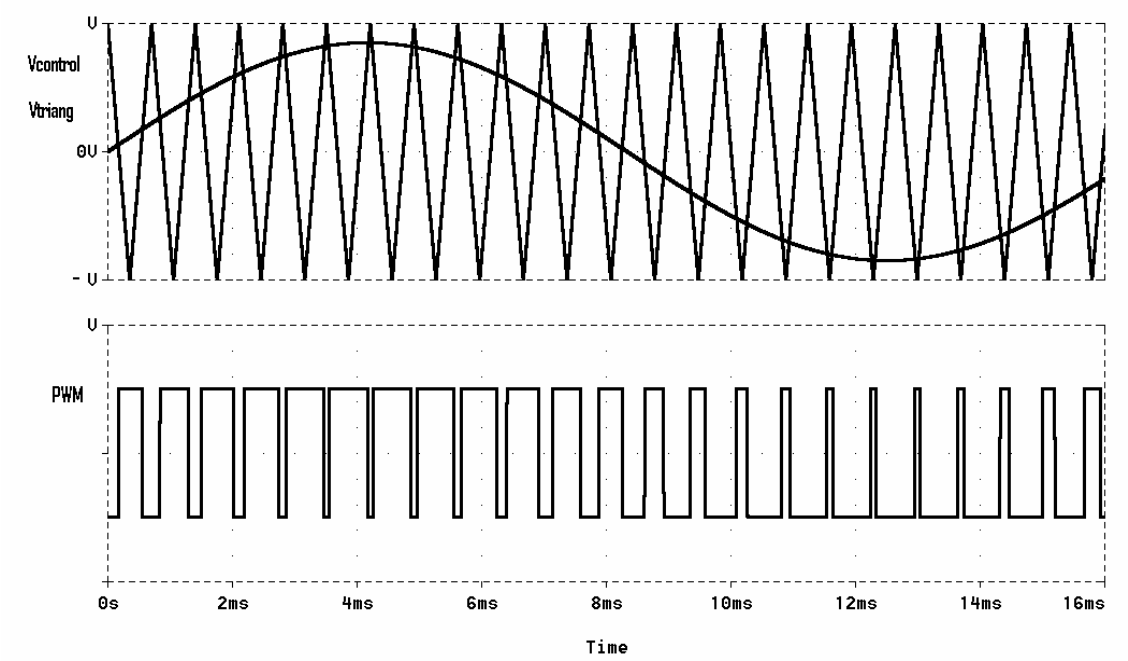


Figure 4-22. Generation of a PWM Signal

4.10 Auxiliary Circuits

Additional circuits are required to make the complete system operational and include:

- Power supplies
- Signal sensing circuitry
- Limiters
- Isolation circuits
- Driver circuits for SCRs, NMOSFETs and IGBTs

4.10.1 Power Supplies and Isolation Circuitry

This system requires multiple power supplies with floating references. The typical configuration is shown in **Figure 4-23**, and consists of a flyback converter similar to the one used in the battery charger. A 100kHz switching signal with a duty cycle of 35% is provided by the 56800E controller and applied to an NMOS technology H bridge.

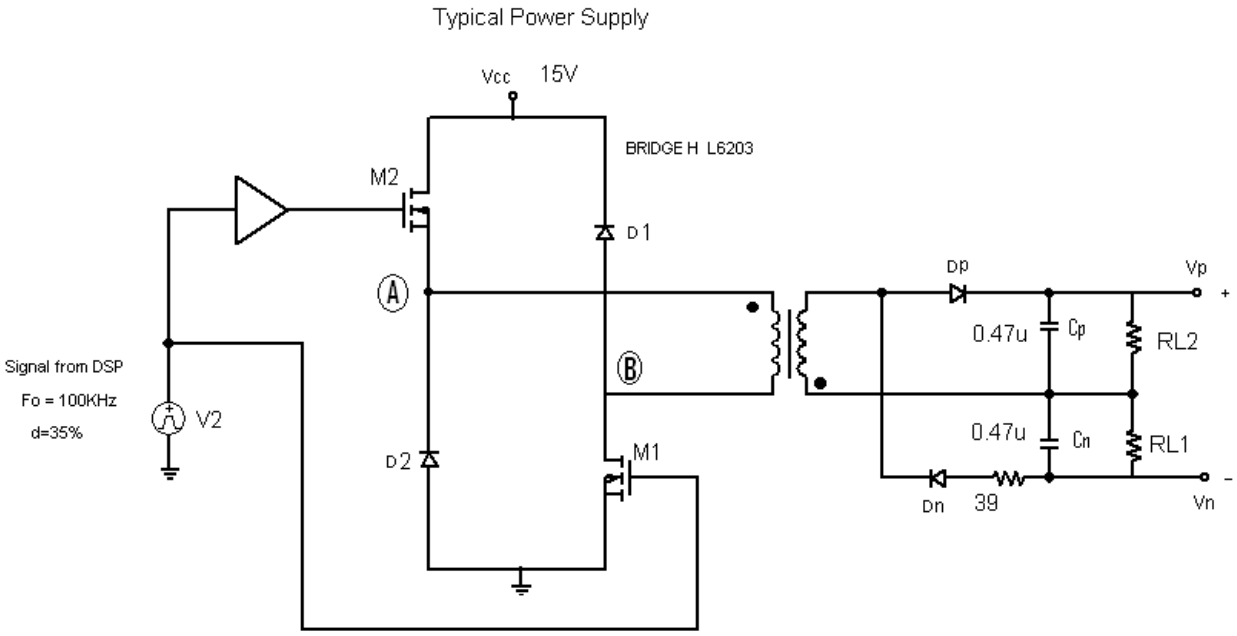


Figure 4-23. H Bridge Configuration Providing Multiple Floating Power Supplies

The output of the isolating transformer is half-wave rectified and applied to the load. Positive and negative voltages are generated. The 39Ω resistor limits the current of the transformer's secondary when M_1 and M_2 are on. Resistors RL_1 and RL_2 represent the load.

All isolated power supplies are connected in parallel to the rail voltage, depicted as nodes A and B in **Figure 4-23**.

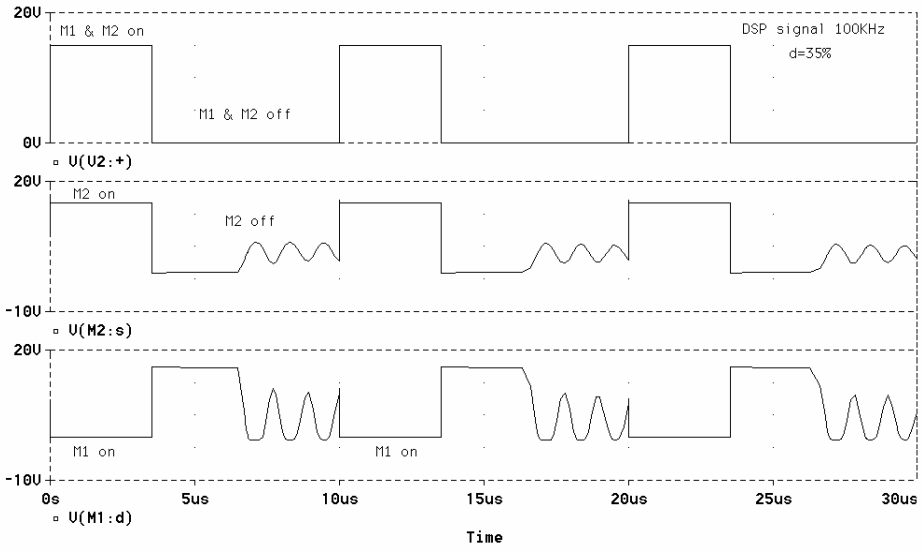


Figure 4-24. Waveforms at A and B

While M_1 and M_2 are on, V_{CC} is connected directly to the primary, increasing the current linearly, as shown in **Figure 4-25**. C_n is simultaneously charging across D_n. The 39Ω resistor acts as a current limiter. M_1, M_2, D_p, C_p, and the transformer shape a traditional flyback power supply.

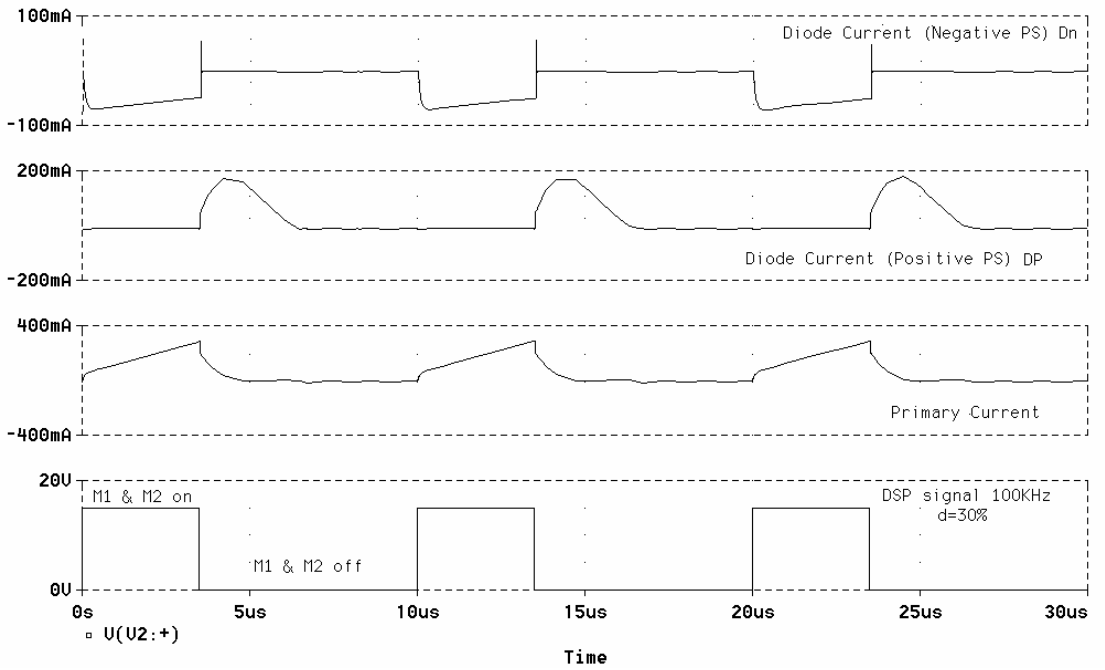


Figure 4-25. Current Waveforms

When M_1 and M_2 are off, an inverted voltage is induced in the primary and D_1, D_2 which limit that voltage to 15V plus 1.2 from two diode junctures, turning D_p on and charging C_p.

There are several transformers connected to the rails A and B which generate isolated power supplies to drive SCRs, IGBTs, and MOSFETs used in the rectifier, inverter, battery charger, and battery booster.

Figure 4-26 shows the control power supply, which uses a LM2576 step-down voltage regulator, to generate $+V_{CC}$ = 15V fed by the redundant DC voltage coming from $18V_{AC}$ or $+V_{BAT}$ or an additional power supply V_{DCR} coming from the battery charger. This circuit also generates the Control Board Power Supply, which is isolated from $+V_{CC}$.

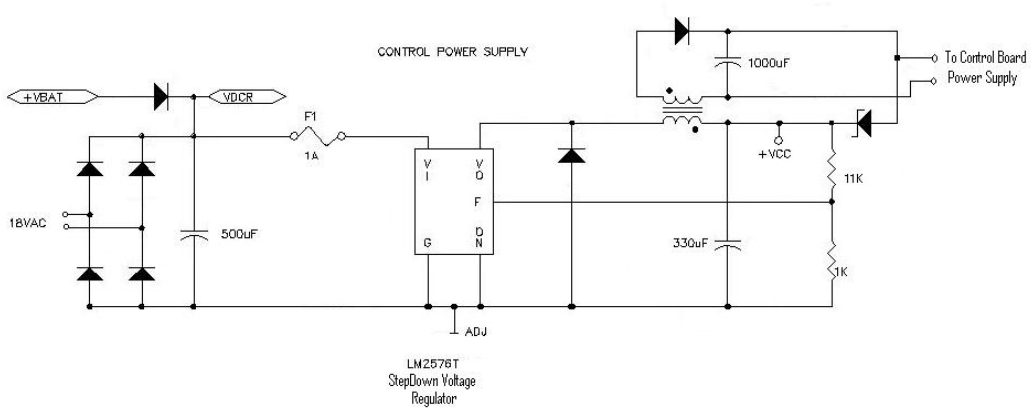


Figure 4-26. Control Power Supply

4.10.2 Sensing Circuits and Reference Voltage Generator

All signals that require sensing as voltages, currents, and temperature at the Analog-to-Digital Converters are converted to the appropriate input levels and diode-limited to avoid damage to the ADCs; this process uses differential amplifiers as seen in **Figure 4-28**, because of the different ground references shown in **Figure 4-27**.

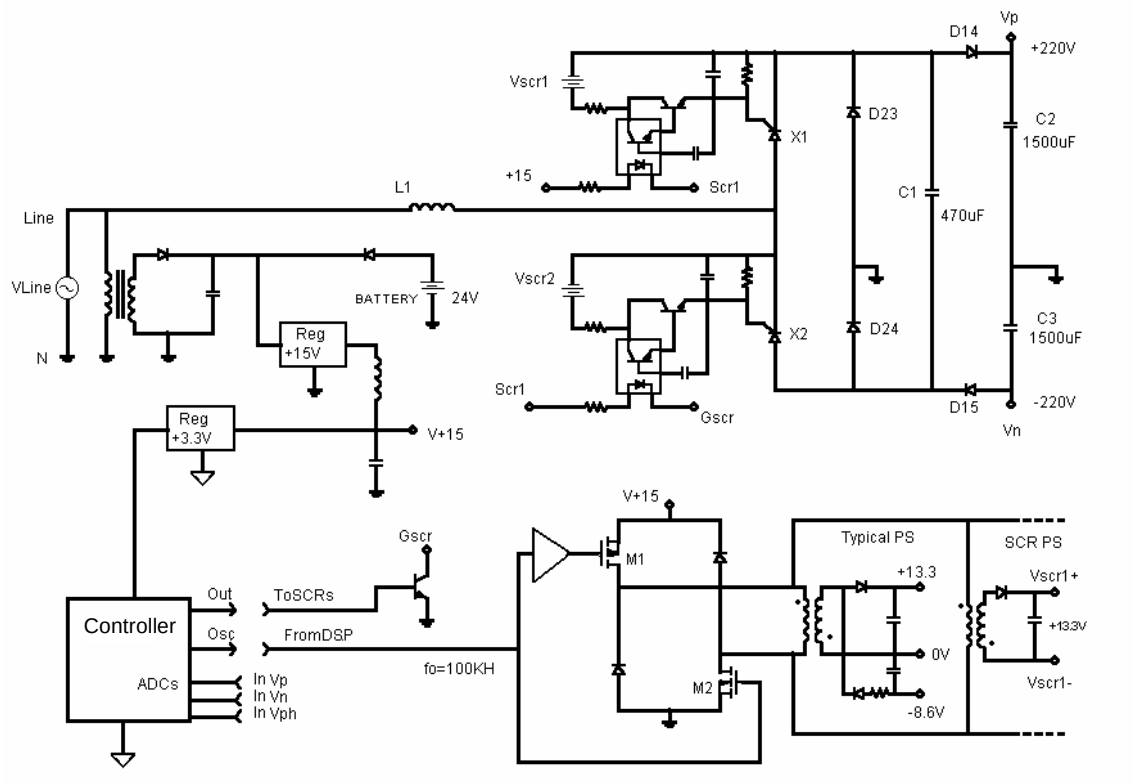


Figure 4-27. Partial View of the Auxiliary Power Supply and Optoisolation Network

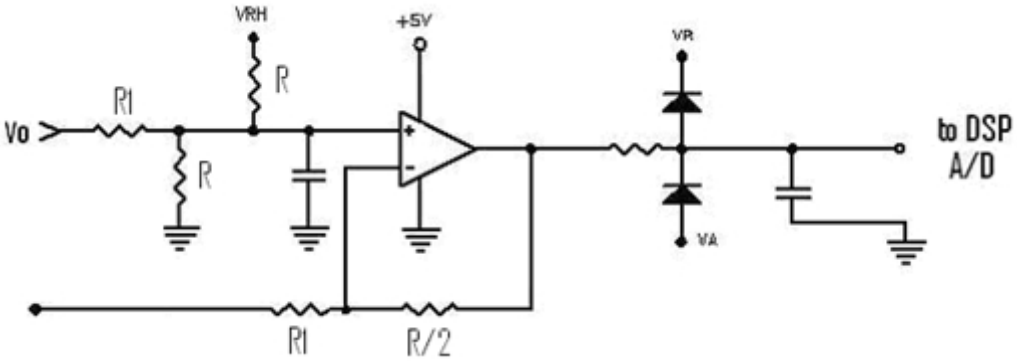


Figure 4-28. A/D Sensing Circuitry

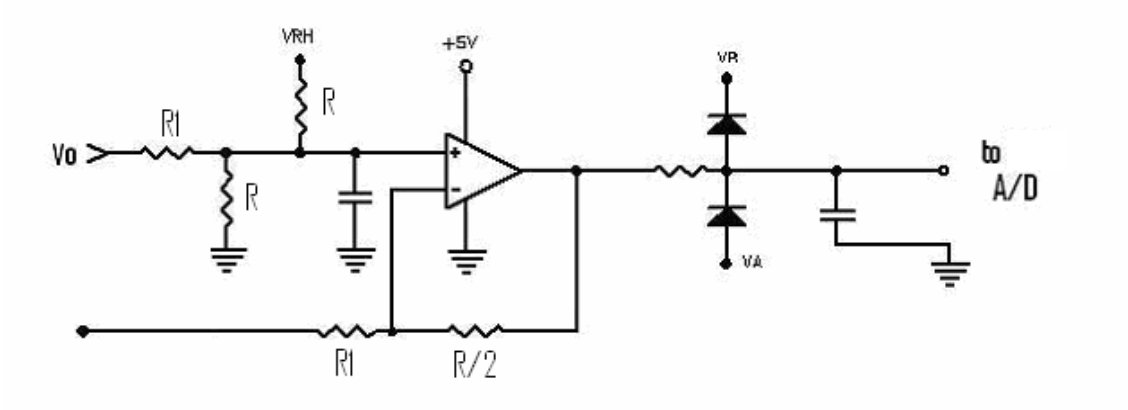


Figure 4-29. General Design of the Sensing Circuits

The value of the resistors required to sense every signal is calculated in order to guarantee full swing, minimizing analog and quantization noise at the ADCs.

The reference level of the ADC is used to shift the AC input signal to swing from 0 to V_{RH} (i.e., 0 volts in the input signal are mapped to $V_{RH}/2$). If the output of the operational amplifier tends to go out of limits for any reason, the diodes protect the ADC inputs.

4.10.3 Voltage Reference Supply

The voltage reference, V_{RH}, is generated by the circuit shown in **Figure 4-29**. This circuit acts as a buffer to the reference voltage V_{REFH} from the controller. This circuit also generates two auxiliary voltage outputs:

- V_A equal to one diode voltage $V\gamma$
- V_B with value V_{RH}-$V\gamma$

These voltages will be used to limit the values of voltage applied to A/D modules inside the controller.

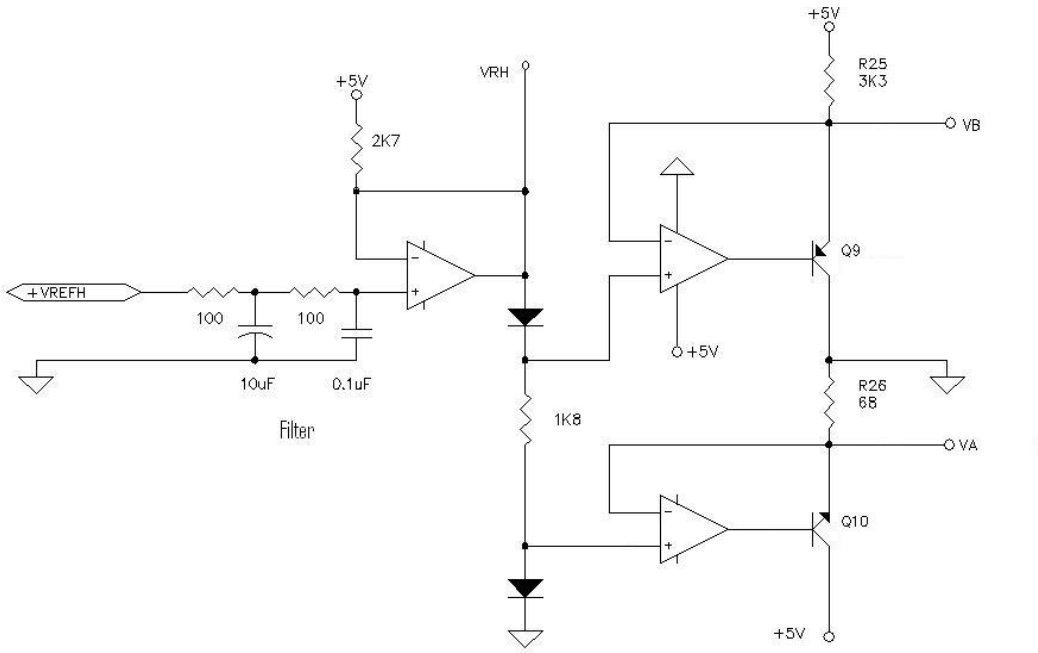


Figure 4-30. Voltage Reference Generator

4.10.4 Silicon Controlled Rectifier (SCR) Gate Drivers

The circuit in **Figure 4-30** shows the basic driver which handles the SCRs' gates using an 4N35 optocoupler to generate an isolated current supply to trigger the rectifier's SCRs.

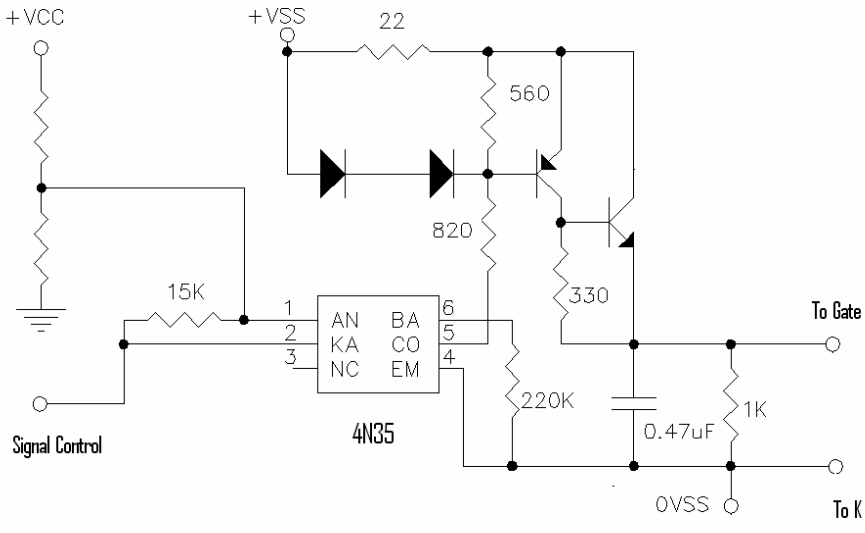


Figure 4-31. SCR Gate Driver

4.10.5 IGBT and MOSFET Gate Drivers Circuit

Figure 4-32 illustrates the basic isolated circuit implemented to drive the IGBT's and MOSFET's gates. It is based in an HCPL 3101 gate drive optocoupler.

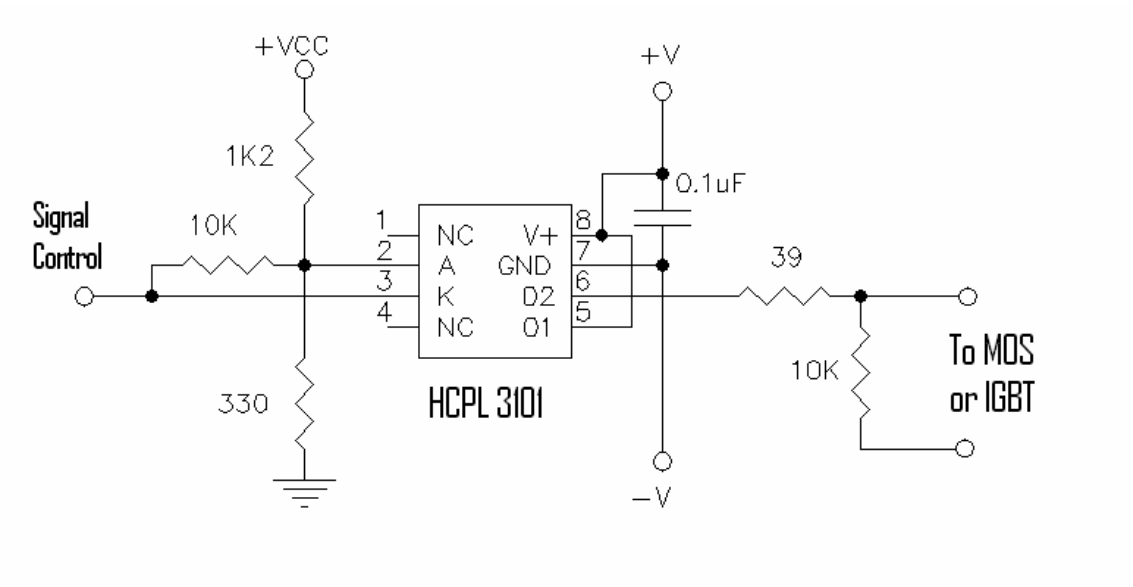


Figure 4-32. IGBT and MOSFET Gate Driver

4.11 Power Transfer Circuits (Bypass) Theory of Operation

Under normal conditions, the UPS inverter feeds all power to the load. However, in order to ensure that the load is supported in the event of a failure, or during maintenance, the equipment must also allow for direct connection to the AC main line. A switching relay is connected as shown in Figure 4-33.

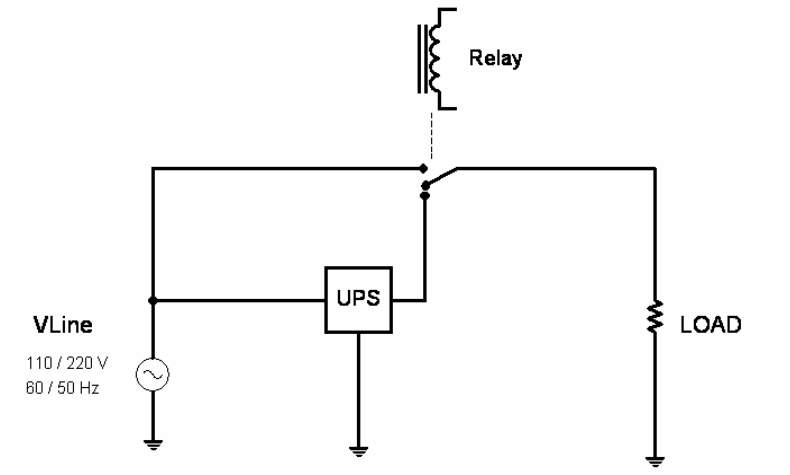


Figure 4-33. Bypass Relay Configuration

The relay's transfer time must be shorter than a period of the AC line. A transfer time of less than 20ms is fast enough to comply with this constraint. The transfer is allowed if phase and voltage conditions are satisfied, as explained in **Section 4.1.4**.

Figure 4-34 shows the circuit implemented to turn the bypass relay on and off using the BP_1 signal. Please note the three different grounds used in the system.

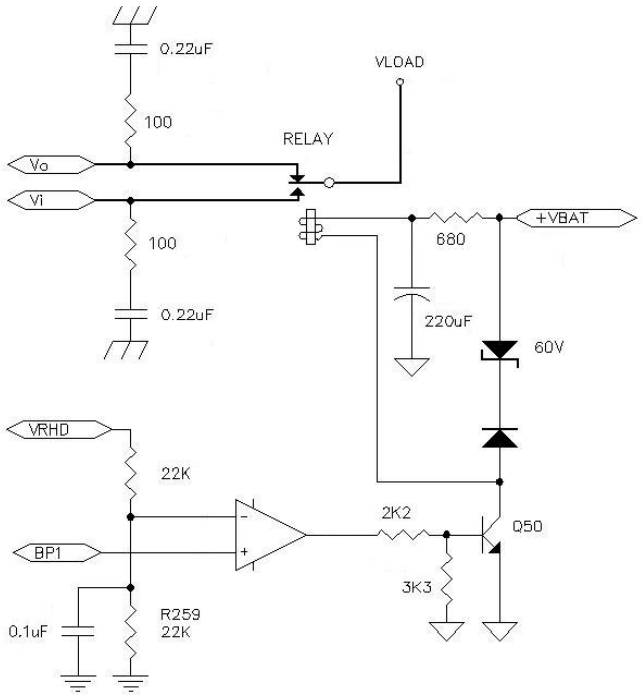


Figure 4-34. Relay Driver

4.12 Overcurrent Protection

Figure 4-35 shows the rectifier and inverter current sensing circuit. **Figure 4-36** shows the circuits implemented to sense the battery charger and battery booster currents. These circuits generate overcurrent signals when the current value reaches a defined level. These two signals are connected directly to the controller's FAULT 0 and FAULT 1 pins.

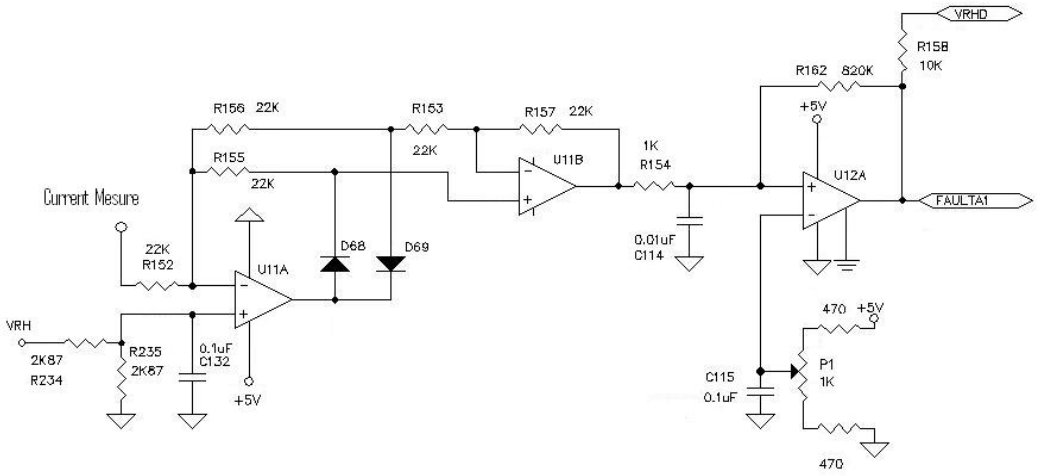


Figure 4-35. Overcurrent Protection for Rectifier and Inverter

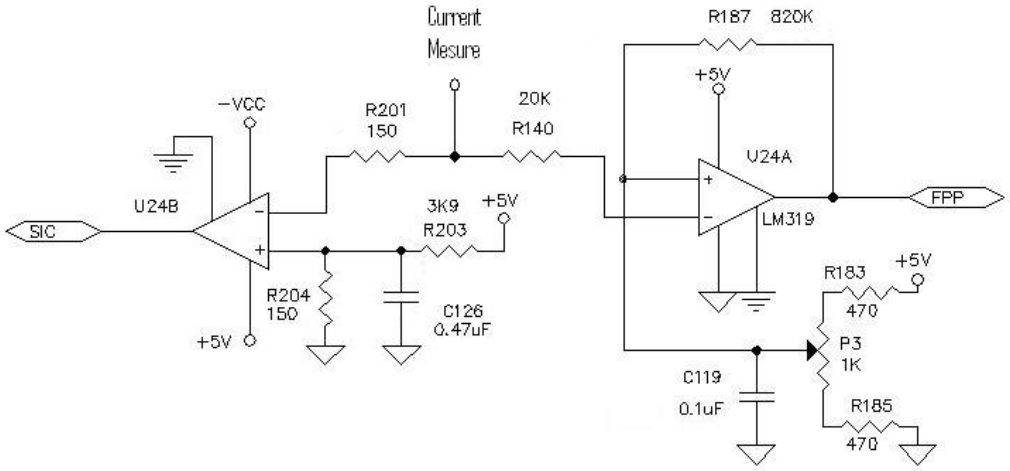


Figure 4-36. Overcurrent Protection for Charger and Push-Pull

The battery charger overcurrent protection generates a fault signal, named “SIC”, which turns off the charger PWM signal. Likewise, the battery booster’s overcurrent protection circuit generates the FPP signal, which turns off the drive signal for the MOSFETs.

4.13 Battery Temperature Sensing

The battery temperature sensing circuit is shown in **Figure 4-37**; it uses an LM-35-CZ, which is a precision centigrade temperature sensor. The output of this circuit is a voltage that is proportional to the temperature. This sensor must be located near the battery.

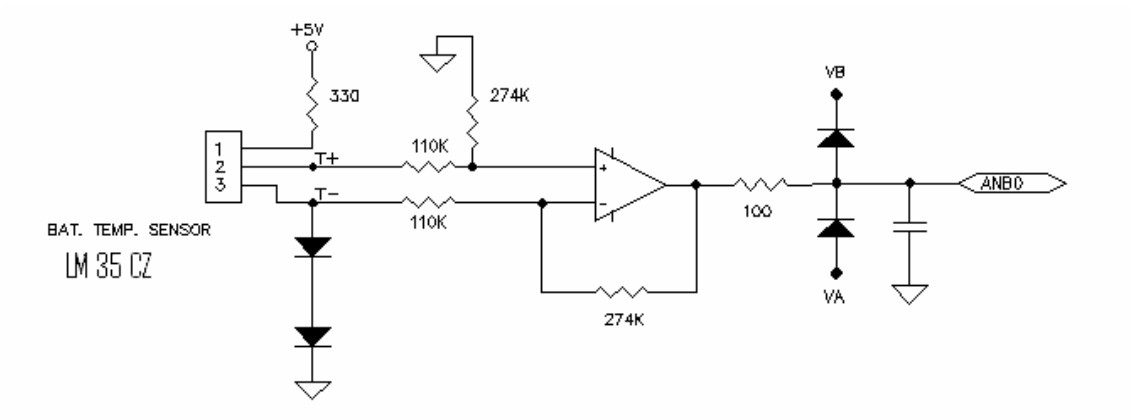


Figure 4-37. Battery Temperature Sensing Circuitry

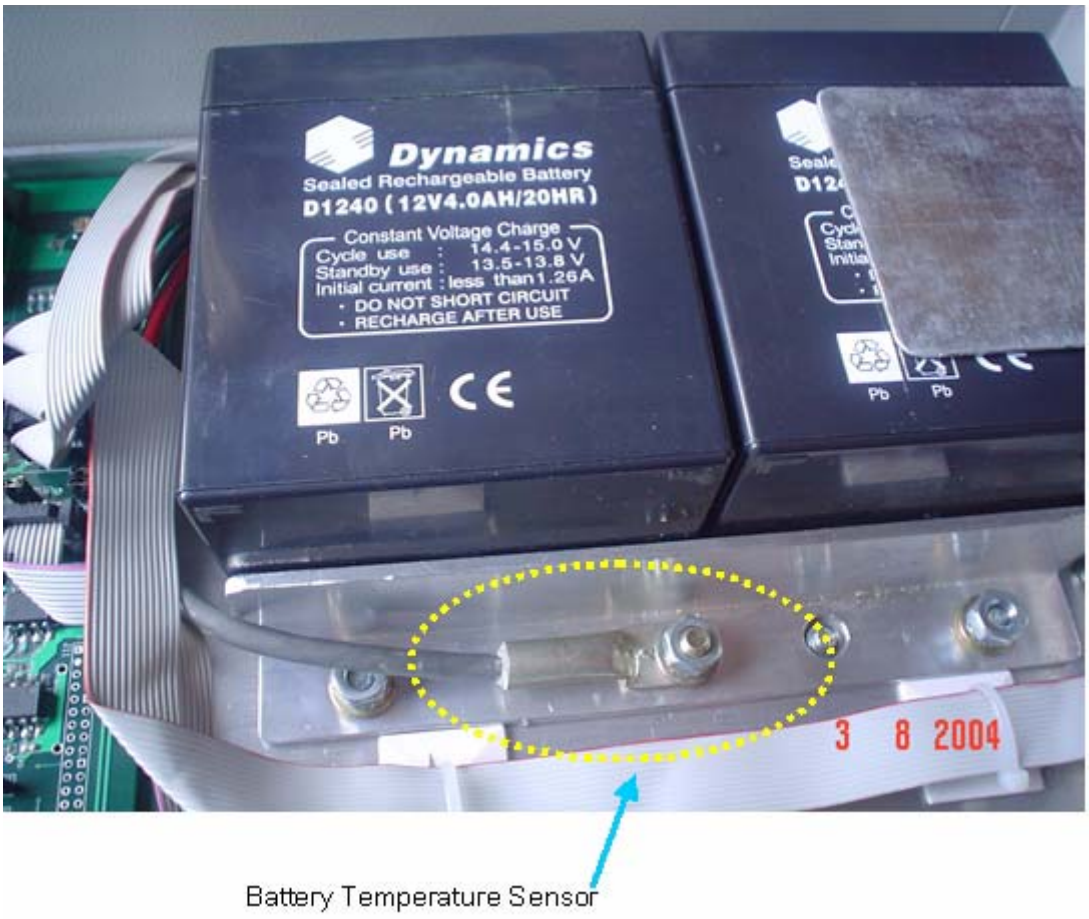


Figure 4-38. Battery Temperature Sensor

5. Control Loops In The Online UPS

The UPS' control algorithms are digitally implemented. The topology chosen for the compensators is the PID (Proportional-Integral-Derivative) and the PI (Proportional-Integral), with a 3-bit resolution. All gain constants are 16-bit implemented. The accumulators are all 32-bit, unless otherwise specified.

5.1 Control Algorithms Discrete Equivalents

An ideal PID analog controller is expressed as follows:

$$Gc(s) = K_p\left(1 + \frac{1}{\tau_i \cdot s} + \tau_d \cdot s\right)$$

An appropriate technique to discretize this transfer function is the backward difference method, which is based on numerical integration theory and has the following equivalence rule:

$$s \rightarrow \frac{d}{dt} \rightarrow \frac{(1-z^{-1})}{T_s}$$

where:
T_S is the sampling period

The PID transfer function in discrete time domain is:

$$G_c(z) = K_p + \frac{K_p \cdot T_s}{\tau_i \cdot (1-z^{-1})} + \frac{K_p \cdot \tau_d \cdot (1-z^{-1})}{T_s}$$

It is implemented as follows:

$$C(k) = K_P \cdot e(k) + K_I \cdot \sum_{j=0}^{k} e(j) + K_D \cdot [e(k) - e(k-1)]$$

where:

$$K_I = \frac{K_p \cdot T_s}{\tau_i}$$

is the discrete time integrator gain

and

$$K_D = \frac{K_p \cdot \tau_d}{T_s}$$

is the discrete time differential gain.

The proportional gain constant, K_P, remains unmodified.

5.2 PFC and Rail Control

Figure 5-1 shows a simplified PFC control loop, valid only for positive AC line voltages. The PFC is a current loop with a set point calculated as the product of the rail voltage control output times the AC line voltage.

The voltage control must keep a constant DC rail. *V. Positive Rail* is sensed and compared to *Voltage Setpoint.* The error signal then passes through a PI control network, which outputs one of the operands used to calculate the current set point. The other operand is the AC Line Voltage, V_{LINE}. This signal is then used as the set point of a second PI control, which forms the current control loop.

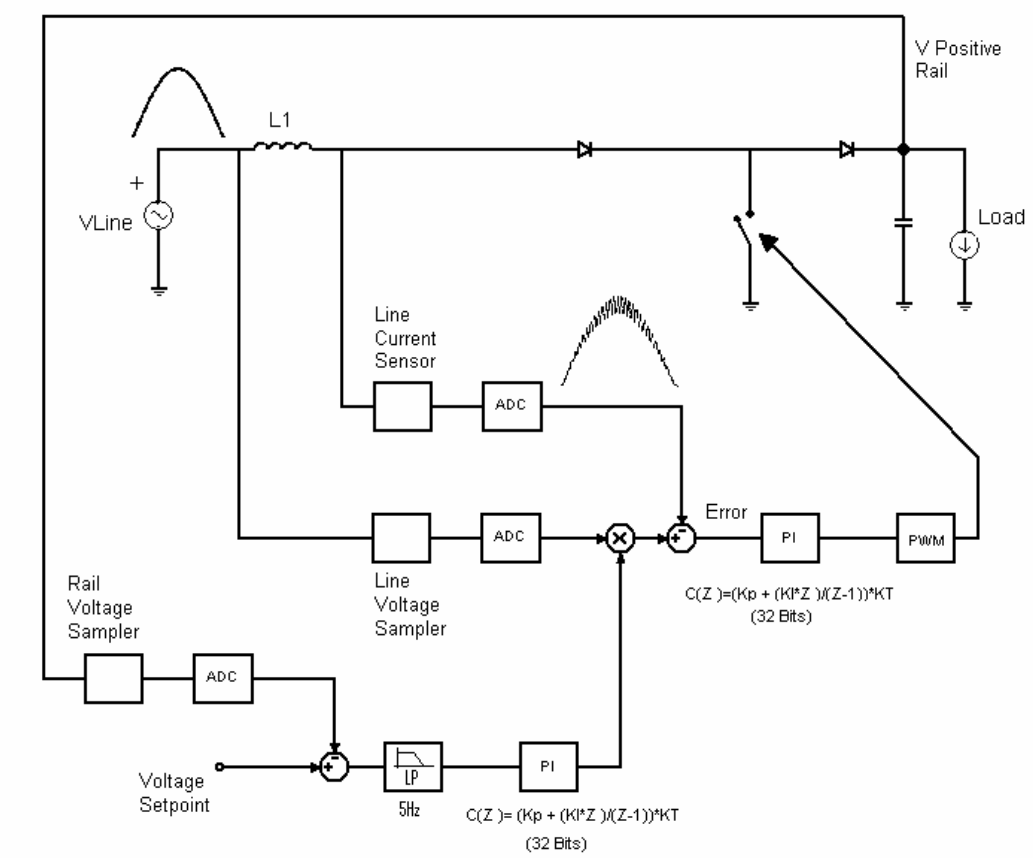


Figure 5-1. PFC and Rail Control Loops (Positive Semicycle)

The goal of the rectifier stage is to maintain the rail DC voltage at ± 220V and to control the input current to mimic the voltage waveform (and thus to make the complete system appear as a resistive load to the AC main line). In order to achieve these two goals, two control loops are required:

A **current control loop**, implemented with a PI controller, generates a current as required by the input inductor. The circuit symmetry makes it possible to implement a control using the line voltage absolute value to control the signal, regardless of its sign. A suitable controller for this application is a PI compensator with the following parameters:

$$Gc(s) = 6.4\left(1 + \frac{1}{0.0002 \cdot s}\right)$$

Using the backward difference method yields the following discrete time equivalent:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j)$$

with :
$K_P = 6.4$
and
$K_I = 1.6$

The second controller keeps the rail voltage constant and uses a low pass filter to reject the 60Hz and 120Hz ripple. The process is much slower than any of these frequencies, and therefore should not attempt to correct higher harmonic disturbances. The chosen controller is a PI. The output of this controller modulates the AC main line voltage to obtain the reference for the current loop.

The low-pass filter is a first-order Infinite Impulse Response (IIR) filter with a 3dB cutoff frequency of 5Hz, implemented with the following transfer function:

$$\frac{f(z)}{e(z)} = \frac{b_0 \cdot z}{z - a_1}$$

with:
$b_0 = 0.01$
$a_1 = 0.9984$

The following is the proper PI controller:

$$Gc(s) = 0.9\left(1 + \frac{1}{0.0563 \cdot s}\right)$$

Using the backward difference method yields the following discrete time equivalent:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j)$$

with:
$K_P = 0.9$
and
$K_I = 0.0008$

5.3 Battery Charger Control

The circuit in **Figure 5-2** shows a schematic battery charger control loop, consisting of an inner voltage loop and an outer current loop. The sensed battery voltage determines the charging current (limited to 1 Ampere). This current is a function of the PWM duty cycle.

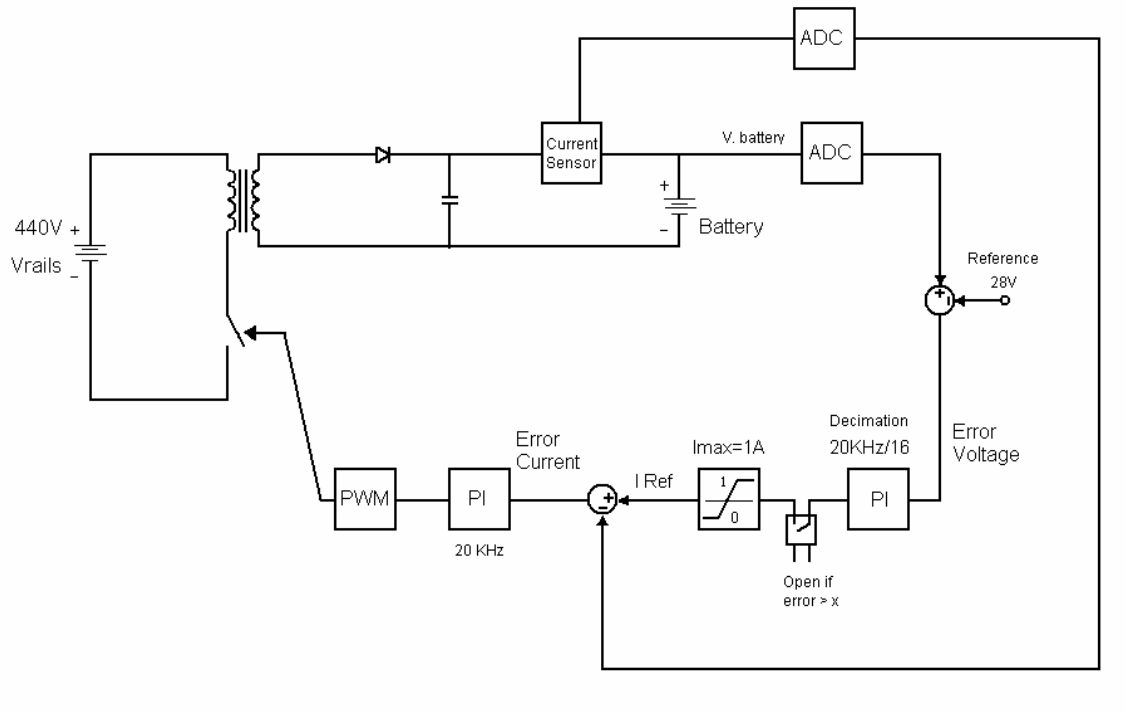


Figure 5-2. Battery Charger Control Loop

Given that the battery voltage is a slow signal, the sampling frequency for this control is decimated by 16 relative to the 20kHz (20kHz/16 = 1250Hz) sampling frequency used elsewhere in the system. The PWM switching frequency is preserved at 20kHz.

To avoid saturation of the voltage compensating network's integrator, its input is deactivated if the control output is above a predefined threshold.

The batteries are charged using the constant current–constant voltage approach. While the battery voltage is lower than the floating voltage (in this case, 28V), a constant current of 1A is applied. When the battery terminal voltage reaches 28V, the voltage is kept constant, decreasing the charge current. The battery charger system is implemented by two nested loops. The inner loop controls the charging current and uses a PI control. The reference for this control is delivered by a voltage control loop, whose output is limited from 0 to 1/3, where 1/3 (in Frac16 notation) represents 1A. When the batteries are discharged, the voltage control increases the current reference, looking for a 28V voltage. However, when the voltage control loop reaches 1/3, the integral action is disconnected and the output is limited, forcing the current loop to set a 1A charging current to the battery. When floating voltage is reached, the voltage control loop reduces its output, reducing the current applied to the batteries.

The controller chosen for the **current control loop** is a PI compensator with the following parameters:

$$\mathrm{Gc(s)} = 0.7\left(1 + \frac{1}{0.001 \cdot \mathrm{s}}\right)$$

Using the backward difference method results in the following discrete time equivalent system:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j)$$

with:
$K_P = 0.7$
and
$K_I = 0.035$

The controller chosen for the **voltage control loop** is a PI compensator with the following parameters:

$$Gc(s) = 0.2\left(1 + \frac{1}{32e - 3 \cdot s}\right)$$

Using the backward difference method results in the following discrete time equivalent system:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j)$$

with:
$K_P = 0.02$
and
$K_I = 0.0005$

5.4 Inverter Control

The objective of the inverter control loop is to supply the load with a voltage defined by the reference signal. This reference signal is generated by the PLL system, at a sampling frequency of 20kHz, high above the 50/60Hz nominal frequency. For this reason, it is reasonable to consider the set point a constant.

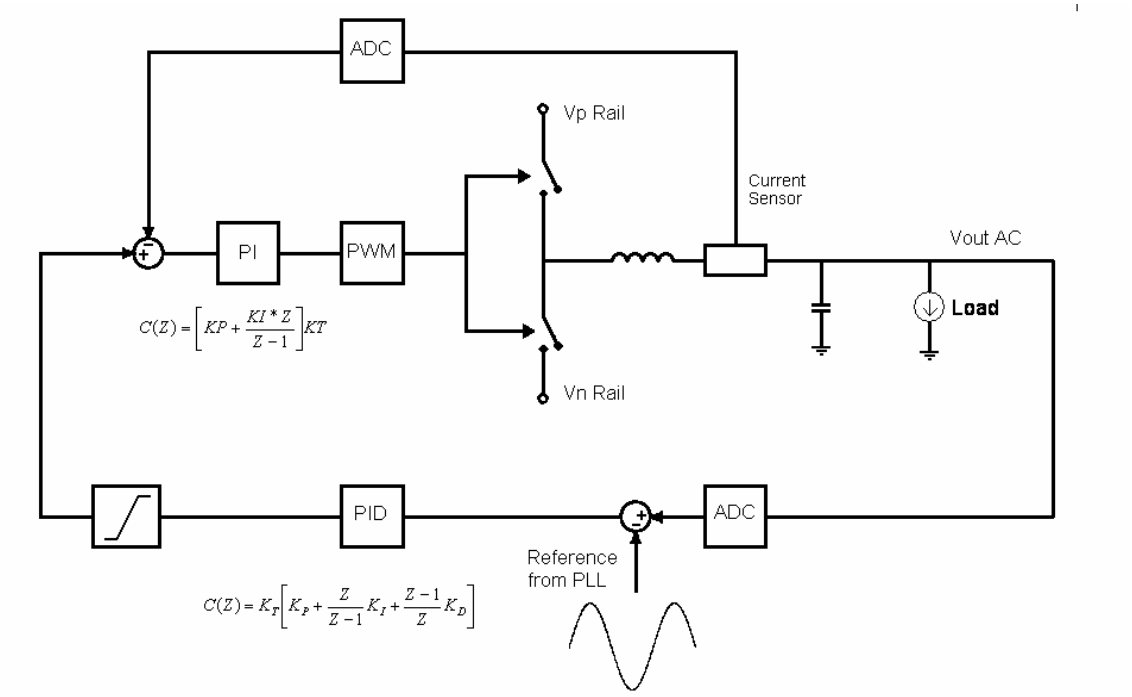


Figure 5-3. Inverter Control Loop

The inverter control is implemented with a nested topology. The outer loop controls the output voltage and the inner loop controls the inductor current. This configuration allows better stability of the feedback system and an implicit limitation of the current delivered to the load. The inner control loop stabilizes the system, acting as a damper for the LC output circuit.

The controller chosen for the current control loop is a PI compensator with the following parameters:

$$Gc(s) = 0.84\left(1 + \frac{1}{0.0017 \cdot s}\right)$$

Using the backward difference method results in the following discrete time equivalent system:

with:
$K_P = 0.84$
and
$K_I = 0.0024$

The outer loop is the voltage control. The goal of this loop is to keep a sinusoidal voltage waveform, regardless of the load characteristics. When nonlinear loads (as a full wave rectifier) are connected, a high current is drawn at the peaks of the signal. The action taken to overcome this condition is to inhibit the integral action by disconnecting the integrator input when the current draw is high. This action reduces the output distortion and enhances the controller response when the load requires high currents.

The controller chosen for the current control loop is a PI compensator with the following parameters:

$$\mathrm{Gc(s)} = 3.42\left(1 + \frac{1}{0.00036 \cdot \mathrm{s}} + 0.0001688 \cdot \mathrm{s}\right)$$

Using the backward difference method results in the following discrete time equivalent system:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j) + K_D \cdot [e(k) - e(k-1)]$$

with:
$K_P = 3.42$
$K_I = 0.475$
and
$K_D = 11.75$

5.5 Battery Booster Control Loop

The function of this controller is to keep the rail DC voltage at ±220V while the system is supported by the battery.

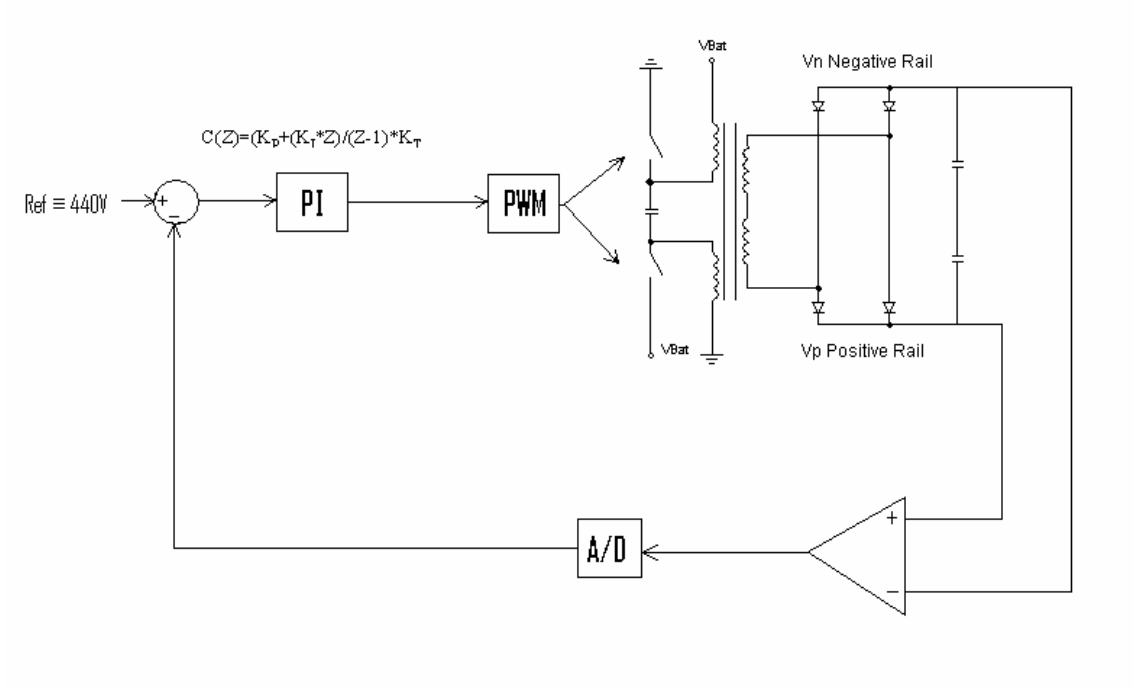


Figure 5-4. Battery Booster Control Loop

Given that the rail voltage is a slow signal, the sampling frequency for this control is decimated by 16 relative to the 20kHz (20kHz/16 = 1250Hz) sampling frequency used in other sections of the system. The PWM switching frequency is preserved at 20kHz.

The system is implemented with a PID compensator in order to regulate the rail DC voltage. The control output modulates the duty cycle of a push-pull power supply. The following are the parameters of such a controller:

$$\mathrm{Gc(s)} = 16\left(1 + \frac{1}{0.064 \cdot \mathrm{s}} + 0.000005 \cdot \mathrm{s}\right)$$

Using the backward difference method yields the following discrete time equivalent system:

$$C(k) = K_P \cdot e(k) + \sum_{j=0}^{k} K_I \cdot e(j) + K_D \cdot [e(k) - e(k-1)]$$

with:
$K_P = 16$
$K_I = 0.2$
and
$K_D = 0.1$

5.6 Minimizing Delay in the Control Loops

The triangular signal used as a carrier wave for the PWM is created from a counter that accumulates at the controller's clock speed (60MHz). The pulse width of the PWM is updated at a rate of 20kHz. This implies that in order to obtain both the 20kHz sampling frequency and a symmetric triangle wave, this counter must count from zero to 1500, then count back from 1500 to zero:

$$\frac{1}{2}\left(\frac{60MHz}{20KHz}\right) = 1500$$

Given that the PWM(s) update their value when the *PWM load* command is given to the PWM peripheral, the delay from the time elapsed between the sampling of the signal and the update of the PWM compare values should be minimized, enhancing the system's stability. The implementation of this delay and the relationship between the PWM reference and the ADC conversion is fully explained in **Section 7.**.

6. Control Board Design Considerations

6.1 56F8346 Controller

Two source voltages are needed: one for the V_{DD_IO} pins and the other for the ADC module. Jumper JG8 is provided to connect/disconnect the Temperature Sense Diode to the ADCA, Channel 7 (ANA7).

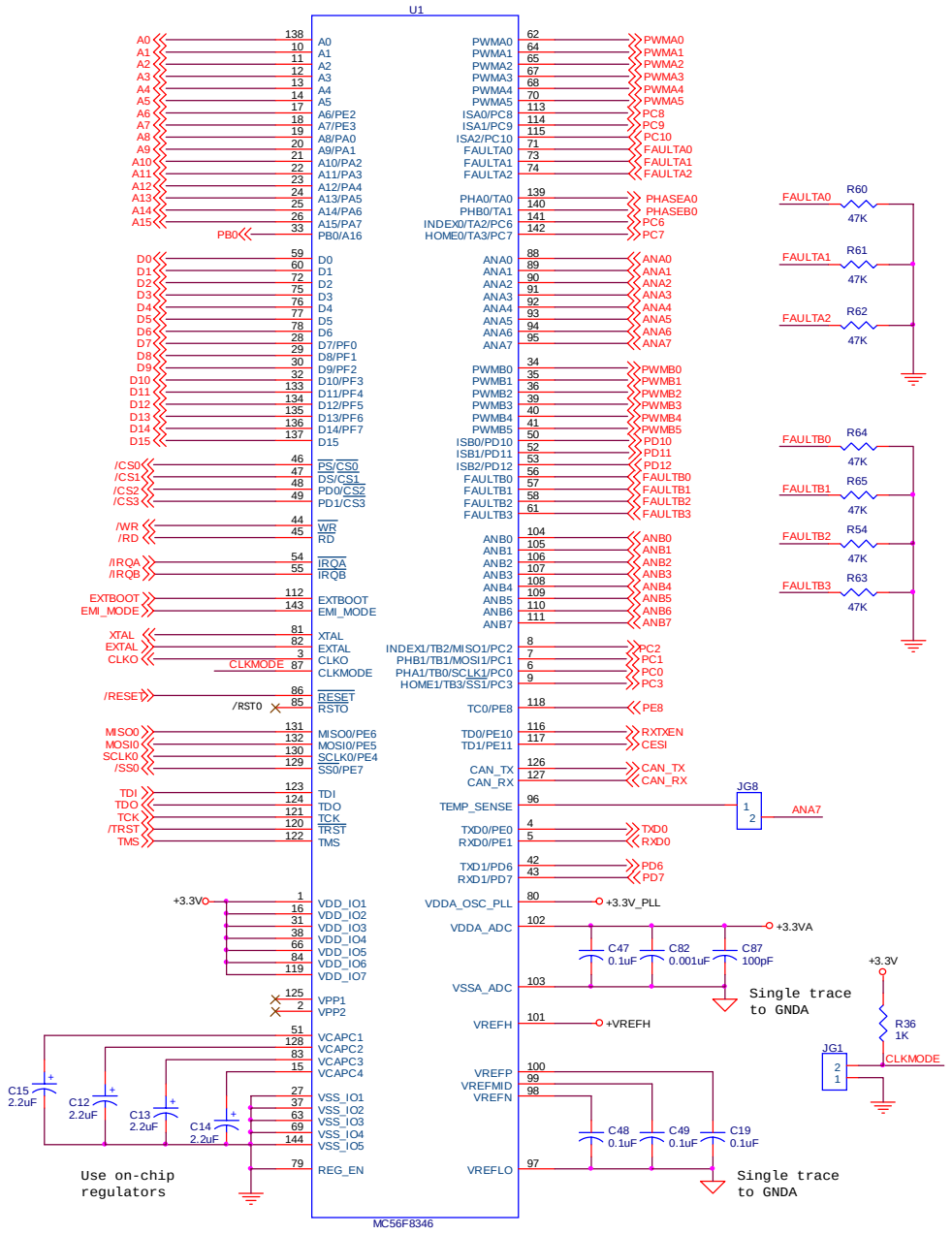


Figure 6-1. 56F8346 Controller

The Control Board provides a Clock Boot Mode jumper, JG1. This jumper is used to select the type of clock source being provided to the processor as it exits reset. The user can select between the use of a crystal or an oscillator as the clock source for the processor.

The fault inputs of the PWM modules are tied to ground as a pull-down configuration in order to prevent false fault conditions; see **Figure 6-2**.

6.2 Reset/Modes/Clock

Logic is provided on the Control Board to generate an internal Power-On Reset; see **Figure 6-2**. Additional reset logic is provided to support the reset signals from the JTAG connector, the Parallel JTAG Interface and the user reset push-button, S1; see **Figure 6-3**.

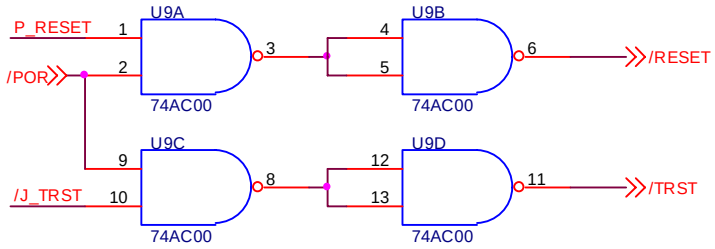


Figure 6-2. Reset Logic

The Control Board uses an 8.00MHz cystal (Y1) connected to processor's oscillator inputs (XTAL and EXTAL). The crystal configuration is shown in **Figure 6-3**.

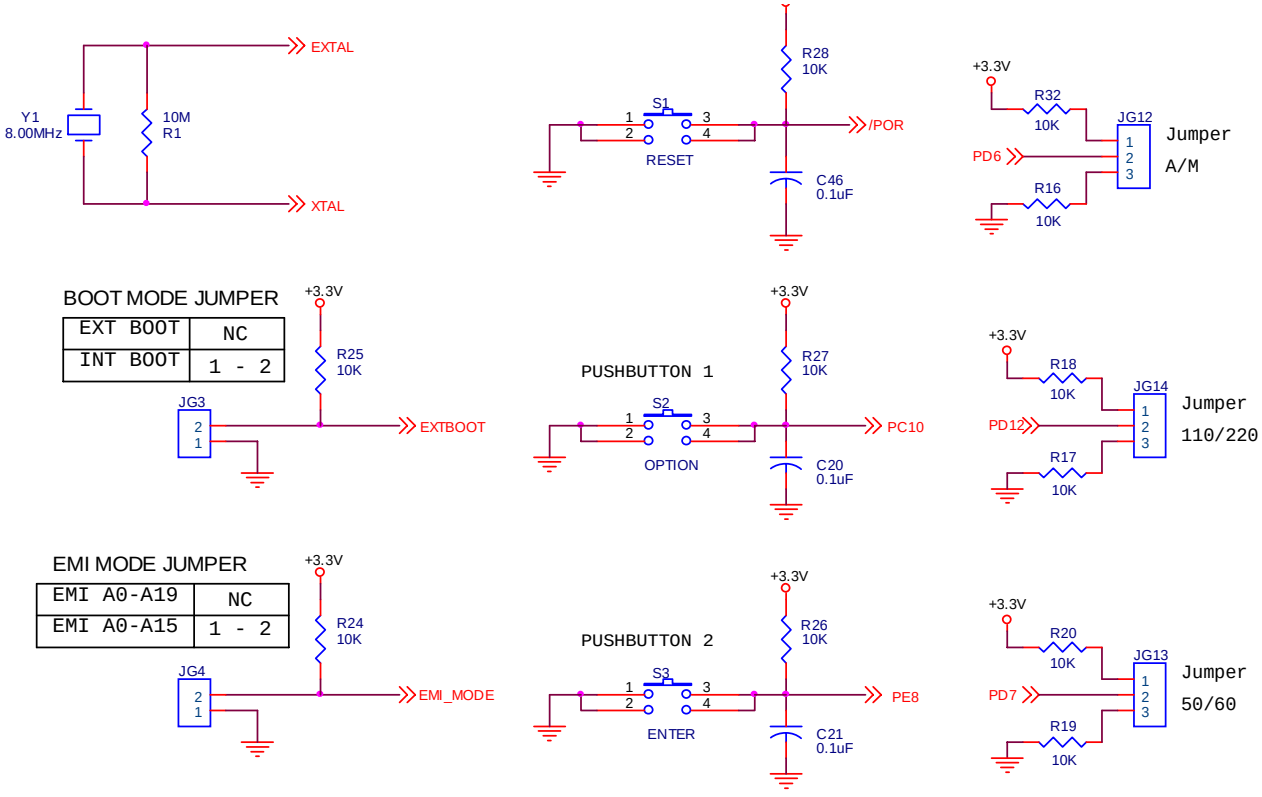


Figure 6-3. Selection Mode and Reset Button

The Control Board provides an External/Internal Boot mode jumper, JG4. This jumper is used to select the internal or external memory operation of the processor as it exits reset.

The Control Board also provides an EMI Boot Mode jumper, JG5. This jumper is used to select the External Memory Addressing Range Operating mode of the processor as it exits reset. The user can select either a 64K address space or an 8M address space.

As seen in **Figure 6-3**, the board includes two user pushbuttons, S2 and S3 (Option and Enter, respectively) that can be used for general purposes. The jumpers JG12, JG13, and JG14 are used to select Auto/Manual modes, 50/60Hz, and 110/220V, respectively.

6.3 Program And Data Memory

The Control Board contains two 128K x 16-bit Fast Static RAM banks. SRAM bank 0 is controlled by CS0 and SRAM bank 1 is controlled by CS1 and CS2; see **Figure 6-4**. This provides a total of 256K x 16 bits of external memory.

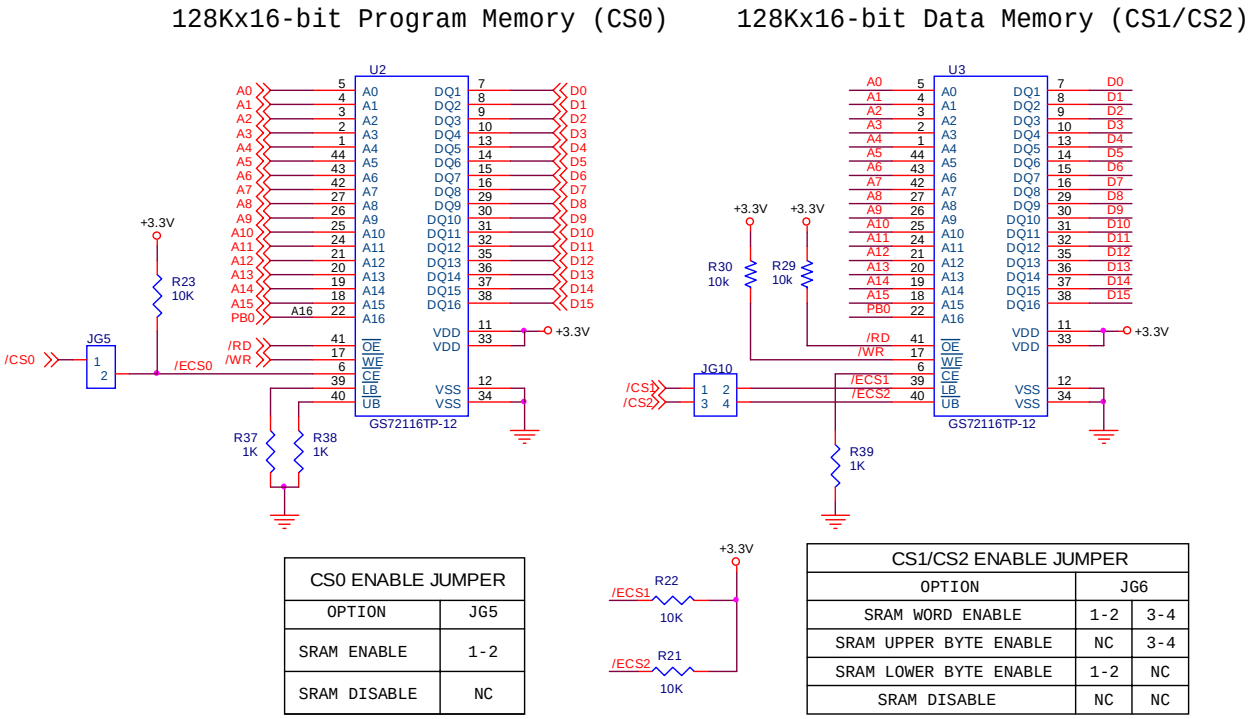


Figure 6-4. Program and Data Memory

6.4 RS-232 Serial Communications

The Control Board provides an isolated RS-232 interface by the use of an RS-232 level converter, Maxim MAX3245EEAI, designated as U4, and two high speed optocouplers, Fairchild 6N136S, designated as U20 and U21.

This circuitry transitions the SCI UART's +3.3V signal levels to RS-232-compatible signal levels and connects to the host's serial port via connector P2.

Flow control is not provided. The SCI0 port signals can be isolated from the RS-232 level converter by removing the jumpers in JG11. The RS-232 level converter/transceiver can be disabled by placing a jumper at JG7; see **Figure 6-5**.

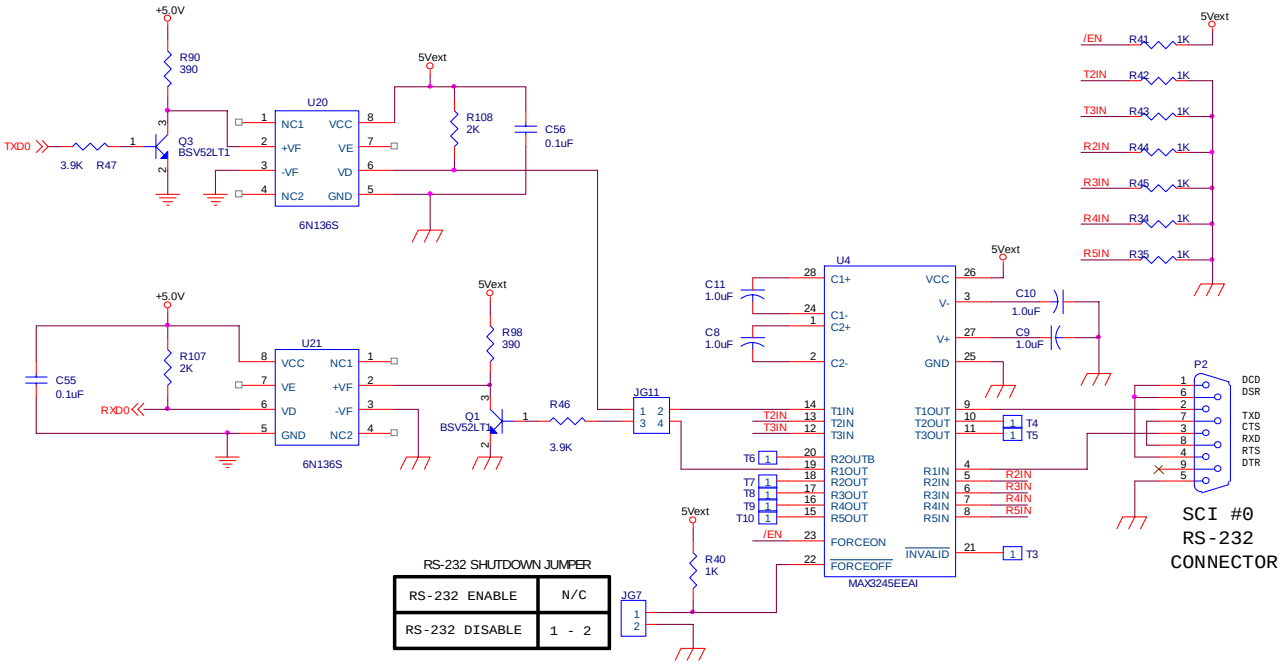


Figure 6-5. RS-232 Serial Communications

6.5 LCD Interface

The Control Board contains an LCD as the primary user interface feedback. The LCD contains a built-in internal driver. The display is controlled by some of the 56F8346's GPIO pins. **Figure 6-5** shows this hardware interface. As seen in **Figure 6-6**, the LCD interface (J12) uses a 4-bit data bus (D4-D7).

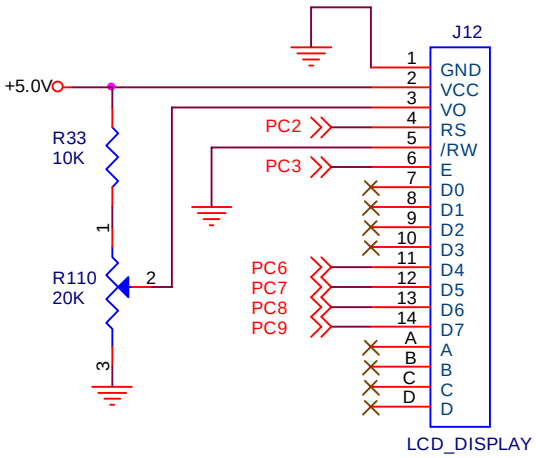


Figure 6-6. LCD Interface

6.6 Peripheral Expansion Connectors

6.6.1 Wireless Board Connector

The Control Board contains a connector, J11, intended for use by the 13192 RF Daughter Card (13192RFC). The MC13192 RF modem daughter card is a low-cost development board that provides a simple interface to Freescale's MC13192 transceiver.

The MC13192 is a short-range, low-power, 2.4GHz ISM band transceiver which contains a complete 802.15.4 physical layer (PHY) modem designed for the IEEE 802.15.4 wireless standard supporting star and mesh networking. For further information, please visit www.freescale.com/zigbee .

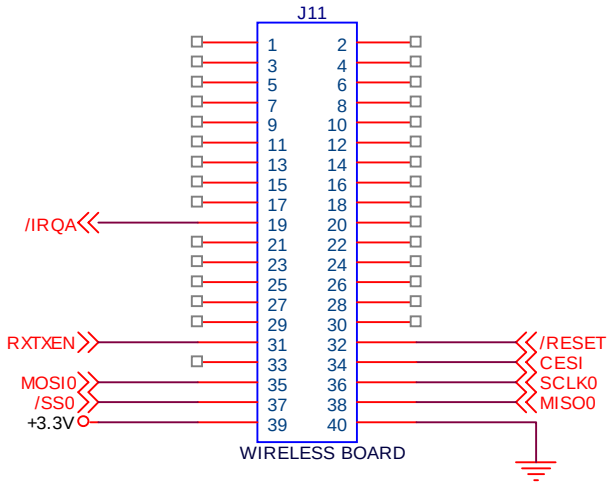


Figure 6-7. Wireless Board Connector

6.6.2 PWM Ports Expansion Connectors

The PWM ports A and B are attached to the J5 and J14 connectors, respectively.

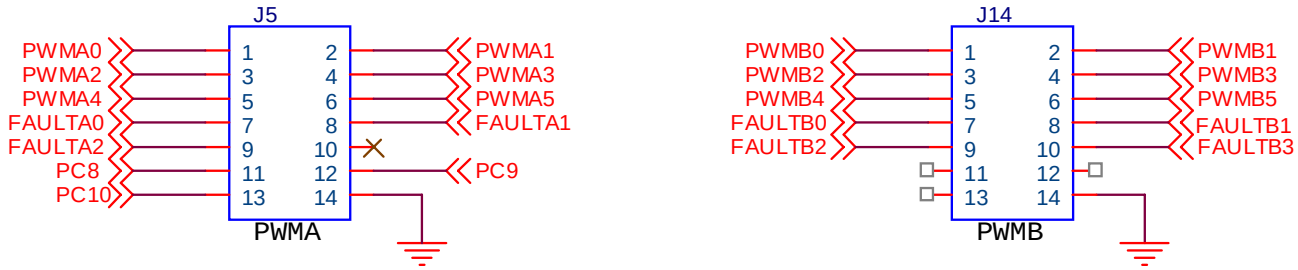


Figure 6-8. PWM Ports Expansion Connectors

6.6.3 A/D Ports Expansion Connectors

The 8-channel Analog-to-Digital conversion port A is attached to the J7 connector and port B is attached to J13; see **Figure 6-9**. There is an RC network on each of the Analog ports' input signals; see **Figure 6-17**.

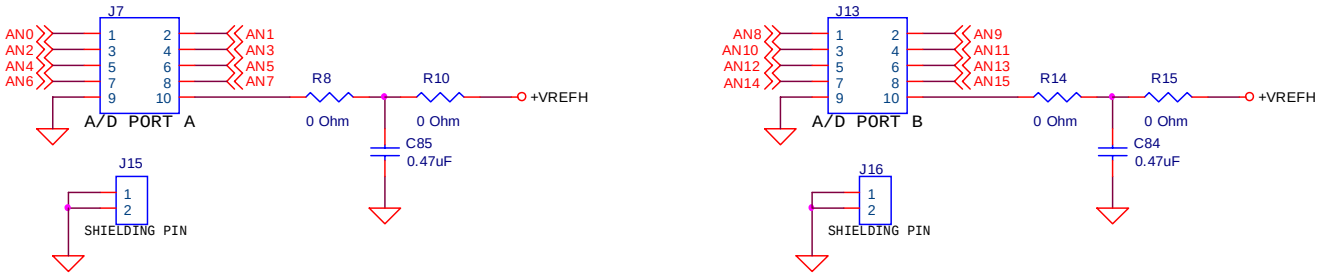


Figure 6-9. A/D Ports Expansion Connectors

6.6.4 Timer A Expansion Connector

The TA0 and TA1 signals of Timer Channel A port are attached to the J9 connector.

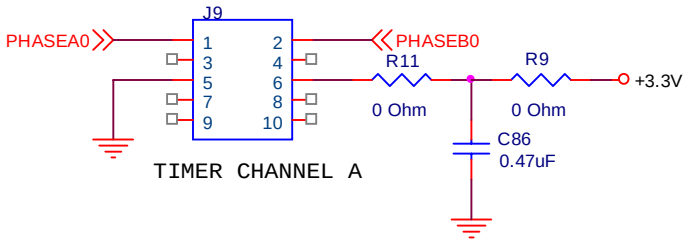


Figure 6-10. Timer A Expansion Connector

6.6.5 GPIOPort C Expansion Connector (Bits 0—1)

PC0 (GPIOC0) and PC1 (GPIOC1) pins are attached to the J8 connector.

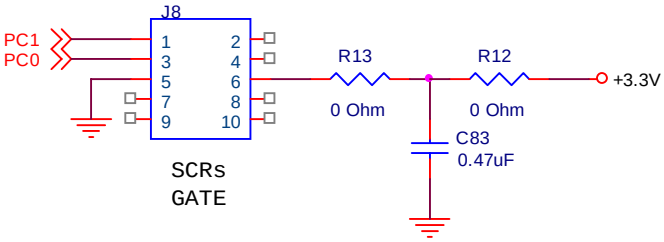


Figure 6-11. GPIO Port C Expansion Connector

6.6.6 GPIO Port D Expansion Connector (Bits 10—11)

PD10 (GPIOD10) and PD11 (GPIOD11) pins are attached to the J4 connector.

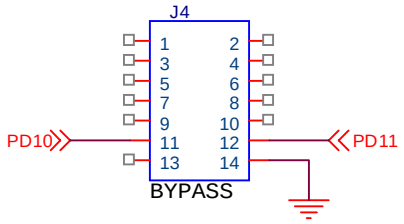


Figure 6-12. GPIO Port D Expansion Connector

6.7 Daughter Card Connector

The Control Board includes two daughter card connectors. One connector, J1, contains the processor's peripheral port signals. The second connector, J2, contains the processor's external memory bus signals. The Daughter Card connectors are used to connect the Ethernet evaluation daughter card; see **Figure 6-13**.

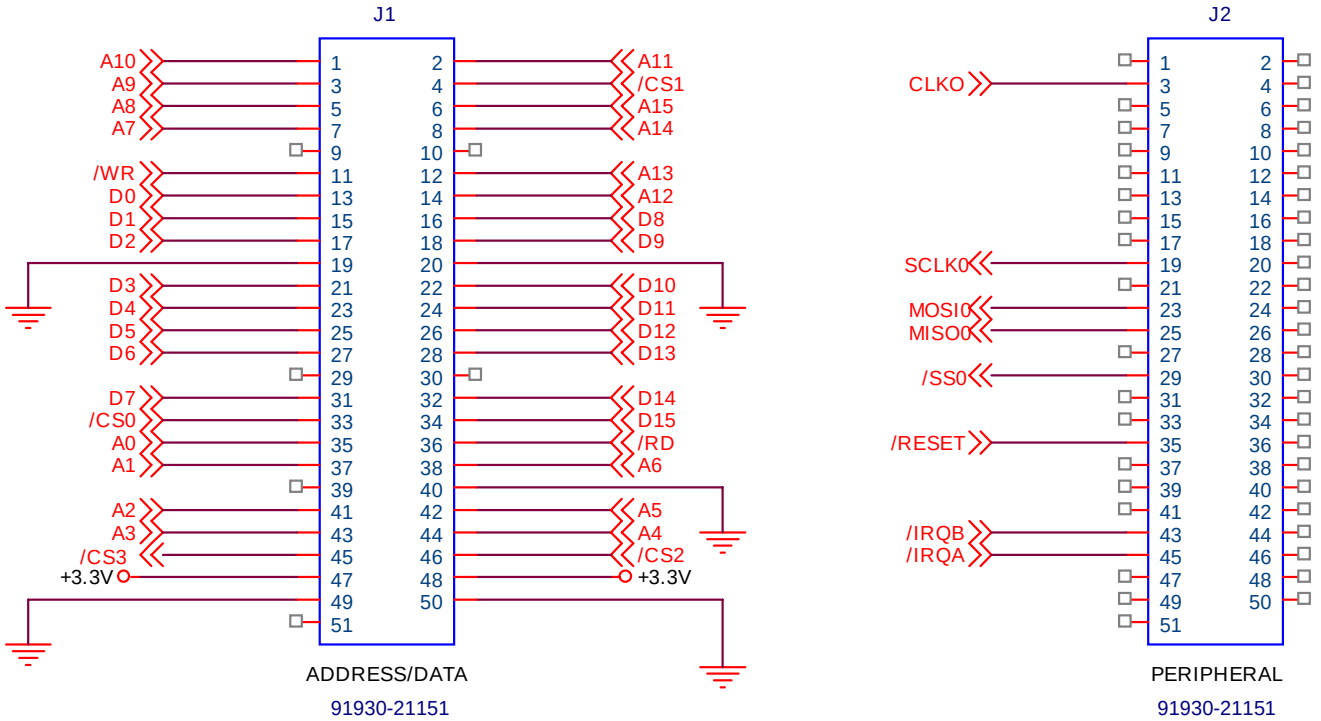


Figure 6-13. Daughter Card Connectors

6.8 CAN Interface

The Control Board contains a CAN physical-layer interface chip that is attached to the FlexCAN port's CAN_RX and CAN_TX pins through optocouplers. The Control Board uses a Philips high-speed, 1.0Mbps, physical-layer interface chip, PCA82C250.

The CANH and CANL signals pass through inductors before attaching to the CAN bus connectors. A primary, J6, and daisy-chain, J6, CAN connectors are provided to allow easy daisy-chaining of CAN devices. CAN bus termination of 120 Ohms can be provided by adding a jumper to JG2. The FlexCAN port is attached to J17 connector; see **Figure 6-14**.

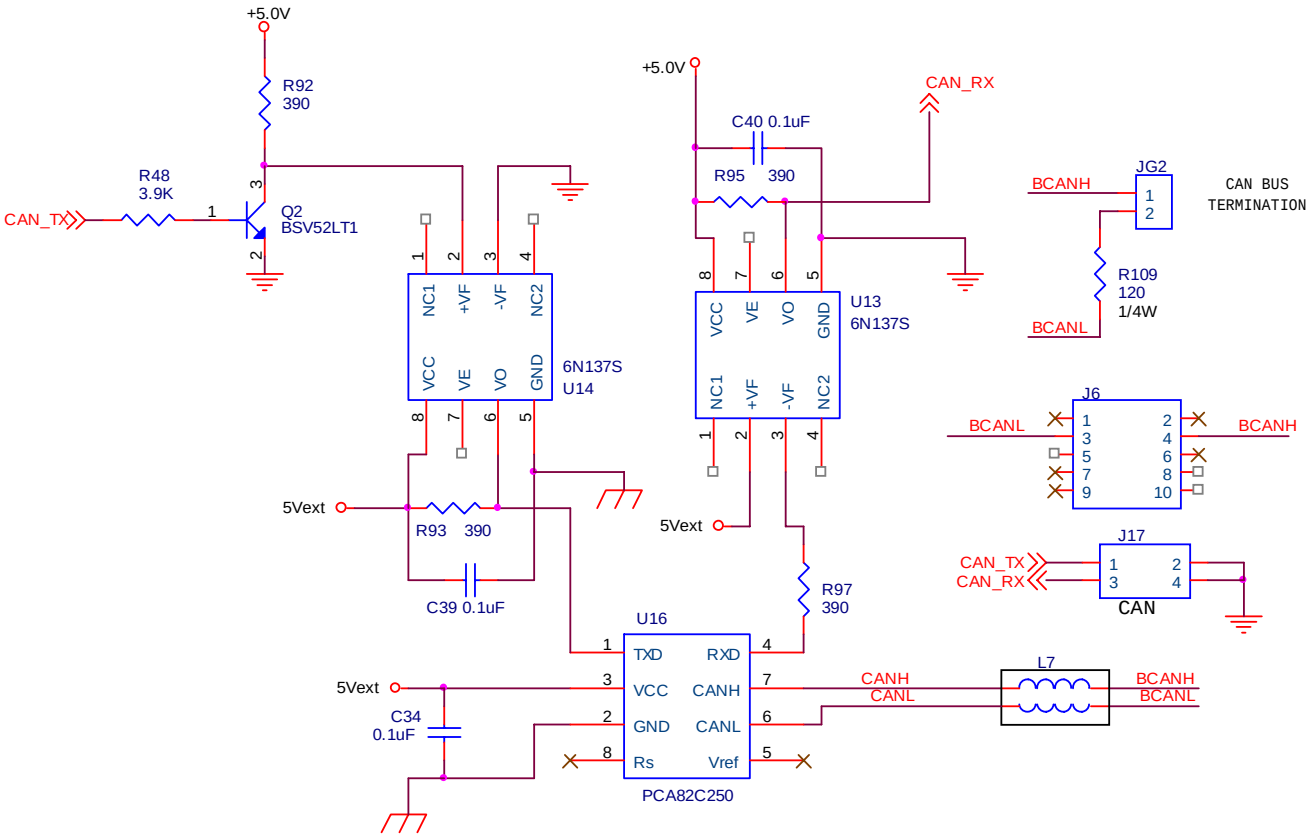


Figure 6-14. CAN Interface

6.9 Debug Support

The Control Board provides an on-board Parallel JTAG Host Target Interface and a JTAG interface connector for external target interface support. Two interface connectors are provided to support each of these debugging approaches. These two connectors are designated the JTAG connector and the Host Parallel Interface connector.

6.9.1 JTAG Connector

The JTAG connector on the Control Board allows the connection of an external Host Target Interface for downloading programs and working with the 56F8346's registers. This connector is used to communicate with an external Host Target Interface which passes information and data back and forth with a host processor running a debugger program.

When this connector is used with an external Host Target Interface, the parallel JTAG interface should be disabled by placing a jumper in jumper block JG9.

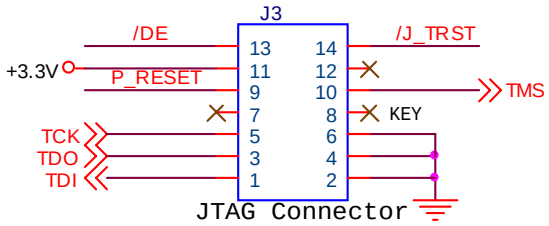


Figure 6-15. JTAG Connector

6.9.2 Parallel JTAG Interface Connector

The Parallel JTAG Interface connector, P1, allows the 56F8346 to communicate with a Parallel Printer Port on a Windows PC. All interface signals are optoisolated. Using this connector, the user can download programs and work with the 56F8346's registers. When using the parallel JTAG interface, the jumper at JG9 should be removed; see **Figure 6-16**.

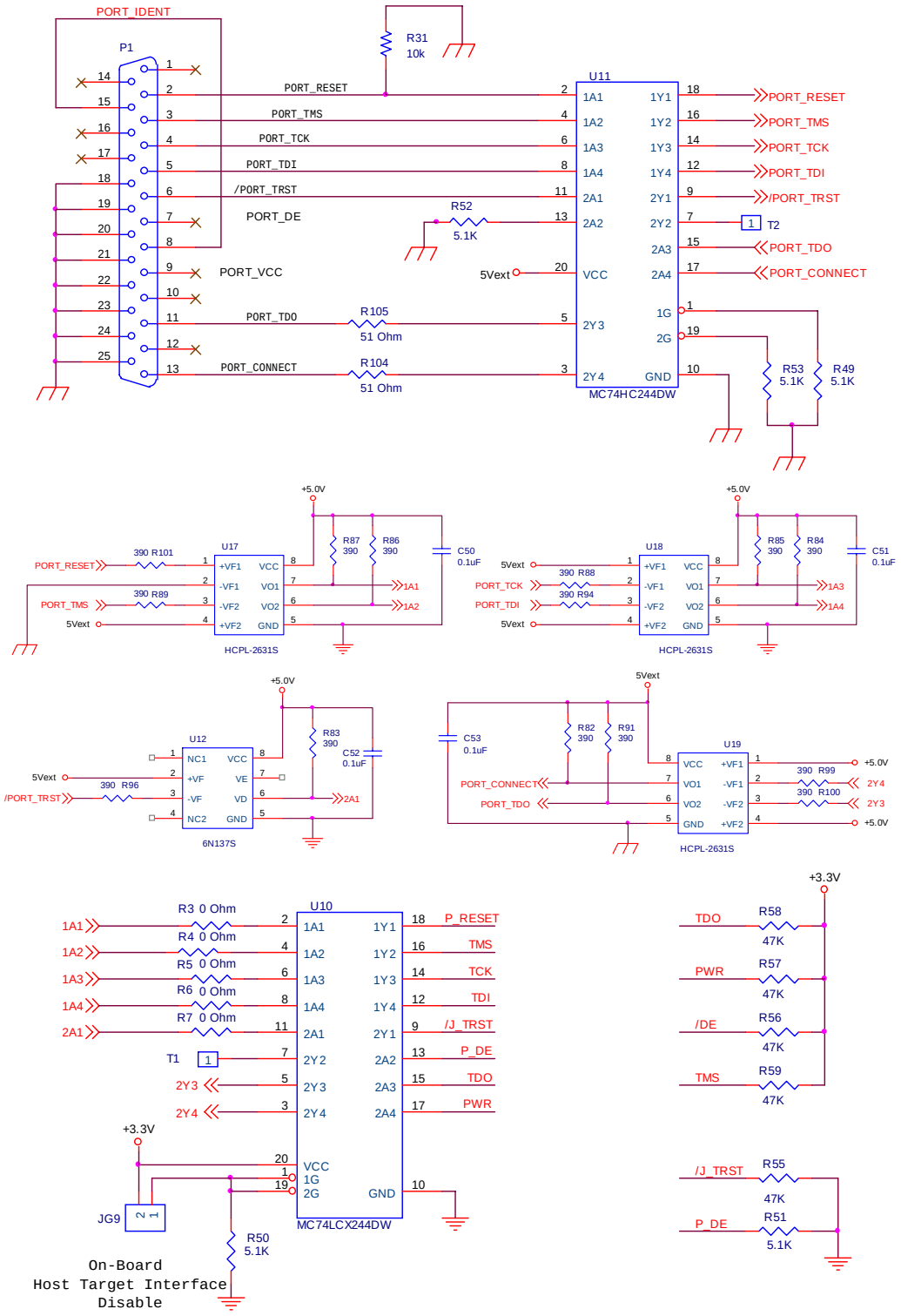


Figure 6-16. Parallel JTAG Interface Connector

6.10 A/D Filters

As **Figure 6-17** shows, all Analog-to-Digital ports of the processor are connected to RC networks that work as low-pass filters.

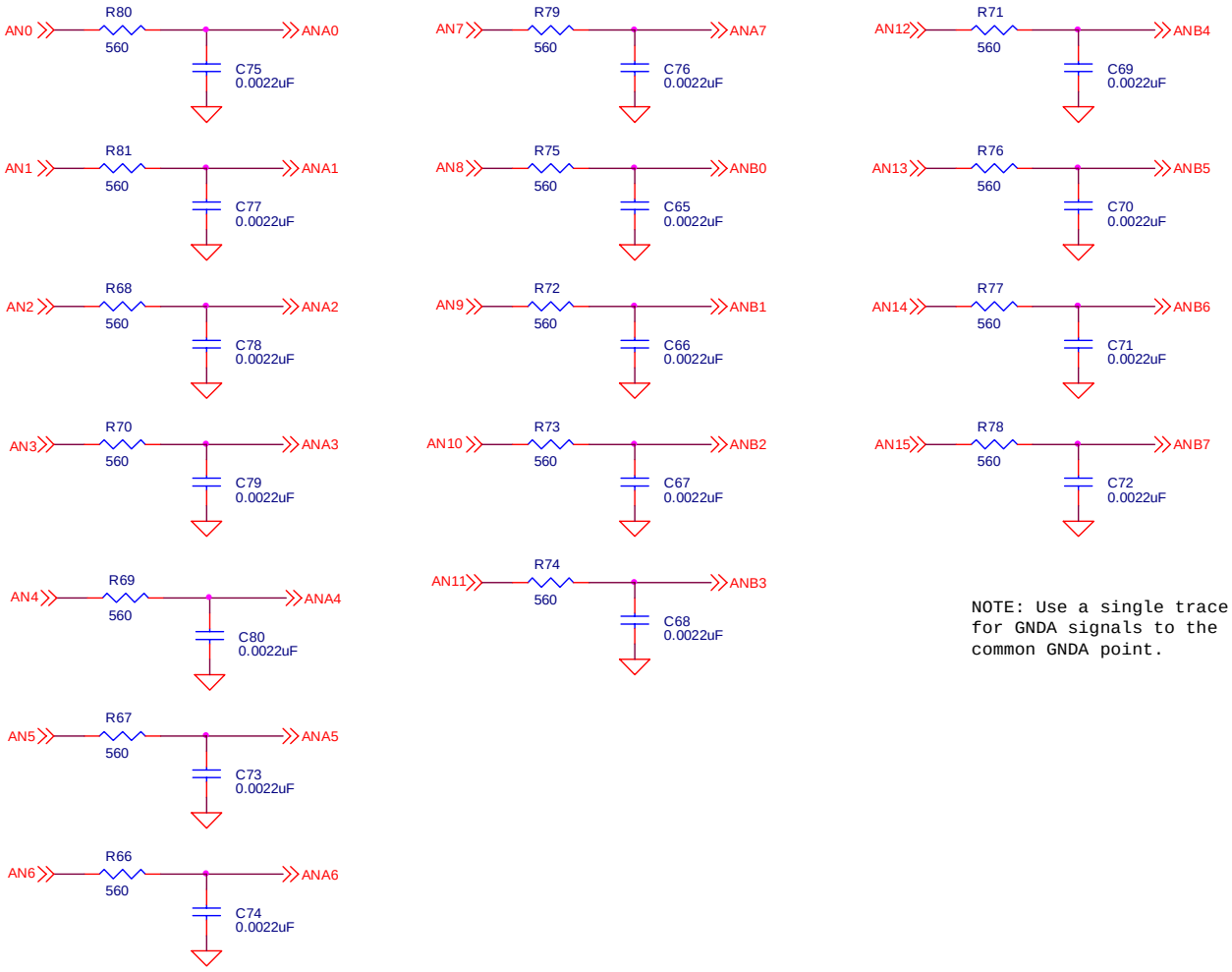


Figure 6-17. Passive Low-pass Filters of the A/D Ports

6.11 Power Supply

The Control Board has two power inputs, P3 and P4, through the 2.1mm coax power jacks.

The main power input (P3) must be fed with a voltage of +12V DC at 1.2A. This power input feeds a 5V DC voltage regulator (U6), which supplies two more 3.3V DC regulators. The first of the 3.3V DC regulators, U8, supplies the voltage to all of the 3.3V DC ICs. The second 3.3V DC regulator, U7, feeds the ADC module of the controller and feeds another 3.0V DC voltage regulator, U15, used as a reference for the controller's ADC; see **Figure 6-18**.

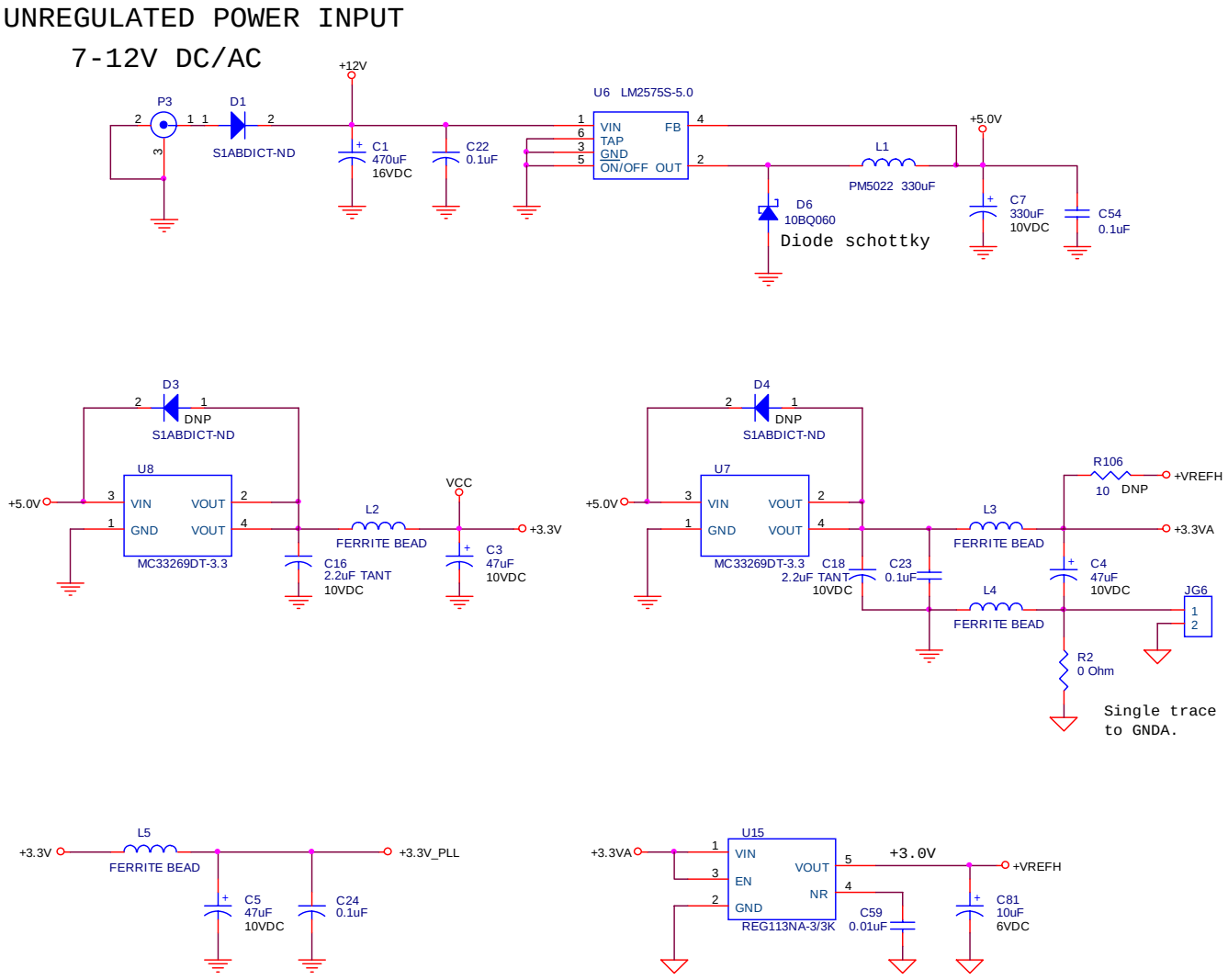


Figure 6-18. Power Input

The second power input (P4) feeds the +5V DC voltage regulator, U5. This regulator supplies the voltage to a part of the optocouplers and the components before them.

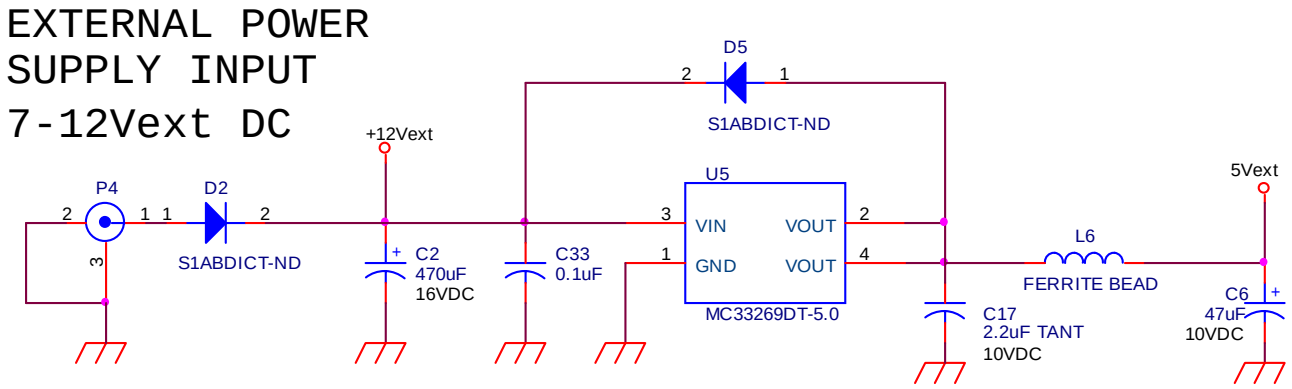


Figure 6-19. External Power Input

Several test points were placed over the Control Board and LEDs indicate when the power sources are turned on; see **Figure 6-20**.

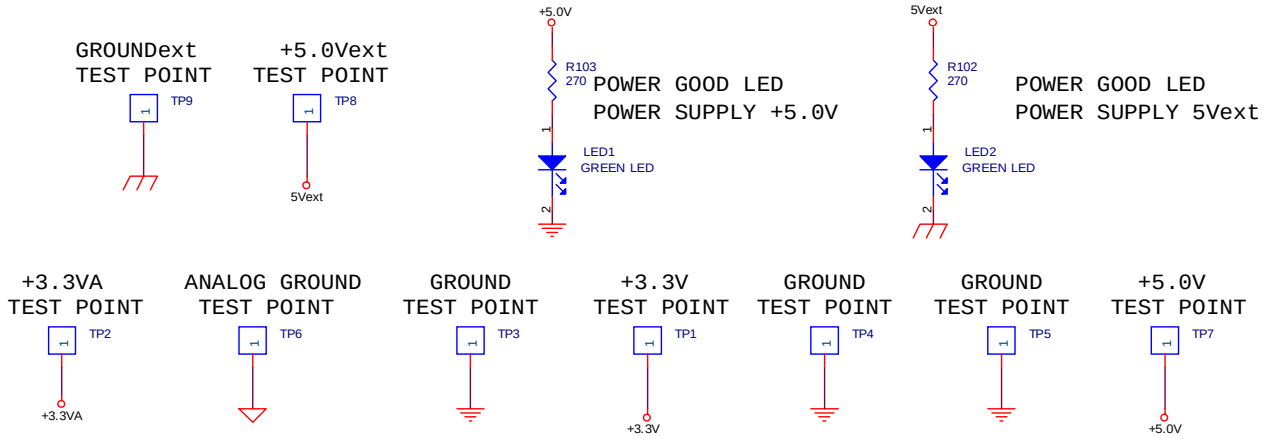


Figure 6-20. Power Supply LEDs and Test Points

7. Control Software Design Considerations

7.1 Peripheral and I/O Pins Assignment

Table 7-1. Assignment of the Analog-to-Digital Converters

Peripheral Input	Signal
ADCA 0	Inverter Output Voltage
ADCA 1	Inverter Inductor Current
ADCA 2	Line Input Voltage
ADCA 3	Line Input Current
ADCA 4	Battery Voltage
ADCA 5	Battery Current
ADCA 6	Rail Voltage Difference (V_P + V_N)
ADCA 7	UPS Load Current
ADCB 0	Battery Temperature Sensor

Table 7-2. Digital Outputs and Inputs

Peripheral Port	Function
PWMA 0 and 1	Inverter Switching Network – Output
PWMA 2 and 3	Battery Booster Switching Network – Output
PWMA 4 and 5	Power Factor Correction Switching – Output Network
Timer A0	Battery Charger PWM – Output
Timer A1	100kHz 35% Duty Cycle signal to auxiliary power supplies – Output
Timer C0	General purpose delay generator (start up and LCD)
Timer C2	Sets delay between PMW load and ADC start of conversion Synchronizes ADC to PWM
GPIOC0	Rectifier SCR Control – Output
GPIOC1	Inverter Enable – Input
GPIOD6	Auto/Manual Frequency Selection Jumper – Input
GPIOD7	50/60Hz Manual Frequency Selection Jumper – Input
GPIOD10	Bypass relay control – Output
GPIOD12	120 / 220V selector

Table 7-2. Digital Outputs and Inputs (Continued)

Peripheral Port	Function
PWMA - Fault 0	Inverter Overcurrent Protection; PWM Fault 0 – Input
PWMA – Fault 1	PFC Overcurrent Short Circuit Protection; PWM Fault 1 – Input
PWMA – Fault 2	Battery Booster Overcurrent Short Circuit Protection; PWM Fault 2 – Input

7.2 Main Execution Routine

All real-time operational software runs in the interrupt service routines. The main execution routine is dedicated to Monitor and Control functions. All real-time processing is accomplished by interrupt service routines.

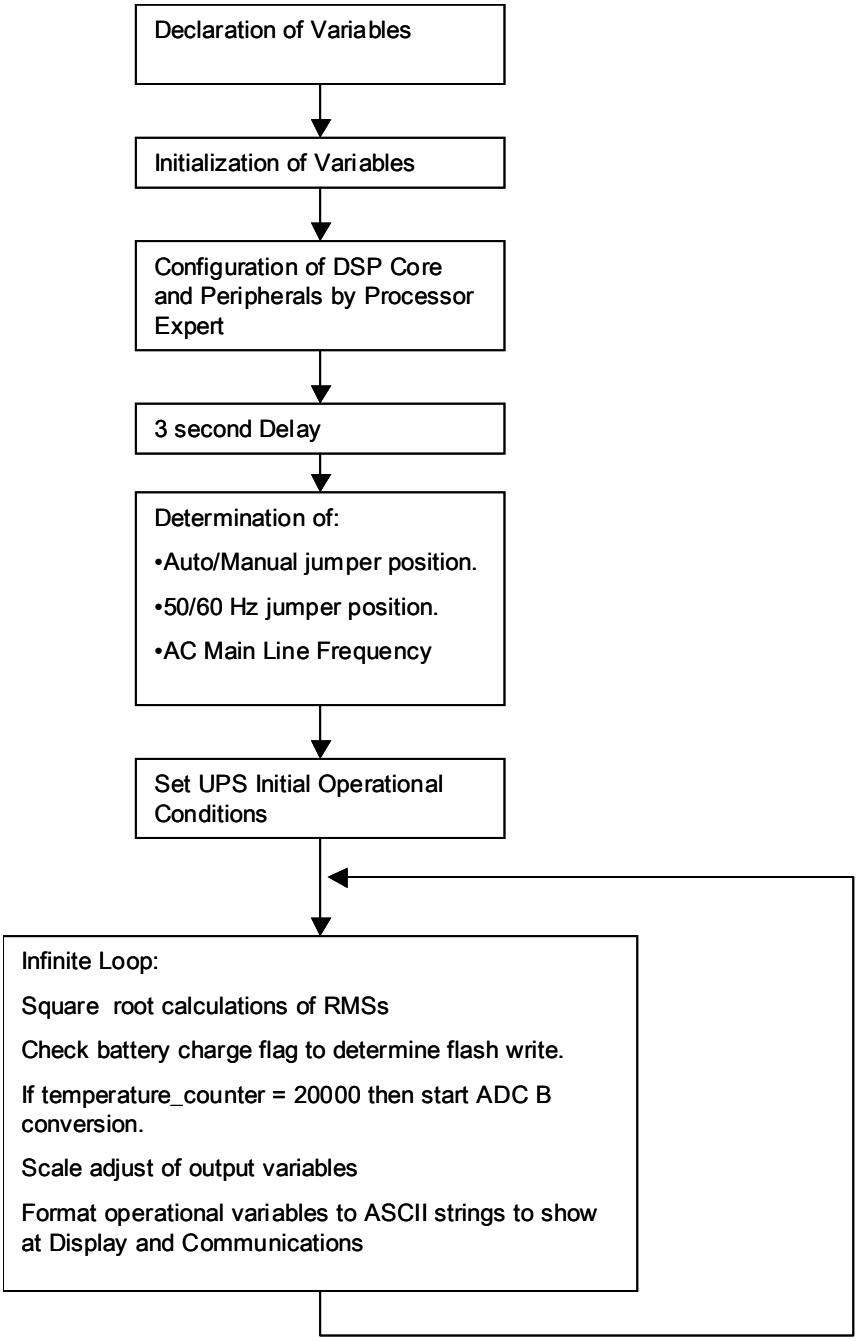


Figure 7-3. Main Routine Flow Diagram

7.3 Interrupt Handlers

The following interruptions perform the system's environment sensing:

- ADCA End Of Conversion
- ADC End Of Conversion
- PWM Fault 0
- PWM Fault 1
- PWM Fault 2
- Delay_Timer_OnInterrupt

7.3.1 ADC End of Conversion Interrupt Service Routine

All real-time controls are executed inside this routine. **Figure 7-4** through **Figure 7-8** show the flow of the program.

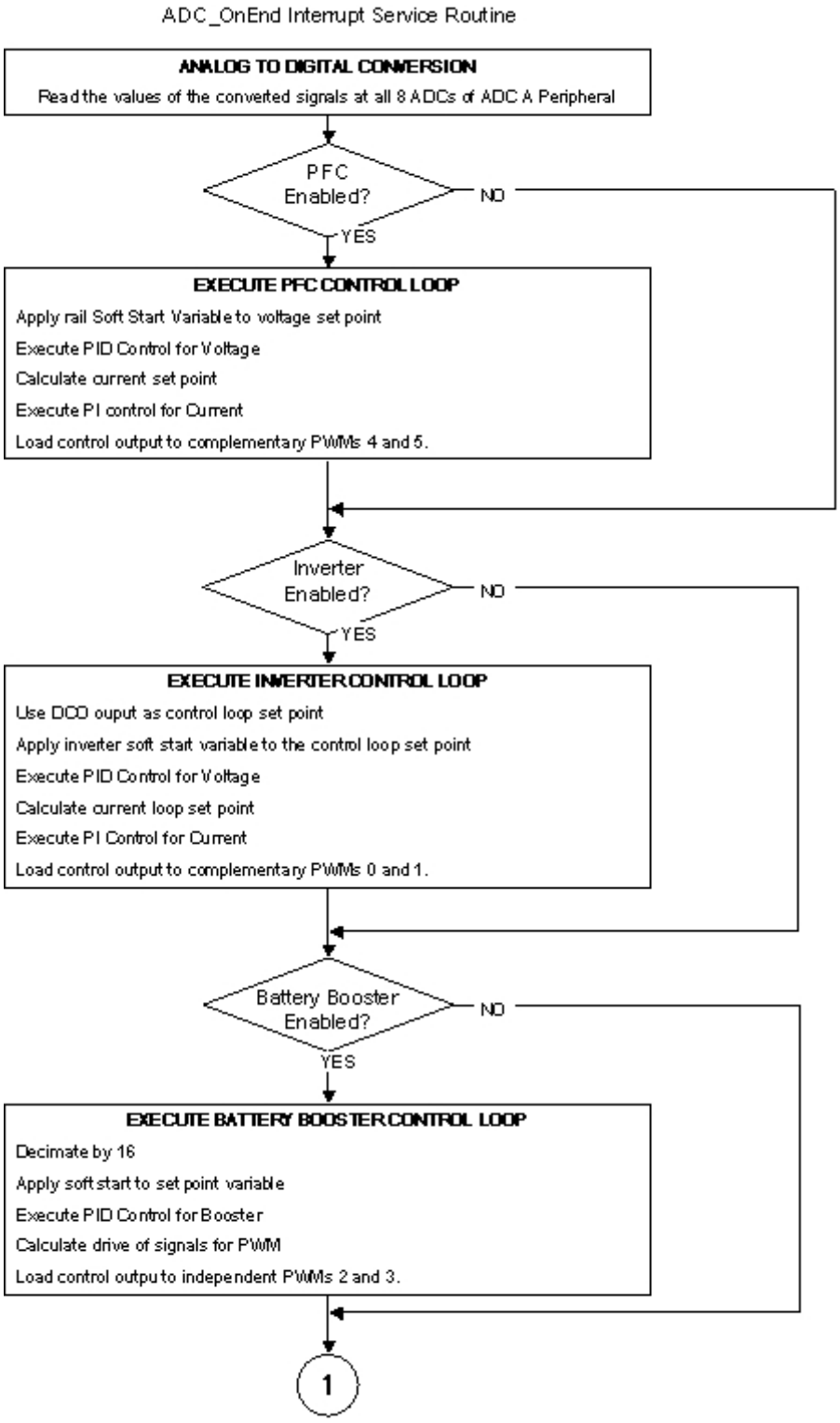


Figure 7-4. Flow Diagram for the Interrupt Service Routine *AD1_OnEnd* (1 of 5)

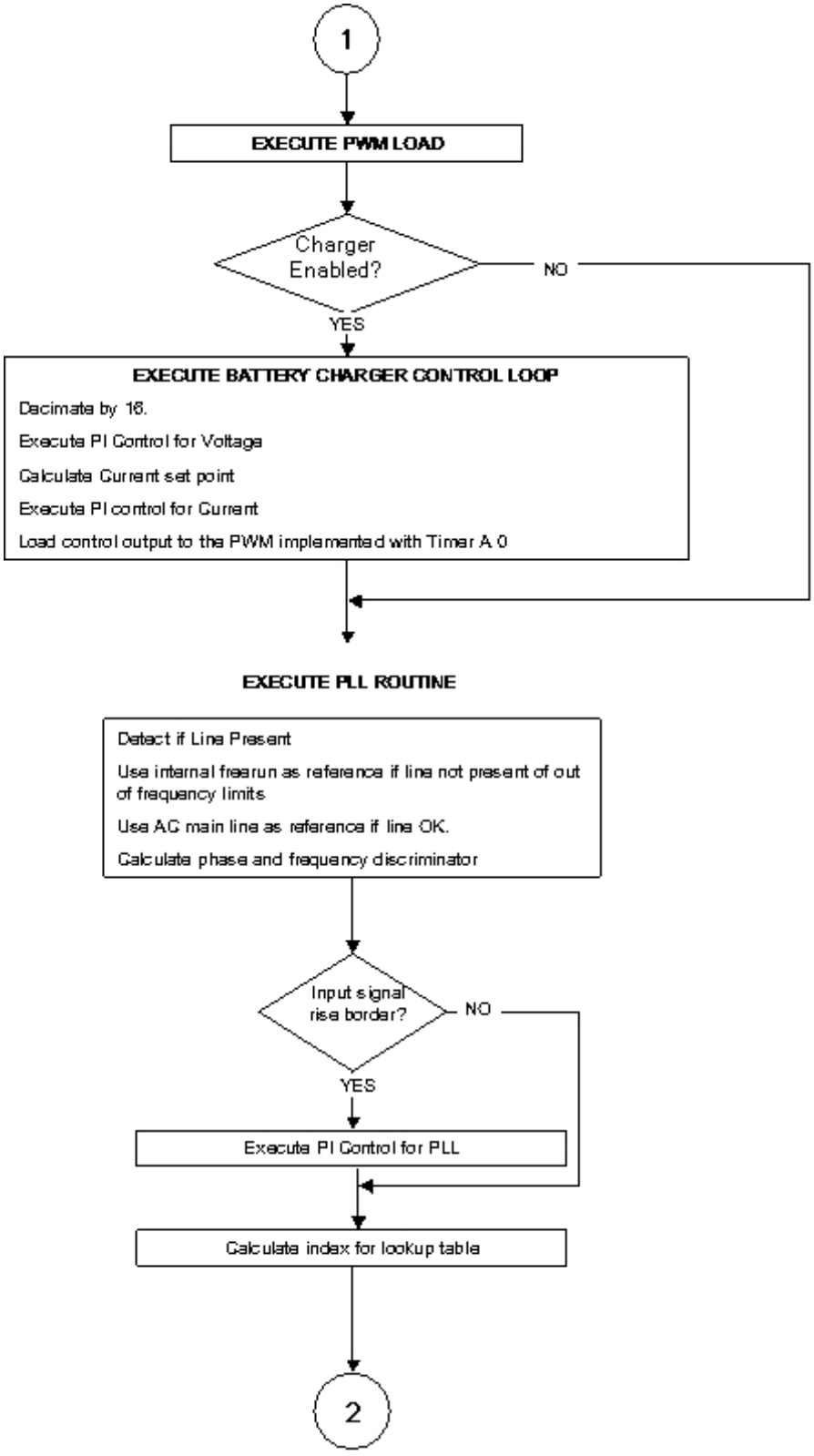


Figure 7-5. Flow Diagram for the Interrupt Service Routine *AD1_OnEnd* (2 of 5)

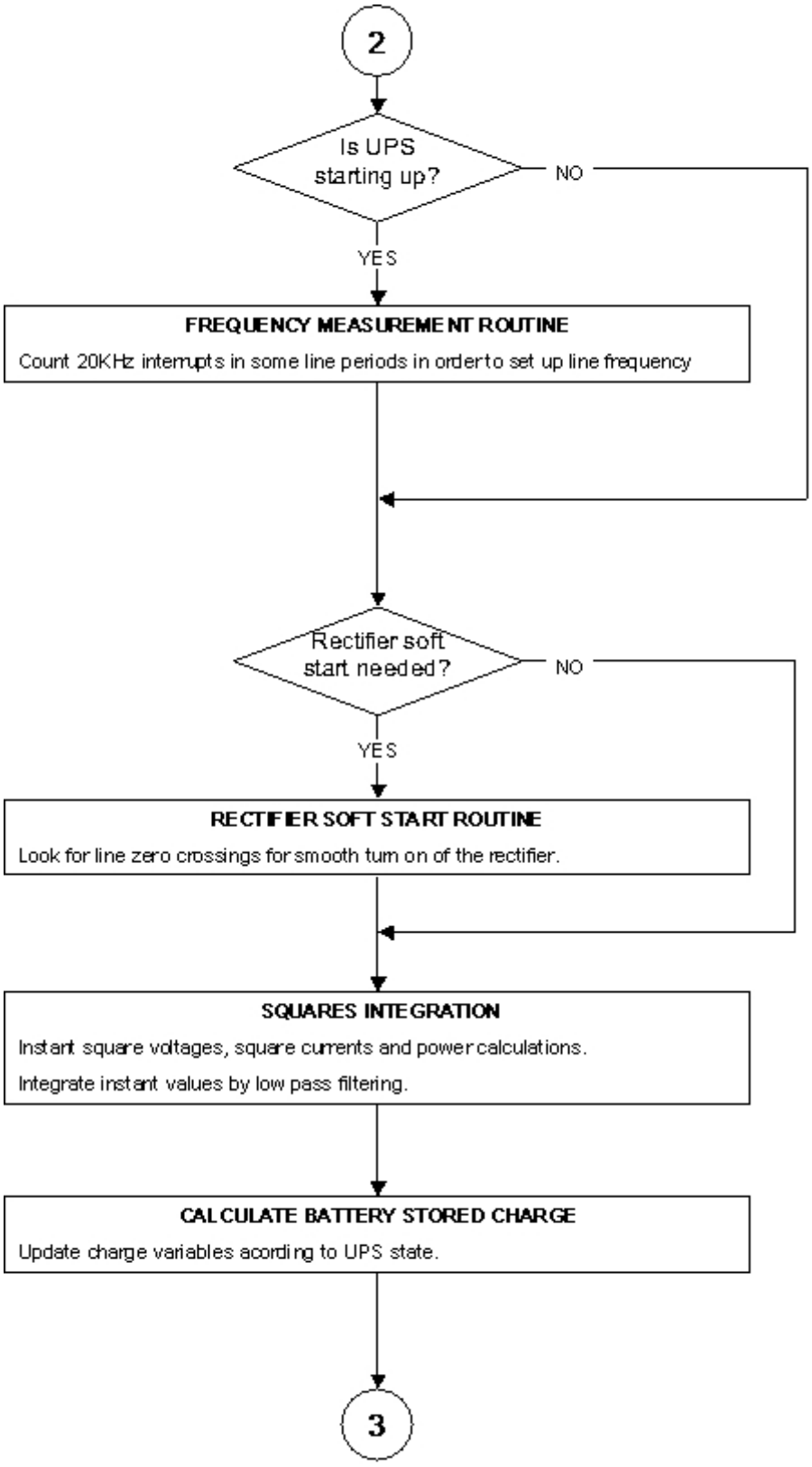


Figure 7-6. Flow Diagram for the Interrupt Service Routing *AD1_OnEnd* (3 of 5)

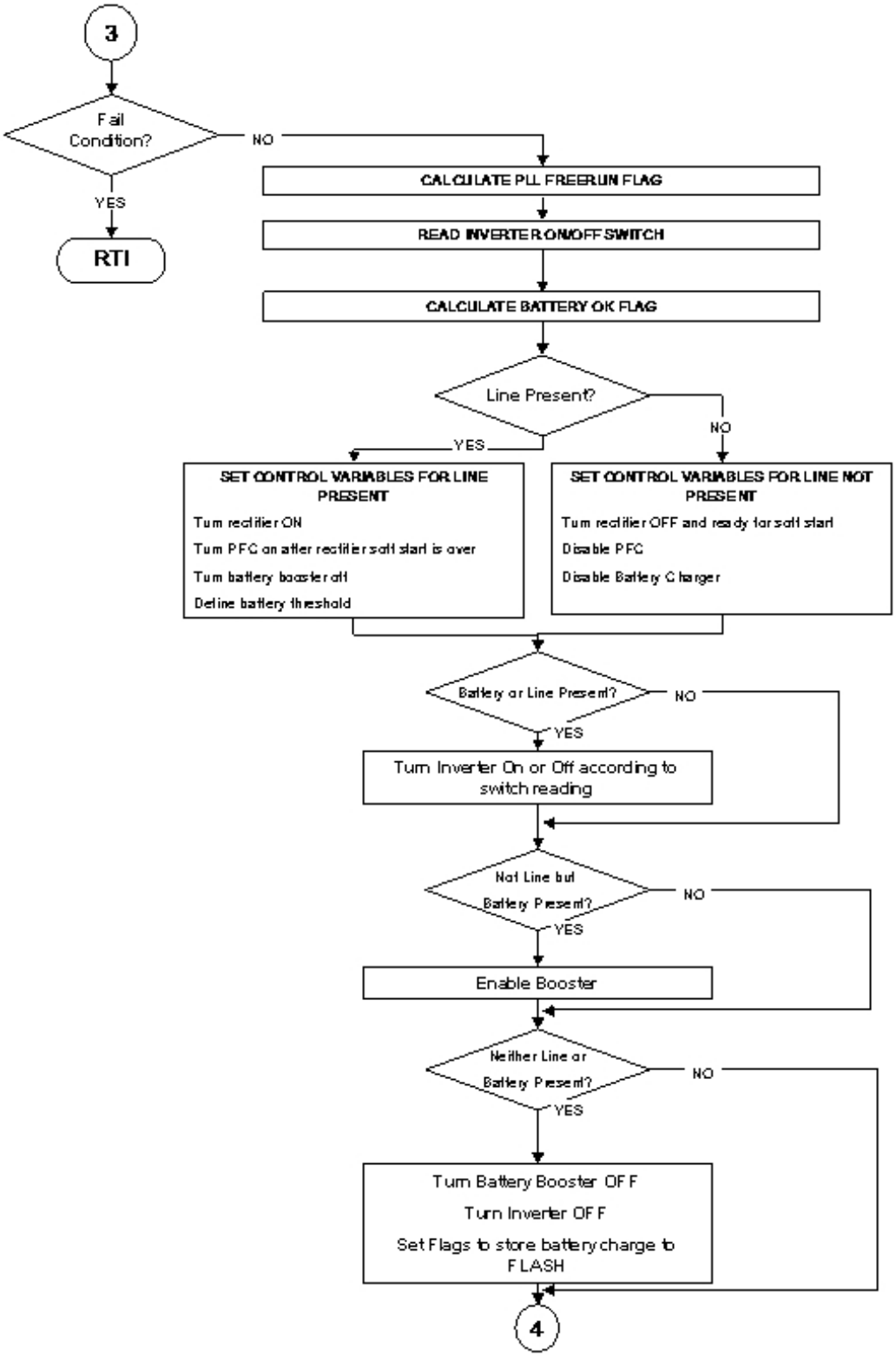


Figure 7-7. Flow Diagram for the Interrupt Service Routine *AD1_OnEnd* (4 of 5)

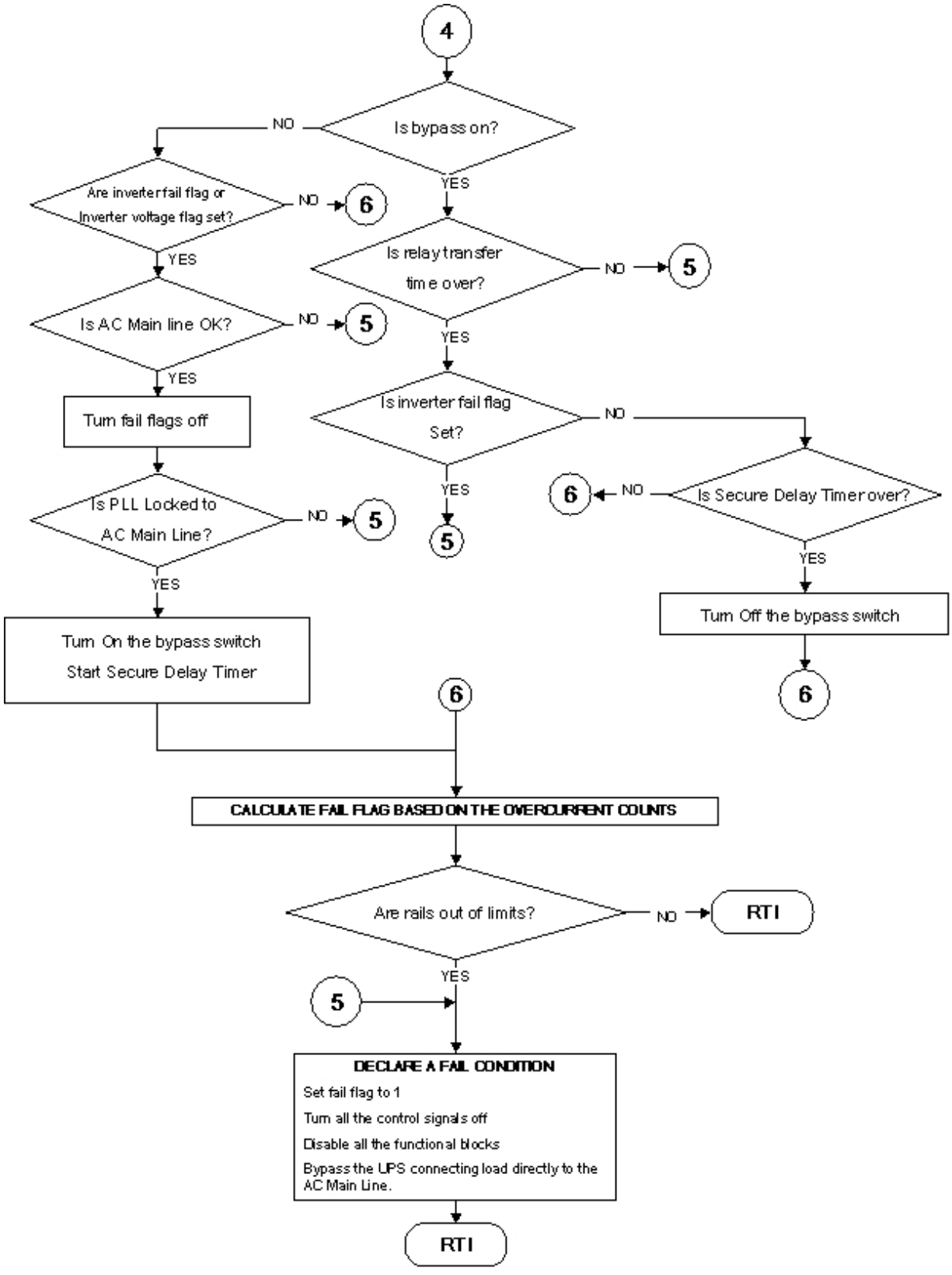


Figure 7-8. Flow Diagram for the Interrupt Service Routine *AD1_OnEnd* (5 of 5)

7.3.2 UPS Overcurrent Protection Interrupt Handlers

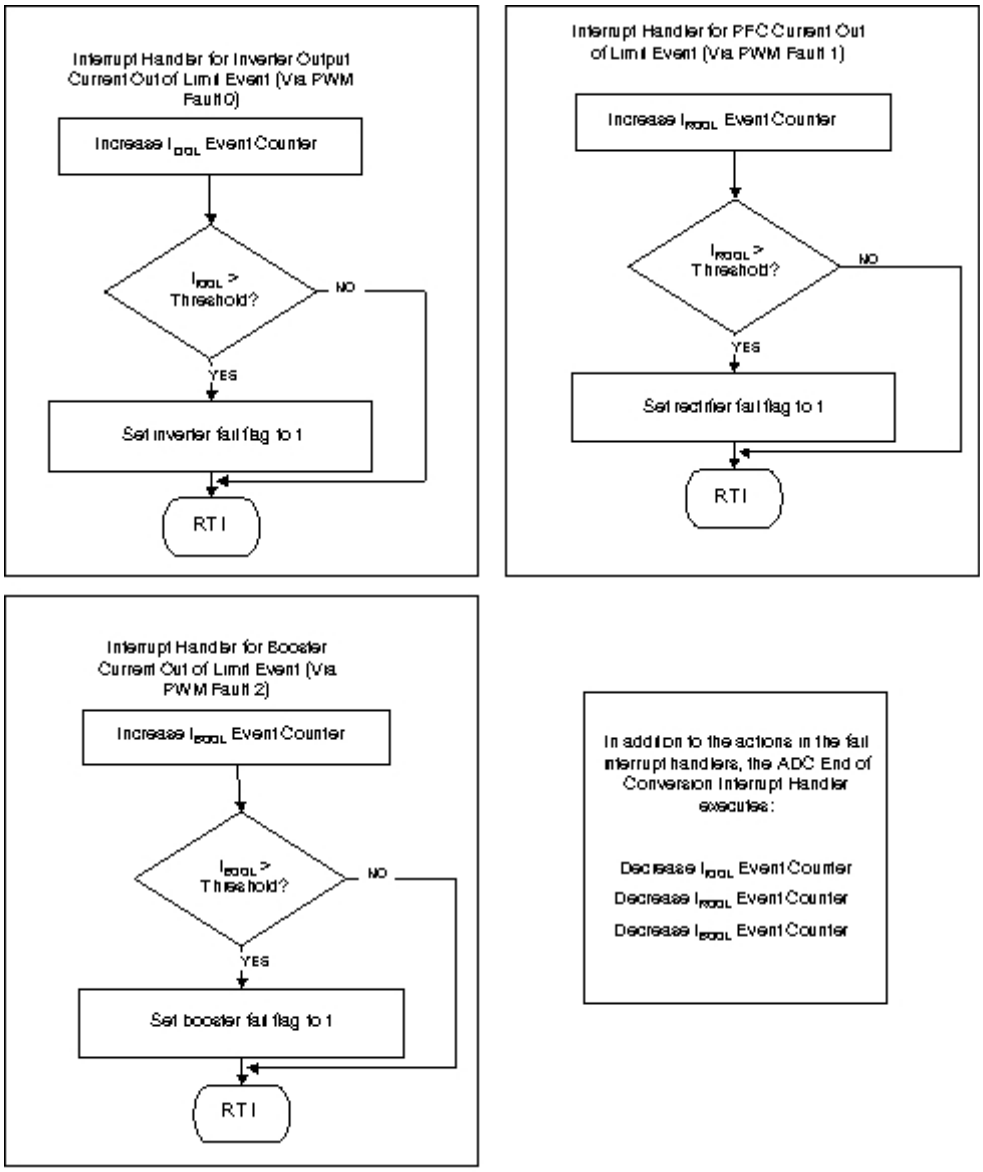


Figure 7-9. Overcurrent Management

Three overcurrent events are sensed and connected to the Fault inputs of the PWM modules. These are:

- Inverter Current out of Limits
- Rectifier (PFC) Current out of Limits
- Booster Current out of Limits

PWM Fault inputs are set up to disable the corresponding PWM pins when overcurrent is detected.

A single overcurrent event should not disable the complete system, as peak currents can occur when switched capacitive loads are connected (i.e., a power supply implemented with rectifier and capacitor). For this reason, a current out of limits counter is implemented in all of the PWM-controlled systems. Every time the interrupt handler routine is called, the counter increases by an amount, A. The 20kHz End of Conversion interrupt decreases the counter by an amount, B. The UPS is sent to the fail condition only if:

kA - Bn > Threshold

where:
k is the number of times the current out of limits event occurs
n is the number of times the 20kHz routine is called

The weights A and B and the threshold are selected to balance a trade-off requiring the system not be shut down when capacitive loads cause periodic overcurrent events, and to shut down the UPS when a continued short circuit condition is detected.

7.3.3 Battery Temperature Reading

The interrupt service routine *AD2_OnEnd* reads and stores the battery temperature to a variable. The execution of the ISR is shown in **Figure 7-10**.

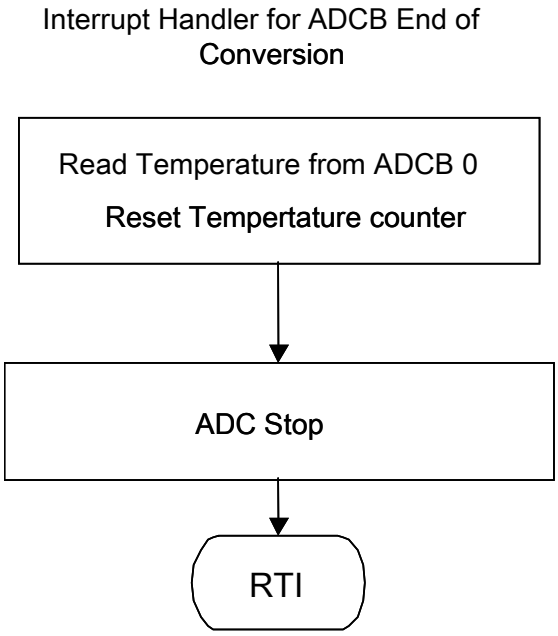


Figure 7-10. Battery Temperature Reading

7.3.4 Delay_Timer_OnInterrupt

This routine merely counts 21 times 140ms (the time interval of the timer) in order to generate a time delay of 3 seconds following the controller's core start-up. This is a wait time to allow the auxiliary power supplies to stabilize.

7.4 Program Loop Timing

The timing of the program originates at the PWM, configured for one complete cycle reload, and center-aligned. The modulo of the PWM is set to 1500, generating a 20kHz triangle wave from the 60MHz internal bus clock.

The SYNC signal of the PWM is passed to Timer C_2 to provide the sync signal to the ADCA peripheral. At Timer C_2, the SYNC signal is delayed in order to reduce the time between the sampling and the updating of the values at the PWM, enhancing the stability of the discrete time control algorithms.

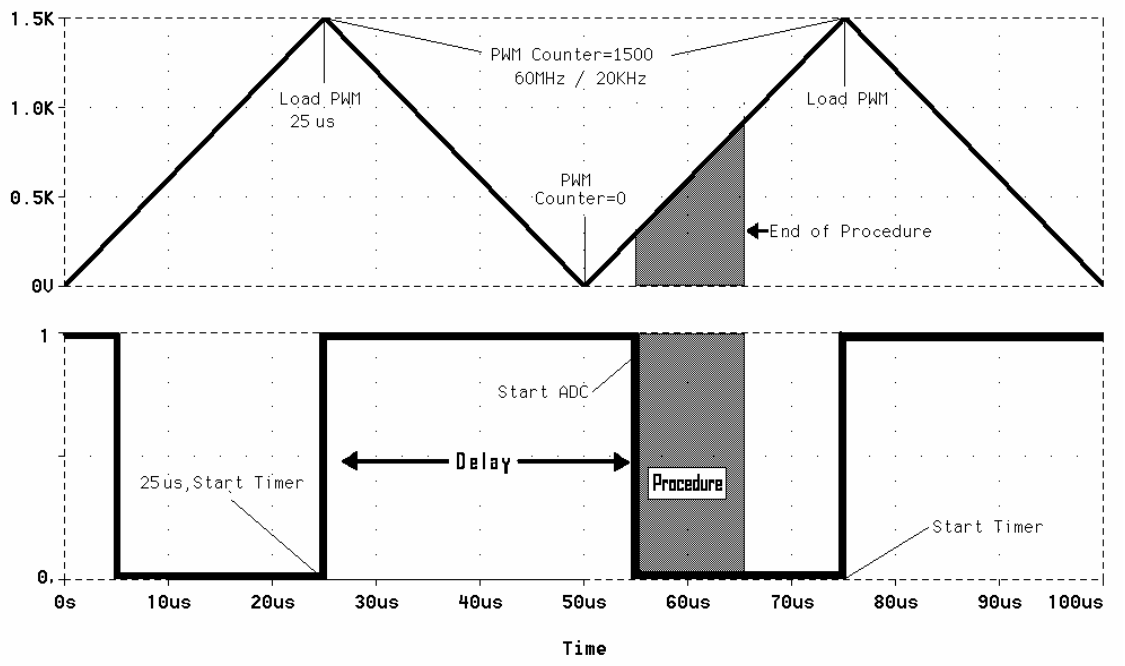


Figure 7-11. Program Loop Timing

After the configured delay, Timer C_2 signals the ADCs to start conversion. At the end of the conversion, the ADC module interrupts the core processor. All control loops are executed in this interruption.

7.5 Inverter Control Loop

The control network for the inverter is constructed with an inner current PI control loop and an outer PID control loop as shown in **Figure 7-12**.

The outer control loop receives the sinusoidal wave synthesized by the PLL module as a reference and compares it with the inverter output voltage. The error signal passes through a PID compensator whose output constitutes the set point for the inner PI control loop. This current set point is compared with the load current.

The transfer function of a discrete time PID control loop is calculated by this equation:

$$C(Z) = K_T\left[K_P + \frac{Z}{Z-1}K_I + \frac{Z-1}{Z}K_D\right]$$

The transfer function of a discrete time PI control loop is calculated by the following equation::

$$C(Z) = \left[K_P + \frac{K_I * Z}{Z - 1} \right] K_T$$

The sine wave generated is used as the set point of the inverter control. The inverter uses the output voltage and current as sensing inputs to the control, which have a double control loop topology, inner PI current control and outer PID voltage control. **Figure 7-12** shows the details of the inverter loop as implemented in the source code. Signal names are the actual variable names in the C code.

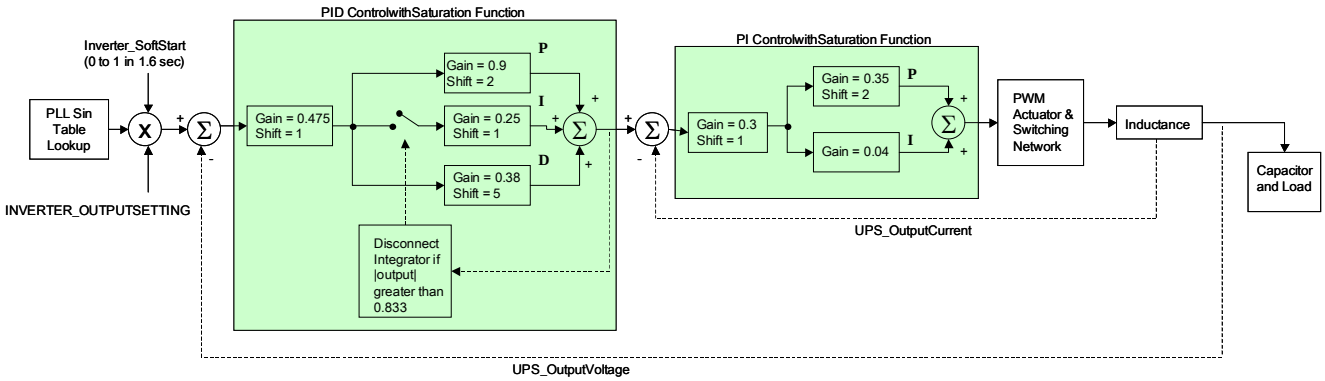


Figure 7-12. Inverter Control Loop Diagram

7.6 PFC Control Loop

Figure 7-13 shows the actual function implementation of the PFC control loop. It consists of two controls, one to control the rail-to-rail voltage and another to control the current drawn to the AC main line. The current set point for the inner loop is the AC line voltage times a factor linearly proportional to the error signal at the rails (and finally dependent on the UPS load current). The hardware implementation in this UPS (please refer to **Figure 7-13**) requires the inner control to work with the absolute values of the signals in order to avoid discontinuities at the current control output.

Both control loops use Proportional–Integral (PI) compensators, implemented with 32-bit integrators. The Z domain transfer function of a PI compensator is:

$$C(Z) = \left[K_P + \frac{K_I * Z}{Z - 1} \right] K_T$$

where:
K_P is the proportional gain
K_I is the integral gain

A total gain, K_T, is implemented in order to allow flexibility when tuning the control; i.e., varying the total loop gain without the necessity of modifying the individual gains.

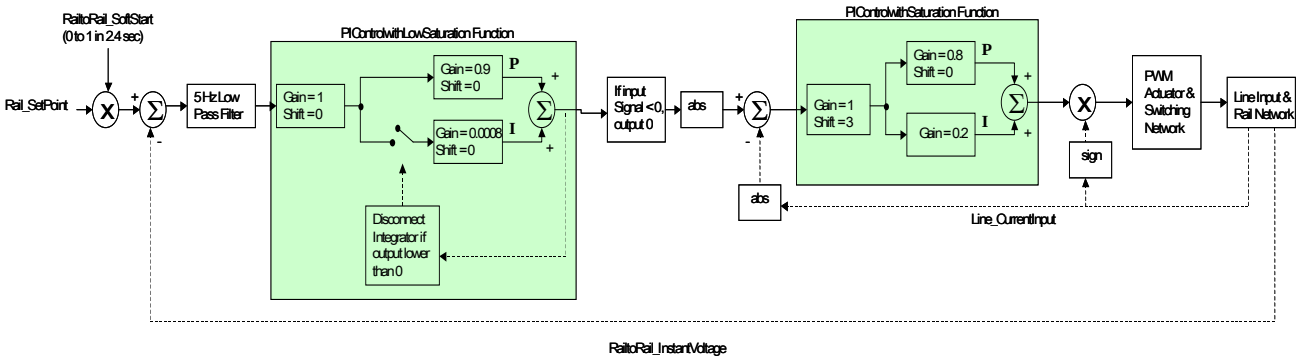


Figure 7-13. PFC Control Loop Diagram

7.7 Battery Booster Control Loop

The battery booster control signals for the switches are not complementary but 180° phase shifted. This is implemented at PWMA channels 2 and 3, defining the compare value of channel 3 as 1500 (the full scale of the PWM) minus the compare value of channel 2, and inverting its output. A protection time of 4ms is implemented between the active state of these signals at maximum duty cycle .

Decimation by 16 is implemented for a routine sample frequency of 1250Hz. **Figure 7-14** shows the battery booster system.

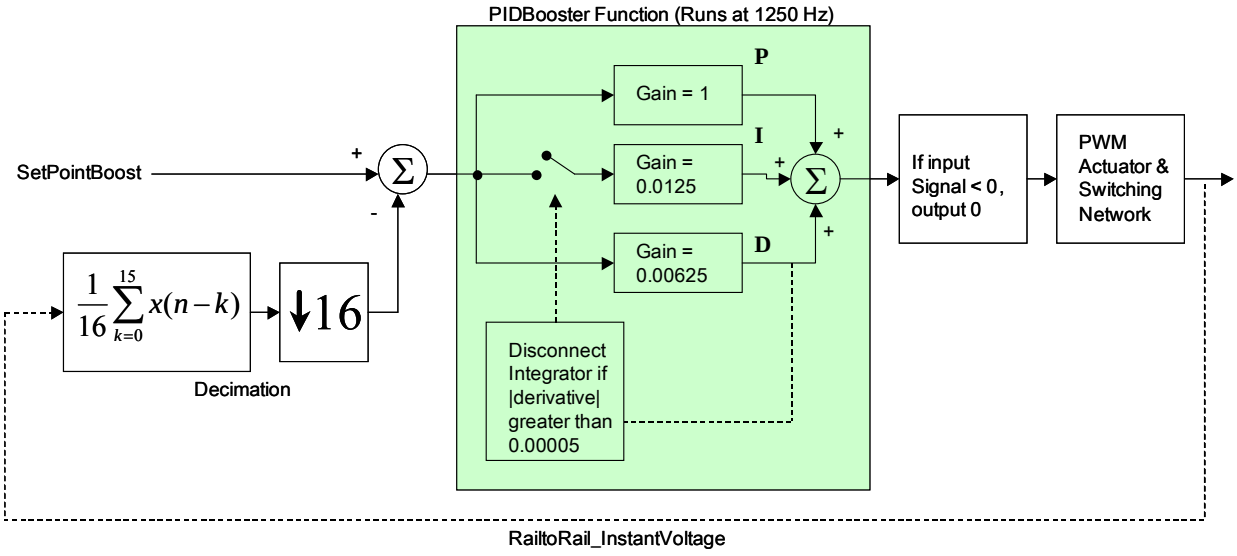


Figure 7-14. Battery Booster Control Loop

7.8 Battery Charger Control Loop

The battery charger is a constant current power supply. The control loop uses the measured battery current to calculate the appropriate voltage set point. Decimation of the sampling rate by 16 allows for better precision for the battery voltage variable. The variables are sensed at 20kHz, but the battery voltage routine is executed at 20kHz/16 = 1250Hz. Moving averaging of the samples is implemented for noise filtering.

Due to the slow speed of the system, the integrator input of the voltage control is disabled when the output of the controls reaches a defined limit.

Please see **Figure 7-15** for a detailed diagram of the battery charger system.

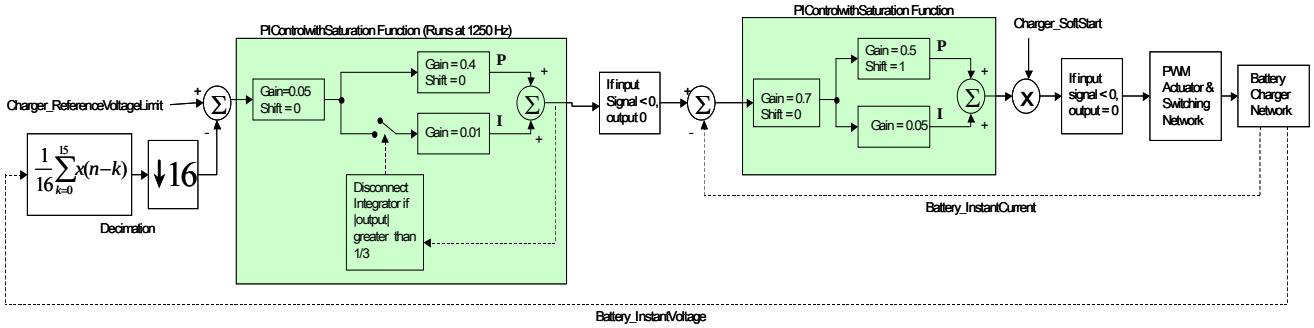


Figure 7-15. Battery Charger Control Loop

7.9 Phase Locked Loop Routine

7.9.1 Phase Discriminator

The Phase Locked Loop uses a phase and frequency discriminator inspired by the Phase Comparator II of the Fairchild Semiconductor's CD4046, modified to fit a synchronous state machine. The continuous arrows in **Figure 7-16** correspond to the asynchronous state machine. The dashed arrows were added to fit the synchronous implementation.

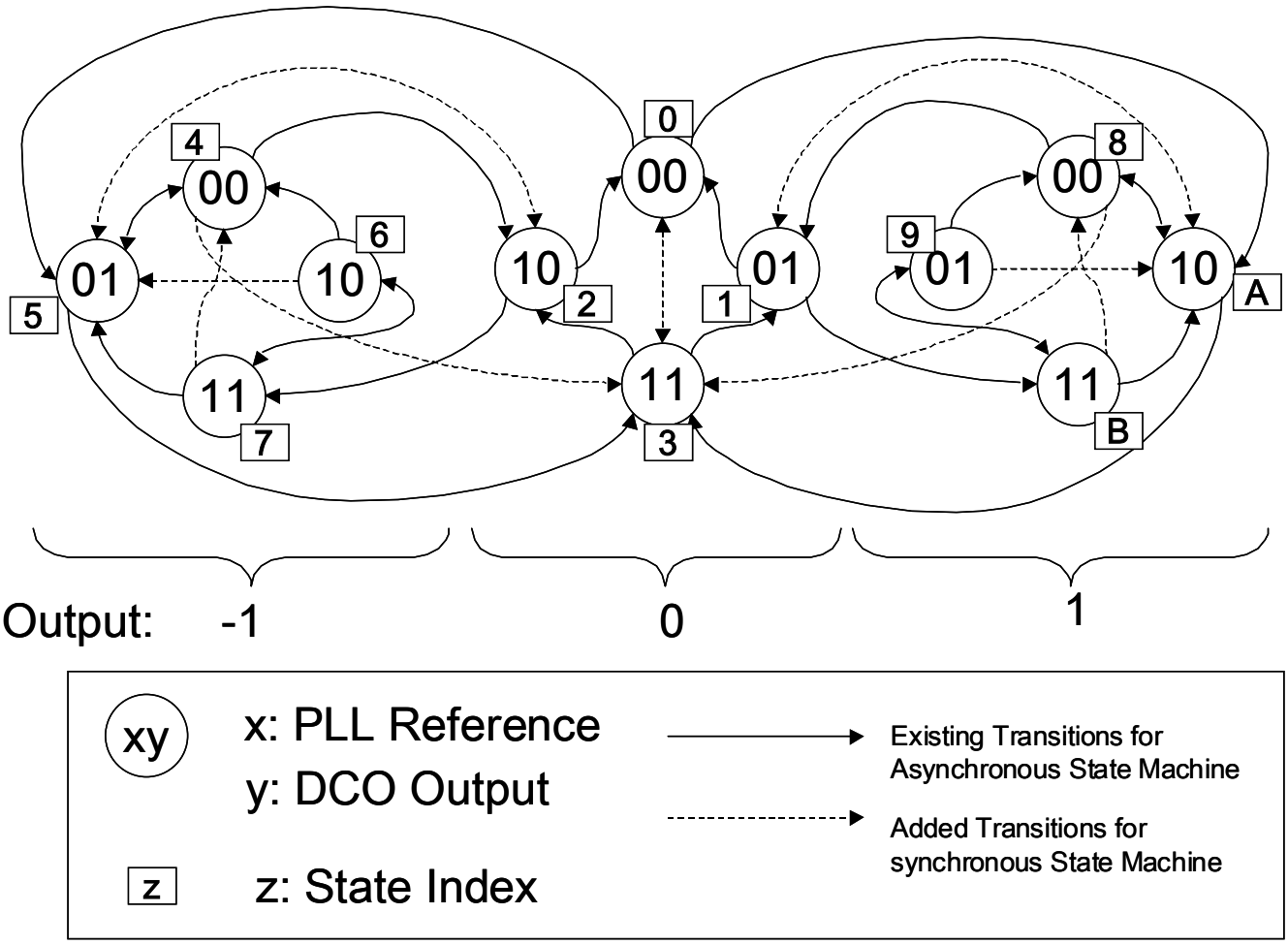


Figure 7-16. Synchronous Phase Discriminator State Machine

Table 7-17 is the transition table for the state machine. The input description is formed by x, the sign of the reference signal to the PLL, and y, the sign of the Digitally Controlled Oscillator (DCO) output. The present state combines with the input *xy* and determines both the next state and the output of the phase discriminator.

Table 7-17. Transitions of the Synchronous Phase Discriminator

State Index	Input xy	Next State (Depending on Input xy)				Output
		11	10	01	00	
	0	3	A	5	0	0
	1	B	A	1	0	0
	2	7	2	5	0	0
	3	3	2	1	0	0
	4	3	2	5	4	-1
	5	3	2	5	4	-1
	6	7	6	5	4	-1
	7	7	6	5	4	-1
	8	3	A	1	8	1
	9	B	A	9	8	1
	A	3	A	1	8	1
	B	B	A	9	8	1

The phase discriminator output feeds a phase accumulator that integrates from the reference signal's positive zero crossing to the next one. After one iteration is ended, the accumulator output is fed to a PI control, then reset. The Proportional—Integral control routine runs at every positive zero crossing.

The output of the control is added to the sine wave index, transforming the table look-up algorithm in a numerically controlled oscillator, then closing the Phase Locked Loop.

7.9.2 Control Loop

The PLL reference may be either AC Main Line or Freerun. The Freerun operation is chosen if the AC Main Line frequency is out of specifications, as explained in **Section 7.10**. See **Figure 7-18** for a block diagram of the PLL Control Loop.

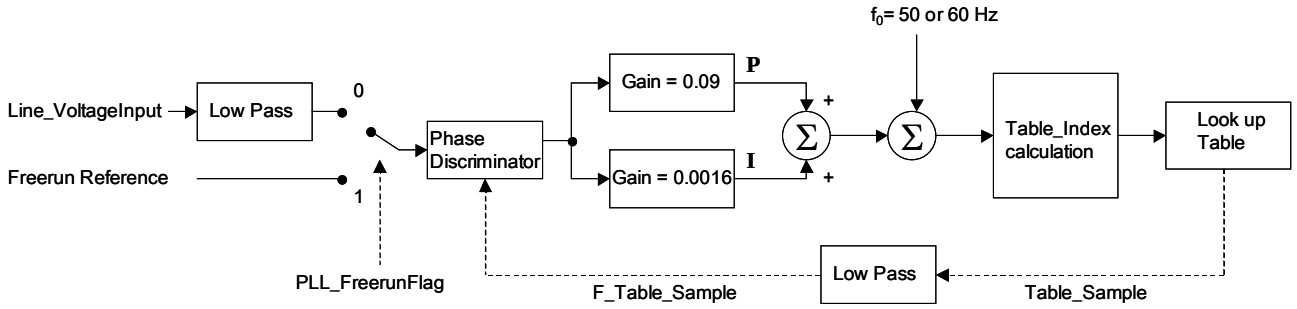


Figure 7-18. PLL Control Loop

7.9.3 Sine Wave Look-Up Table

The sine wave is generated by look-up of a table of 800 samples from a full cycle of a sine wave, scaled from –1 to 1 in Frac16 fixed point fractional number format.

At 20kHz sample frequency, the sine wave table pointer increments 2.4 memory positions per sample to yield a 60Hz wave. To generate a 50Hz wave, the increment is 2.0 positions. The custom data type *FixedInt* was created to implement fractional advances. The phase value is stored in a long signed integer. Only the most significant 16-bit integer word is used to address two consecutive samples of the table, then a linear interpolation using the fractional part of the index is performed.

```
typedef union{
struct{
unsigned int IntL;
int IntH;
}I;
long L;
}FixedInt;
```

FixedInt Data Type Definition

The *FixedInt* can be used as a long (*variable.L*) or as its integer part (*variable.I.IntH*), plus its unsigned fractional part (*variable.I.IntL*). The integer part is used to address the table, and is a modulo 800 circular counter.

7.10 Frequency Measurement Routine

The AC main line frequency is measured with two purposes:

- Auto-sensing line frequency when the UPS starts up
- Determining the quality of the AC main line frequency
  If out of limits, the UPS will switch to the internal reference mode.

To avoid divisions, period rather than frequency is measured in the form of interrupt counts (50µs each). The time window for interrupt counts is 10 cycles of the line signal.

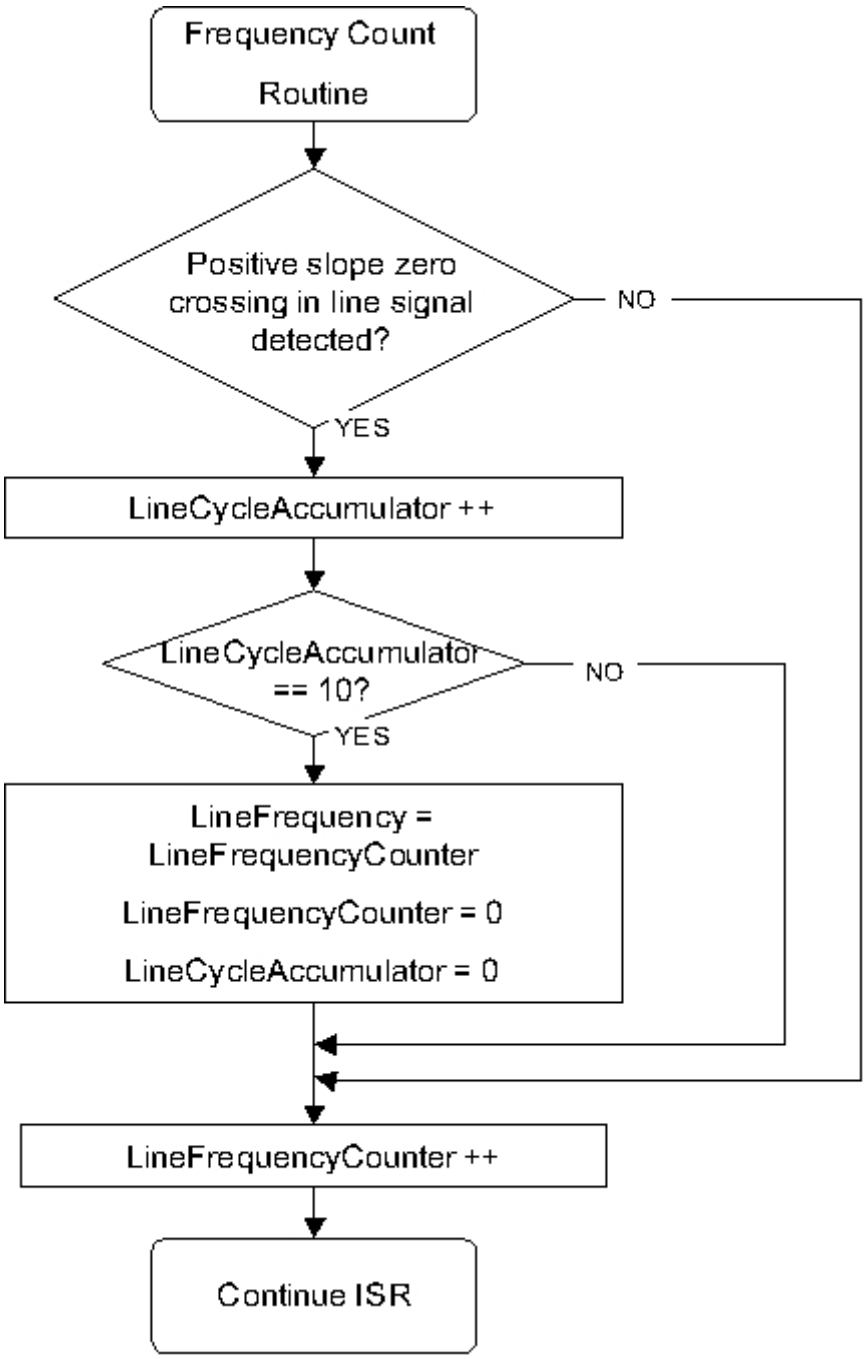


Figure 7-19. Frequency Measurement Routine

7.11 Rectifier Soft Start Routine

In order to implement the rectifier soft start, a software synchronized ramp is generated and compared with the rectifier trigger level signal, which starts at a value higher than the peak of the ramp, then decrements towards zero. The result of the compare operation generates a PWM signal that is set at the zero crossing time minus time T, and is always reset at the zero crossing. As the trigger level decrements, the time τ increases from zero to a half period. This signal can drive the input SCRs that form the full wave rectifier at the input of the UPS.

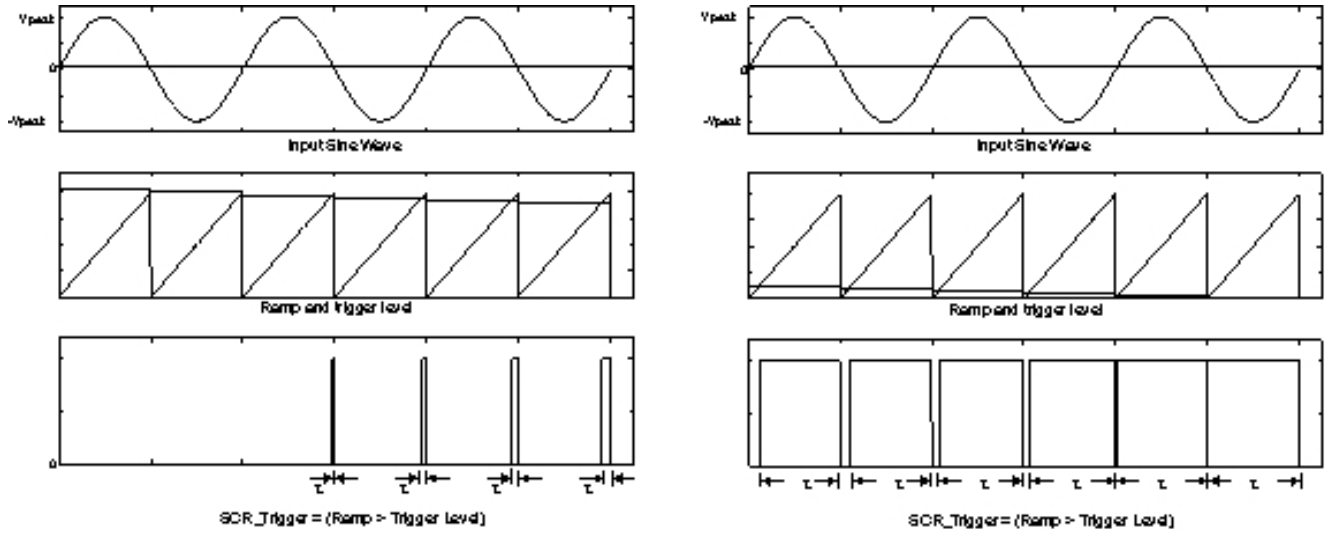


Figure 7-20. SCR Control Signal Generation at the Beginning and End of the Rectifier Soft Start

7.12 Root Mean Square (RMS) Sensing

RMS sensing is accomplished by filtering the squared input signal by two cascaded second-order digital filters. This operation is done in real time, sample-by-sample in the converter interruption routine. The next two blocks are performed in noncritical processing times.

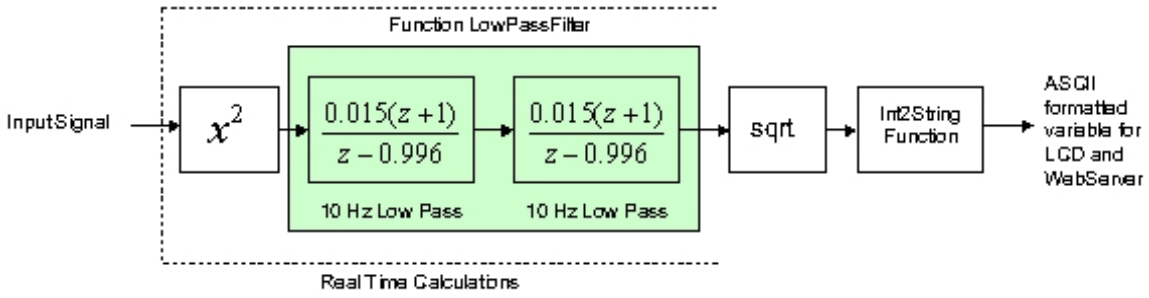


Figure 7-21. RMS Sensing System

7.13 Power Sensing

Like RMS sensing, power sensing is performed in two operations, first multiplying and filtering in real time, then adjusting the display processing routine. This routine is applied to calculate both the input and the output power.

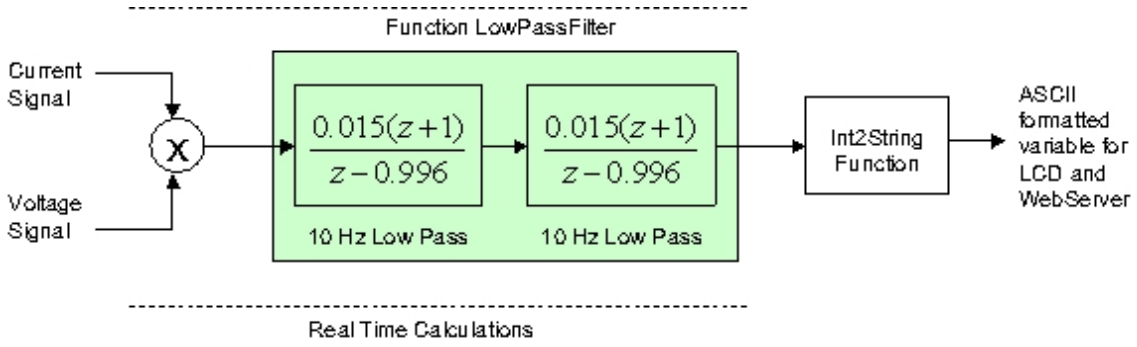


Figure 7-22. Power Sensing System

7.14 Routine for Measurement of the Battery Stored Charge

Measuring the battery stored charge is complicated by several requirements:

- Determining the state of the battery charge when first connected to the UPS
- The ability to modify the algorithm according to the capacity of the battery connected
- Integrating the charging/discharging current over long periods (hours for the recharging process), when the system sampling rate is 20kHz
- Performing low gain integrations to minimize the quantization effect
- Correcting the effect of different sensing circuits
  The hardware is implemented with one sensing circuit for the battery charge, but a different sensing circuit for the battery discharge. Shunt resistors have in a 10:1 ratio for charge/discharge, respectively.

These requirements are met by implementing three nested integrators with variable gains and integration time windows, depending on whether the battery is accepting or yielding charge. The same constants are used to tune the charge measurement algorithm, depending on the battery size.

Full charge is indicated when the outer integrator reaches one in Frac32 notation.

The first time that the batteries are connected to the system, the battery stored charge is unknown until the charging current decreases to reach the floating regime. When such a condition is detected, the battery stored charge variable is forced to one (100%), and the integration/deintegration summations are executed to track the battery charge. This condition is stored in the Valid Battery Charge flag variable.

The charging capacity integration while the battery is accepting charge is then:

$$\text{Battery\_StoredCharge} = \sum_{\text{IntegrationWindow3}=1}^{n} \frac{2}{1200}\left(\sum_{\text{IntegrationWindow2}=1}^{1200} \frac{1}{1200}\left(\sum_{\text{IntegrationWindow1}=1}^{1000} \frac{3}{2000}\text{Battery\_InstantCurrent}\right)\right)$$

The integration used to update the charge when the battery is delivering charge is:

$$Battery\_StoredCharge = \sum_{IntegrationWindow\ 3=1}^{n} \frac{2}{500}\left(\sum_{IntegrationWindow\ 2=1}^{1200} \frac{1}{120}\left(\sum_{IntegrationWindow\ 1=1}^{1000} \frac{-30}{2000} Battery\_InstantCurrent \right)\right)$$

The constants presented correspond to 7.2AH/10H batteries. Note the asymmetry between charging and discharging constants.

To avoid deletion of the charge information when the system is powered down, the flags and integrators are stored in Flash memory. It is the operator's responsibility to record this information in the system.

7.15 General Purpose Digital Filters Used in the Implementation

7.15.1 10Hz Low Pass Filter, 2 Stages

When averaging is required in RMS and power sensing, a cascade of two 10Hz low-pass filters is implemented to minimize quantization effects.

Each stage corresponds to the digital implementation of a first-order Butterworth filter, with a 3dB cut-off frequency of 10Hz.

The Laplace transfer function of the analog filter is:

$$H(s) = \frac{6.2732x10^{6}}{6.299x10^{6} + 99843S}$$

Applying a bilinear transformation over the transfer function, with fs = 20kHz yields the Z-domain transfer function:

$$H(Z) = \frac{1.56833x10^{-3} + 1.56833x10^{-3}Z^{-1}}{1 - 0.99686Z^{-1}}$$

The transposed direct form II is chosen for discrete time implementation using 32-bit registers for the accumulators (w_1 and w_2):

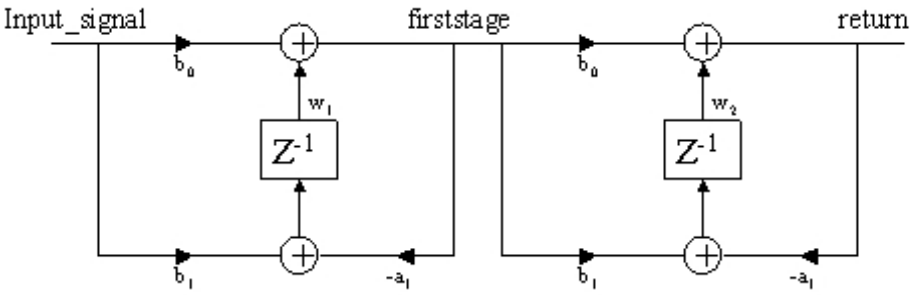


Figure 7-23. Implementation of Low-Pass Filter

The calculated gains are:
$b_0 = 1.56833408x10^{-3}$
$b_1 = 1.56833408x10^{-3}$
$a_1 = -0.996863$

7.15.2 High-Frequency Noise Rejection Filter

A second-order low-pass filter is used when high-frequency noise is to be rejected from the line signal (i.e., before zero crossing algorithms).

The design starting point is an analog low-pass Butterworth filter with a 3dB cut-off at 1kHz. For a second-order filter, the Laplace transfer function is:

$$H(S) = \frac{180S^2 - 1.44x10^7 S + 6.4266x10^{16}}{1.6012x10^9 S^2 + 1.4346x10^{13} S + 6.4267x10^{16}}$$

Applying the bilinear transform to calculate a discrete time implementation yields:

$$H(Z) = \frac{2.0083x10^{-2} + 4.0165x10^{-2} Z^{-1} + 2.0083x10^{-2} Z^{-2}}{1 - 1.561Z^{-1} + 6.4135x10^{-1} Z^{-2}}$$

For the accumulators (w_1 and w_2), 32-bit registers were chosen. To allow the implementation of a1 = -1.561, whose magnitude is greater than one, without losing the use of saturated arithmetic, the coefficients are modified and shifts to the left are introduced, resulting in the following implementation of a transposed direct form II:

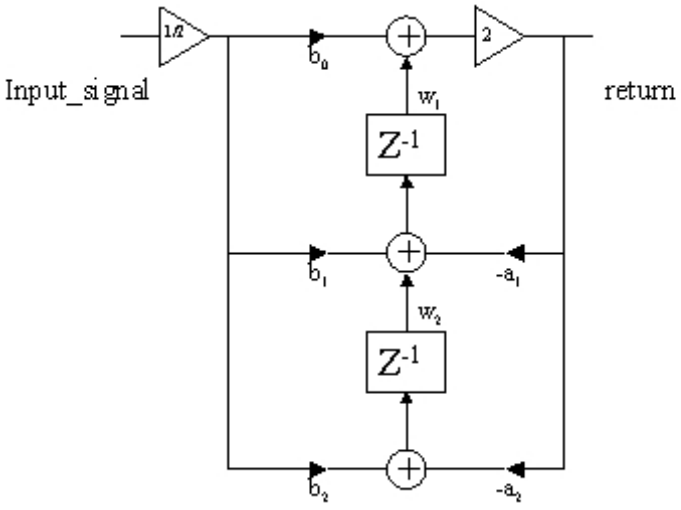


Figure 7-24. Implementation of the High Frequency Reject Filter

The final gains used in this implementation are:
$b_0 = 2.0083x10^{-2}$
$b_1 = 4.0165x10^{-2}$
$b_2 = 2.0083x10^{-2}$
$a_1 = -7.80509\ x10^{-1}$
$a_2 = 3.206757\ x10^{-1}$

7.16 Flash Memory Access

Flash memory is used to store the charge value and flags controlling the battery load measurement. The embedded bean functions *SetLongFlash* and *GetLongFlash* interface with Flash.

7.17 Development Tools Used for Software Implementation

7.17.1 Processor Expert™

Processor Expert beans have been used to initialize the internal PLL bus generator, ADC, PWMs, Timer interruption, pulse generation, GPIO pins, interrupt vector and interrupt service routines.

7.17.2 Intrinsic 56800E Functions

The following functions from the "*intrinsics_56800E.h*" library are used:
turn_on_sat();
turn_off_conv_rndg();
Word16 add(Word16,Word16);
Word16 mult(Word16 , Word16);
Word 16 msu(Word32,Word16,Word16);
Word 16 sub(Word16,Word16);
Word16 mac_r(Word32,Word16,Word16);
Word16 abs_s(Word16)
Word16 shlfts(Word16,Word16);
Word16 msu_r(Word32,Word16,Word16);
Word16 negate(Word16);
Word32 L_mac(Word32,Word16,Word16);
Word32 L_msu(Word32,Word16,Word16);
Word32 L_mult(Word16,Word16);
Word32 L_shifts(Word32, Word16)
Word32 L_add(Word32, Word32);

These functions allow filter calculation using the 56800E core processor's full set of features for fixed-point arithmetic and saturation.

7.17.3 Direct Register Writes

Direct write to configuration registers by the Processor Expert Support Library *PESL* was used in the following cases:

- Timer C initialization
- Enable and Disable Software control in the PWM module
- Turning PWMs 0 and 1 off and on
- PWM and Timer A duty cycle set up

7.17.4 Assembly Inlines

No explicit assembly inlines have been used in the C source code.

8. Connectivity Software Design Considerations

8.1 Connectivity Description

The OUPS' connectivity allows the user to communicate with the board via the TCP/IP protocol, supporting such known standards as HTTP Web pages or e-mails.

The software used for OUPS connectivity will run on a 56F8346 controller, a member of the 56F8300 family. The 56F8346's operational software is downloaded by the CodeWarrior IDE into the device's internal Flash and executed from there. The TCP/IP stack software provides the functionality that allows remote UPS configuration and monitoring using transparent bi-directional transfer of Internet Protocol (IP) and Transmission Control Protocol (TCP) traffic between an HTTP client and the UPS unit across an Ethernet network.

This software is based in OpenTCP Software, an open source project developed by Viola Systems. It brings a highly reliable, portable TCP/IP stack for 8/16-bit microcontrollers. More information about OpenTCP software can be found at: http://www.opentcp.org/.

For detailed documentation of the OpenTCP API, please refer to:
http://www.opentcp.org/documentation/api/html/index.html

8.2 Peripheral and I/O Pins Assignment

Table 8-1. Digital Outputs and Inputs Used for Connectivity

Peripheral Port	Function
A0-A15	Address Bus for Ethernet Controller
D0-D15	Data Bus for Ethernet Controller
CS2	Chip Select for Ethernet Controller
WR	Write Signal for Ethernet Controller
RD	Read Signal for Ethernet Controller
IRQA	Interrupt Request from Ethernet Controller
Reset	Reset

8.3 Implementation

The OpenTCP software allows an easy implementation for defining proper layers.

Figure 8-2 shows the implementation of the OpenTCP software for the 568F346.

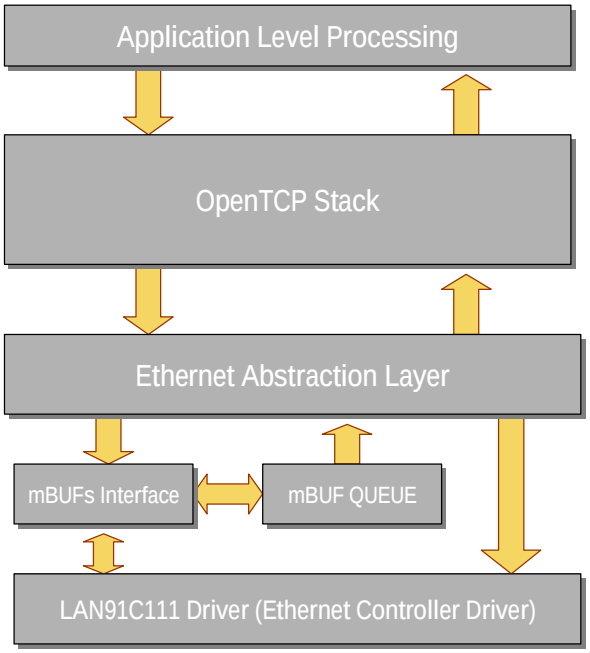


Figure 8-2. Implementation of OpenTCP for OUPS

8.3.1 LAN91C111 Driver

The LAN91C111 Ethernet Controller allows the processor to communicate with the Ethernet bus. Its driver handles the proper initialization for a correct physical layer and the reception and transmission of packages.

8.3.2 Network Buffers

This block is the part of the OpenTCP port for the 56F8346 that handles the buffering of frames coming from the Ethernet Controller adapter. Ethernet frames travel up the stack in the form of special buffers, called mBufs, special structures that include fields to support the OpenTCP macros to process a frame on a byte-by-byte basis.

8.3.3 Ethernet Abstraction Layer

The Network Adapter Block is the part of the TCP/IP stack that interfaces the packet processing portion of the stack to the actual Ethernet Controller. This block handles Ethernet frames coming and going through the network. Incoming frames are detected by the Network Adapter Block and transferred to the top portion of the OpenTCP stack. The TCP/IP connectivity interfaces to the UPS control software by sending and receiving frames through the well-defined OpenTCP networking interface.

8.3.4 OpenTCP Stack

The OpenTCP Stack is open source, reliable code which handles packet processing and allows a direct and standard interface with the OUPS code.

8.3.5 Application Level Processing

Application functions are called by the OUPS and allow the OUPS to perform such required activities, as servicing HTTP Web pages or sending e-mails.

8.4 Connectivity Initialization

Connectivity initialization is performed during the UPS initialization routine. All layers, from the lowest level and continuing upwards, must be initialized properly in order to execute the application layers, which in this case are the HTTP Server and the SMTP Client.

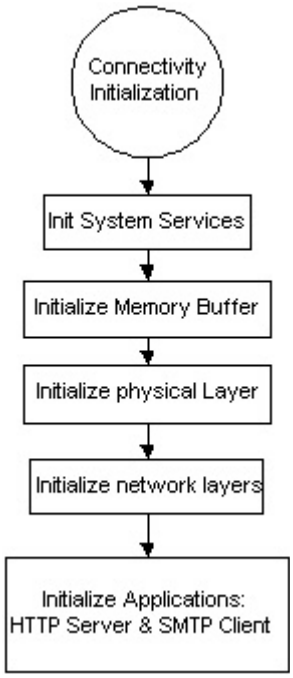


Figure 8-3. Connectivity Initialization

8.4.1 System Services Initialization

Routines in **Table 8-4** are performed to initialize system services for connectivity.

Table 8-4. System Services Initialization Functions

Function	Description
void timer_pool_init(void)	This function resets all timer counters to zero and initializes all timers to an available (free) state
void nBufInit(void)	Initializes the memory buffer subsystem

8.4.2 Physical Layer Initialization

Table 8-5 details routines used to initialize the Ethernet Controller and physical layer.

Table 8-5. Physical Layer Initialization Functions

Function	Description
void eth_init(void)	Generic initialization for 56800E SMSC Ethernet boards Called from application start-up code Initializes the Ethernet chip select, interrupt, etc.
void smc_init(void)	Initialization routine called by network device driver code
int smc_open(void)	Opens and initializes the SMC chip/board

8.4.3 Network Layers Initialization

Routines in Table 8-6 are used to initialize all network layers.

Table 8-6. Network Layers Initialization Functions

Function	Description
void arp_init (void)	Call this function to properly initialize Address Resolution Protocol (ARP) cache table and so that ARP allocates and initializes a timer for its use
INT8 udp_init (void)	Initializes User Datagram Protocol (UDP) socket pool to put everything into a known state at start up
INT8 tcp_init (void)	Initializes all sockets and corresponding Transmission Control Blocks (TCBs) to a known state Timers are also allocated for each socket and everything is brought to a predefined state

8.4.4 Application Layer Initialization

Routines in Table 8-7 initialize the application layer. The OUPS works as an HTTP Server and an SMTP Client.

Table 8-7. Application Layer Initialization Functions

Function	Description
INT8 https_init(void)	Initializes the HTTP server
void smtpc_init (void)	Initializes the SMTP client

8.5 Main Application

After initialization, the software enters an infinite loop. During this loop, the OUPS will perform simple monitor and control functions along with the main connectivity loop. The most critical section of control is executed in interrupt service routines, and these will always have the highest priority.

Figure 8-8 shows the tasks performed during the main application loop, although some of the activity is performed in the interrupt context when the microcontroller receives an IRQA interrupt.

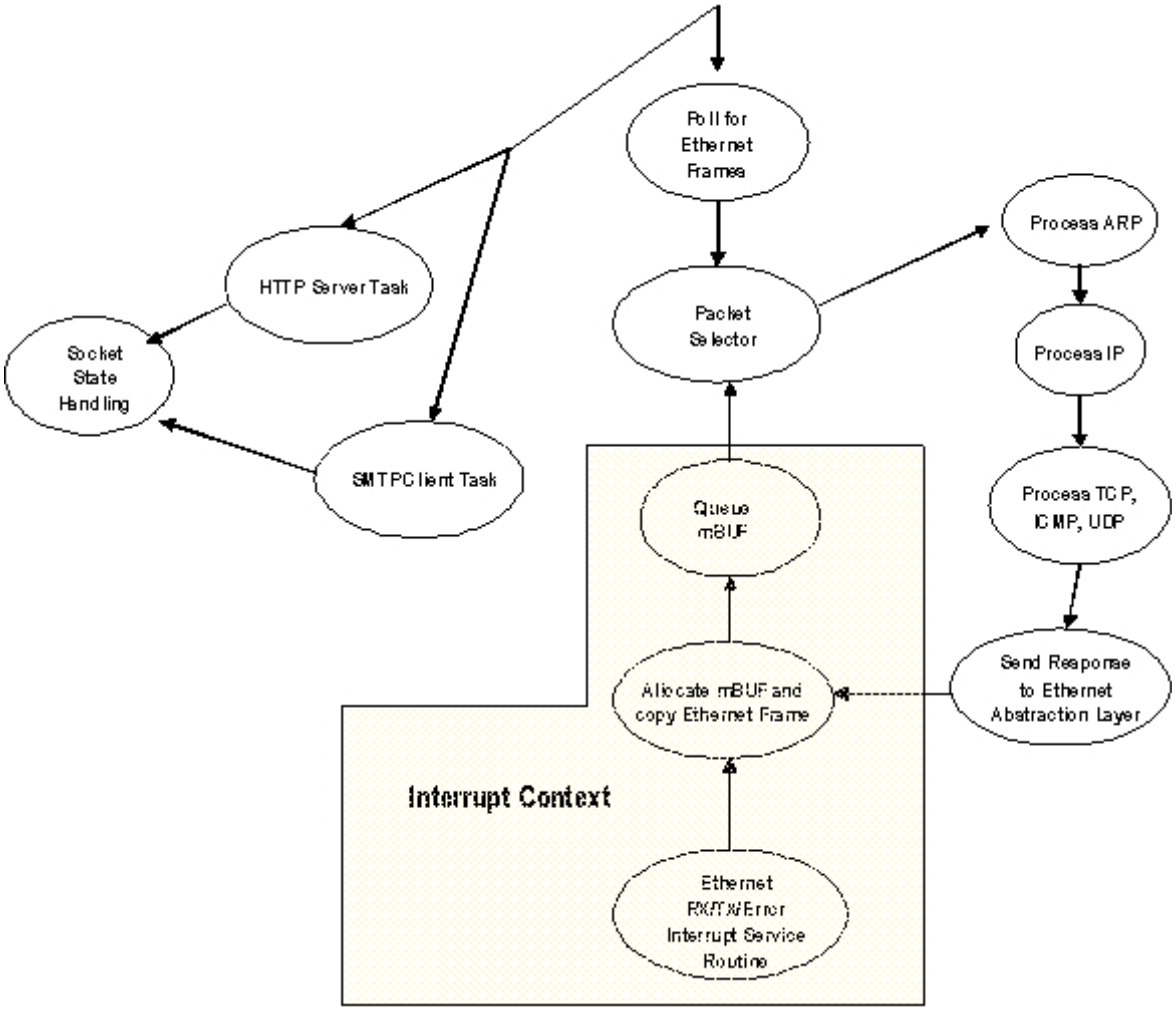


Figure 8-8. Main Application Diagram

8.5.1 Stack Loop

During the infinite loop, the software will execute the steps outlined in **Figure 8-9**.

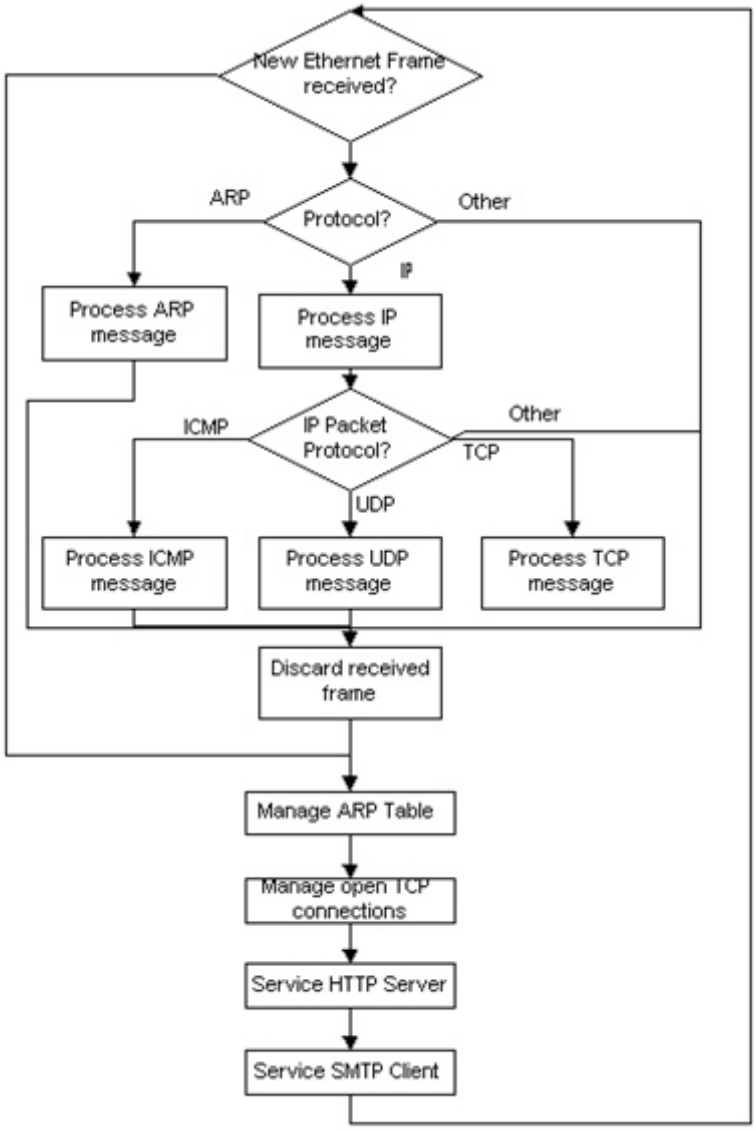


Figure 8-9. Main Loop Diagram

The defines in **Table 8-10** are required for the main loop.

Table 8-10. Defines Used by Main Connectivity Loop

#define NETWORK_CHECK_IF_RECEIVED() ETH_Receive()	
Description:	Invoke this macro periodically to check if there is new data in the Ethernet controller
#define NETWORK_CHECK_IF_RECEIVED() ETH_Receive()	
Description:	Invoke this macro when the received Ethernet packet is no longer needed and can be discarded

Routines in **Table 8-11** are required for the main loop.

Table 8-11. Functions Used by Main Connectivity Loop

UINT8 process_arp (struct ethernet_frame* frame)	
Description:	Invokes the *process_arp* function when the ARP packet is received
INT16 process_ip_in (struct ethernet_frame* frame)	
Description:	Processes the IP packet received by checking necessary header information and storing it according to the *received_ip_packet variable*
INT16 process_icmp_in (struct ip_frame* frame, UINT16 len)	
Description:	Invokes the *process_icmp_in* when the IP datagram containing an ICMP message is detected This function checks the accuracy of and ICMP message received and sends ICMP replies when requested
INT16 process_udp_in(struct ip_frame* frame, UINT16 len)	
Description:	Invoke this function to process UDP frames received
INT16 process_tcp_in (struct ip_frame* frame, UINT16 len)	
Description:	Invoke this function to process TCP frames received
void arp_manage (void)	
Description:	Iterates through ARP cache aging entries If a timed-out entry is found, removes it (dynamic address) or updates it (static address)
void tcp_poll (void)	
Description:	Checks all TCP sockets and performs various actions if time-outs occur The type of action is performed is defined by the state of the TCP socket
void https_run (void)	
Description:	This function is the main “thread” of the HTTP server program and should be called periodically from the main loop
void smtpc_run (void)	
Description:	This function is the main “thread” of the SMTP client program and should be called periodically when the SMTP client is active It is responsible for sending commands and data to the SMTP server and making callbacks to the user function stubs

8.6 Interrupt Handlers

The connectivity software requires one interrupt from the Ethernet Controller. This interrupt is located in IRQA.

Table 8-12. Interrupts Used by Connectivity Software

void smc_isr (void)	
Description:	This interrupt can be called with any of the following sources: • Reception: A packet was received • Transmition Complete: When at least one packet transmission was completed or one of the following errors occurs: —- Transmit Underrun —- SQE Error —- Lost Carrier —- Late Collision —- 16 Collisions • Transmition Empty: Set if TX FIFO goes empty • Allocate: When TX ram pages are allocated correctly • Receiver Overrun: Receiver aborts due to an overrun caused by a failed memory allocation, a packet is greater than 2Kb, or if a message was discarded • Ethernet Protocol Handler: Set when the Ethernet Protocol Handler detects a special condition (LINK OK, Counter ROLL Over, etc.) • Early Receive Interrupt: Set when a packet is being received and the number of bytes received exceeds a threshold • Physical Error: Set when one of several physical layer issues occur (LINK FAIL, Loss of Synchronization, CodeWord Error, etc.)

9. Results

9.1 Inverter Performance

The following parameters were measured using a Fluke 43B Power Quality Analyzer:

- RMS Amplitude
- Real Power
- Reactive Power
- Total Power
- Power Factor
- Harmonic Distortion
- Current Crest Factor

The following load types were connected at the inverter output for these measures:

- Linear Load: 110Ω
- Nonlinear Load: A full-wave rectifier followed by 220µF capacitor and 330Ω in parallel
- Full load: The linear load in parallel with the nonlinear load
  The full load measurement was made under online (AC Main) and offline (battery operation) conditions.

9.1.1 Inverter Waveforms Under No Load Conditions

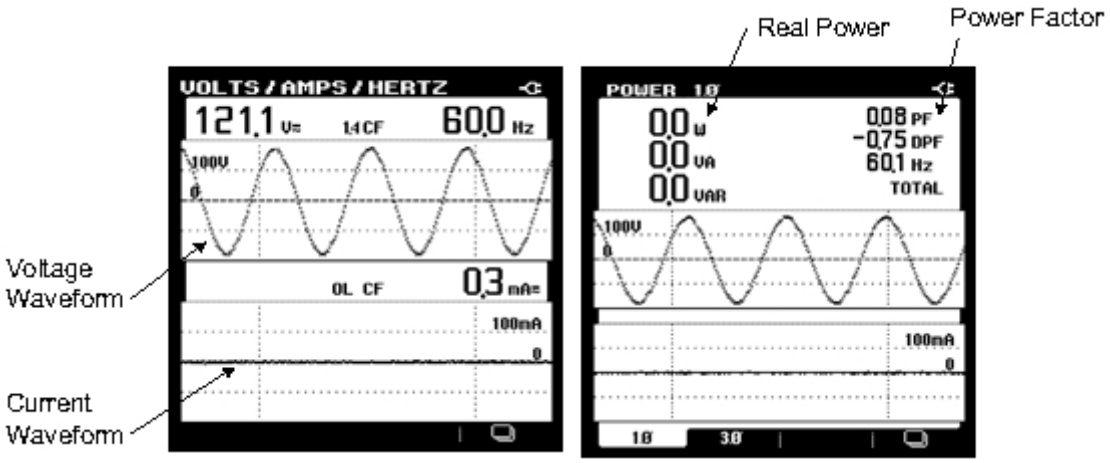


Figure 9-1. Inverter Waveforms—No Load

9.1.2 Inverter Voltage Harmonics Under No Load Conditions

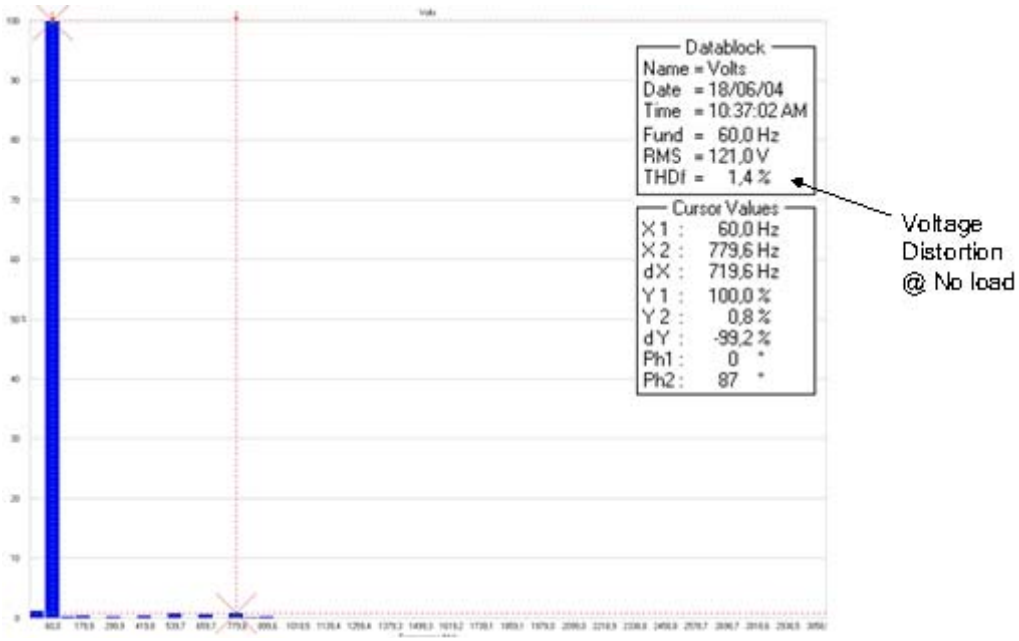


Figure 9-2. Inverter Voltage Harmonics—No Load

9.1.3 Inverter Waveforms Under Linear Load Conditions

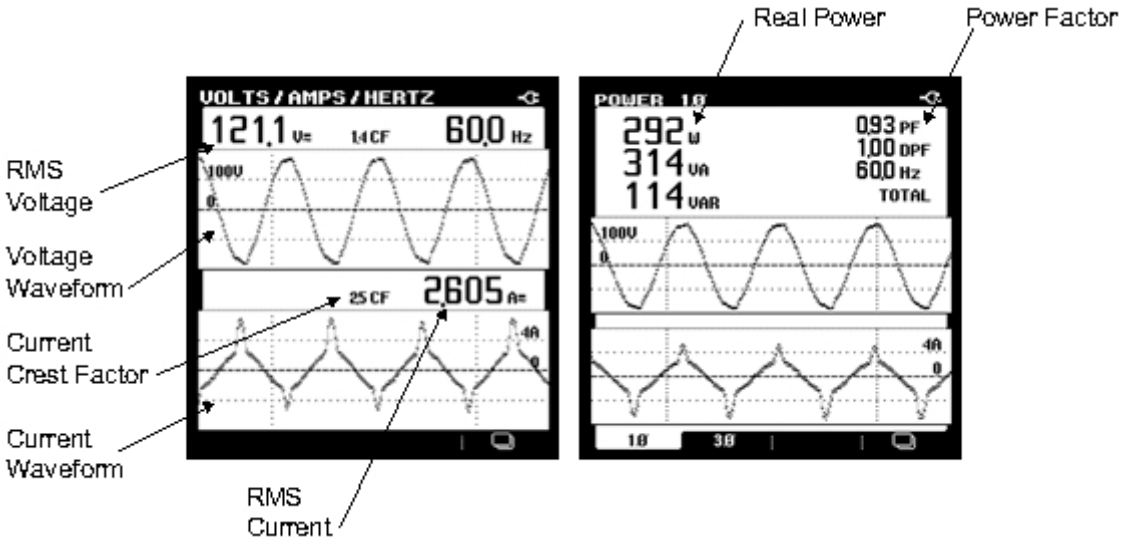


Figure 9-3. Inverter Waveforms—Linear Load

9.1.4 Inverter Voltage Harmonics Under No Load Conditions

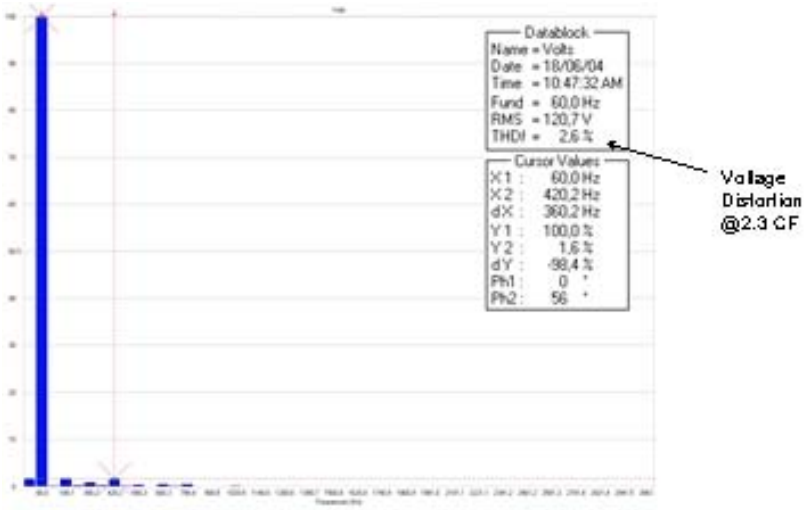
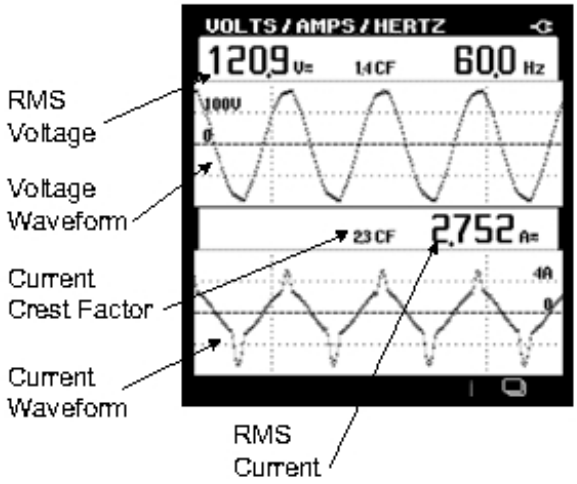


Figure 9-4. Inverter Voltage Harmonics Under Linear Load Conditions

9.1.5 Inverter Waveforms Under Full Load Conditions—Battery Operated



All other inverter waveforms are the same as when the rails are fed by the PFC instead of battery booster.

Figure 9-5. Inverter Waveforms Under Full Load Conditions—Battery Operated

9.1.6 Inverter Voltage Harmonics Under Full Load Conditions—Battery Operated

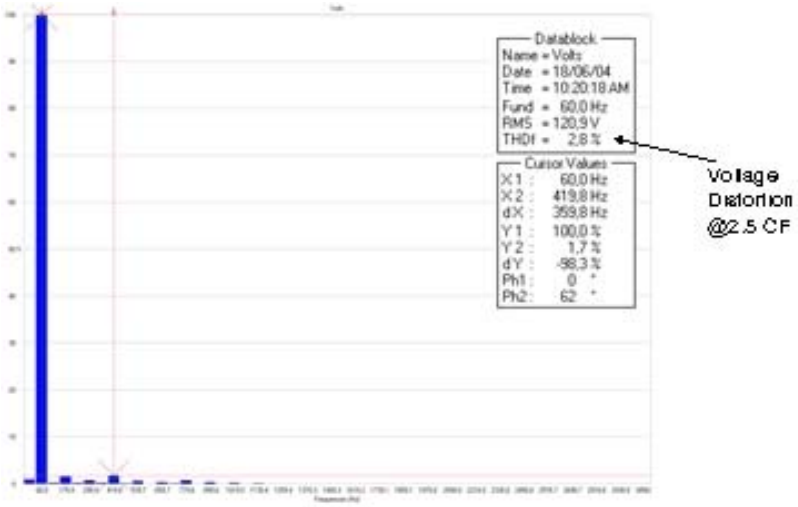


Figure 9-6. Inverter Voltage Harmonics—Full Load, Battery Operated

9.1.7 Inverter Transient Response—60W Cold Bulb

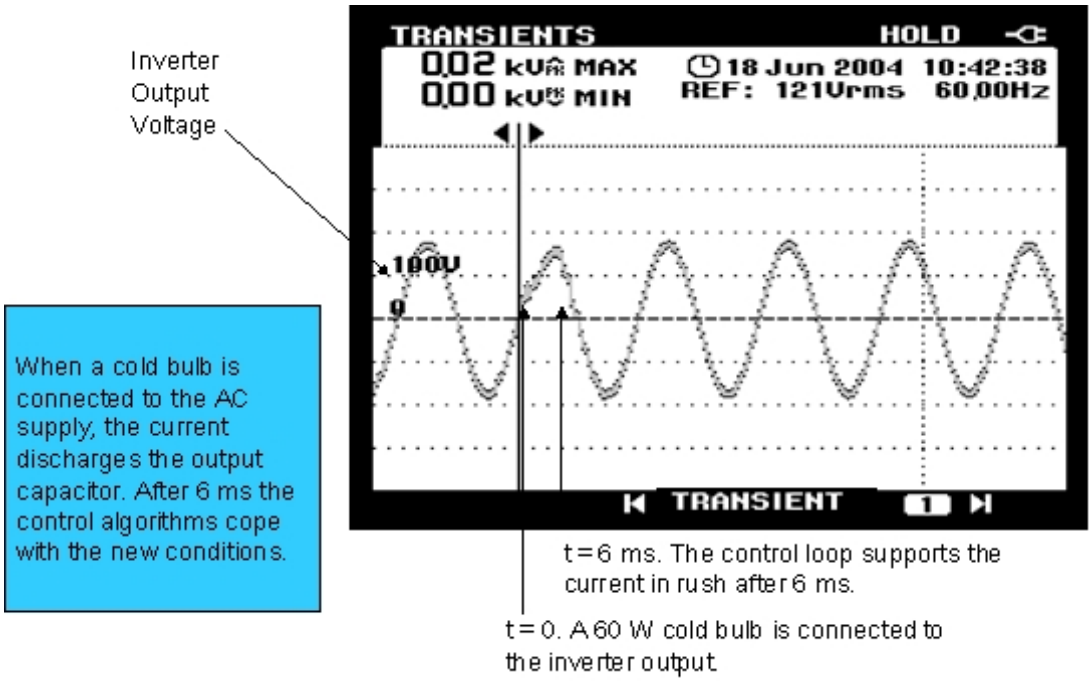


Figure 9-7. Inverter Transient Response—60W Bulb

9.1.8 Inverter Transient Response to a Resistive Load (200W)

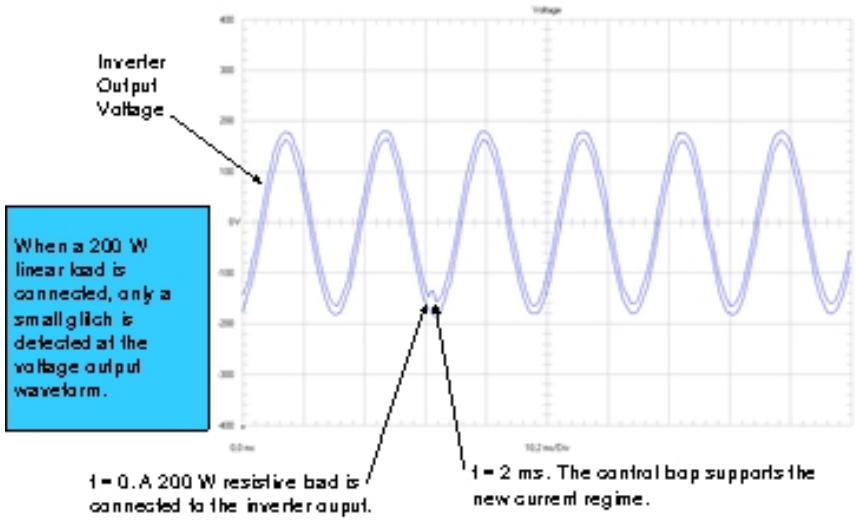


Figure 9-8. Inverter Transient Response—Resistive Load (200W)

9.2 PFC Performance Measurements

The following parameters were measured using a Fluke 43B Power Quality Analyzer:

- RMS Amplitude
- Real Power
- Reactive Power
- Total Power
- Power Factor
- Harmonic Distortion

Loads connected to the PFC for measurements:

- Inverter with full load plus battery charger
- Inverter with linear load plus battery charger
- Inverter with no load plus battery charger

For all measurements, the inverter was phase and frequency locked to the AC main line (60Hz).

9.2.1 PFC Performance Under Full Load Conditions

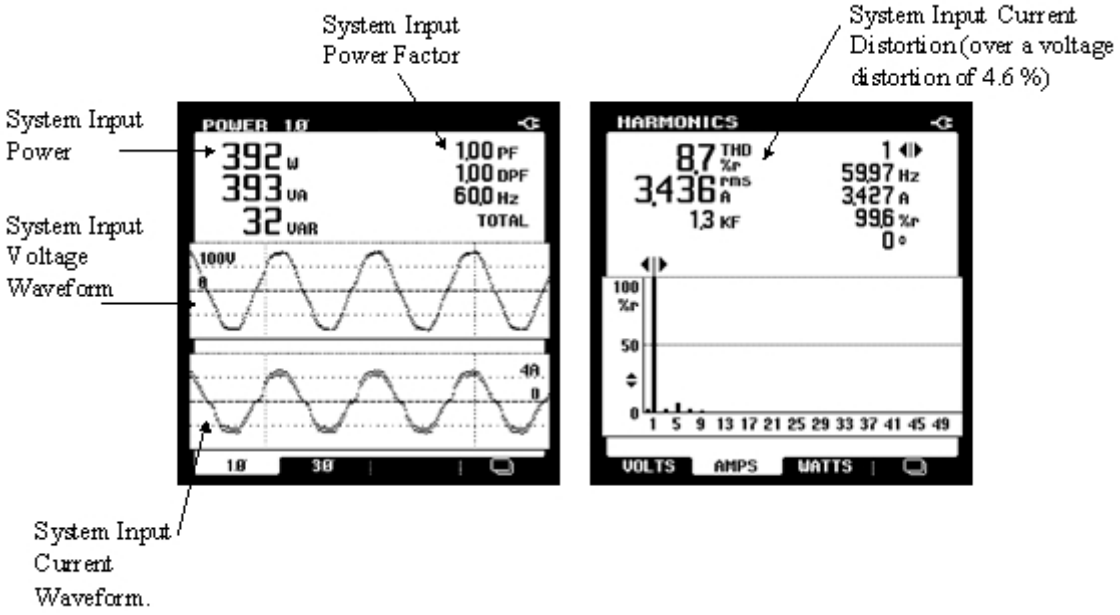


Figure 9-9. PFC Waveforms—Full Load Conditions

9.2.2 PFC Performance at 298W Linear Load Conditions

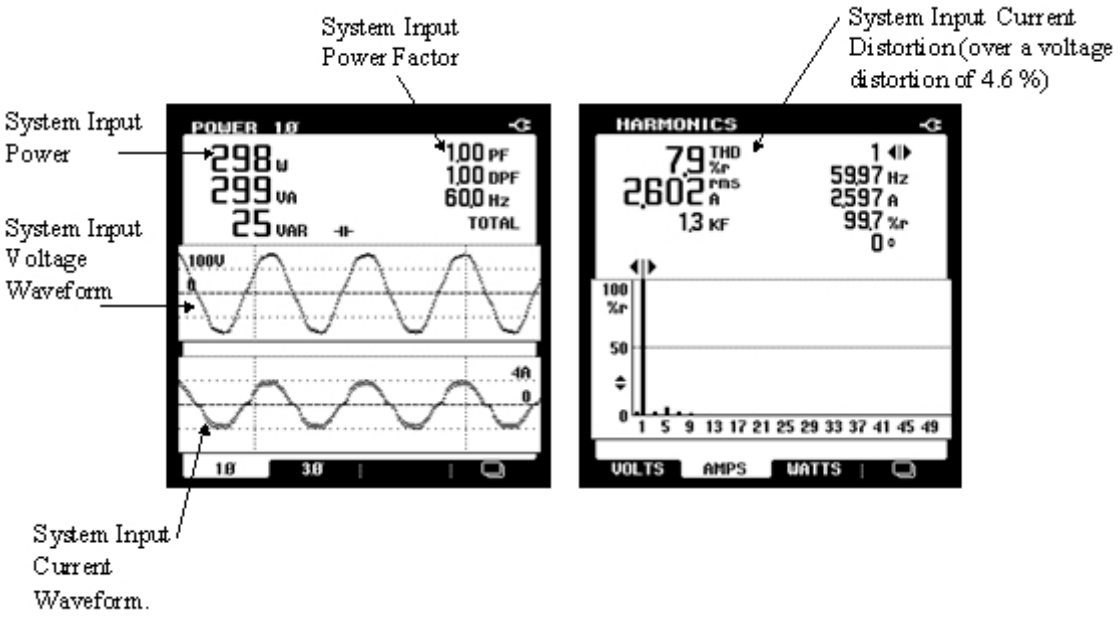


Figure 9-10. PFC Waveforms—Linear Load Conditions (298W)

9.2.3 PFC Performance at 59W Linear Load Conditions

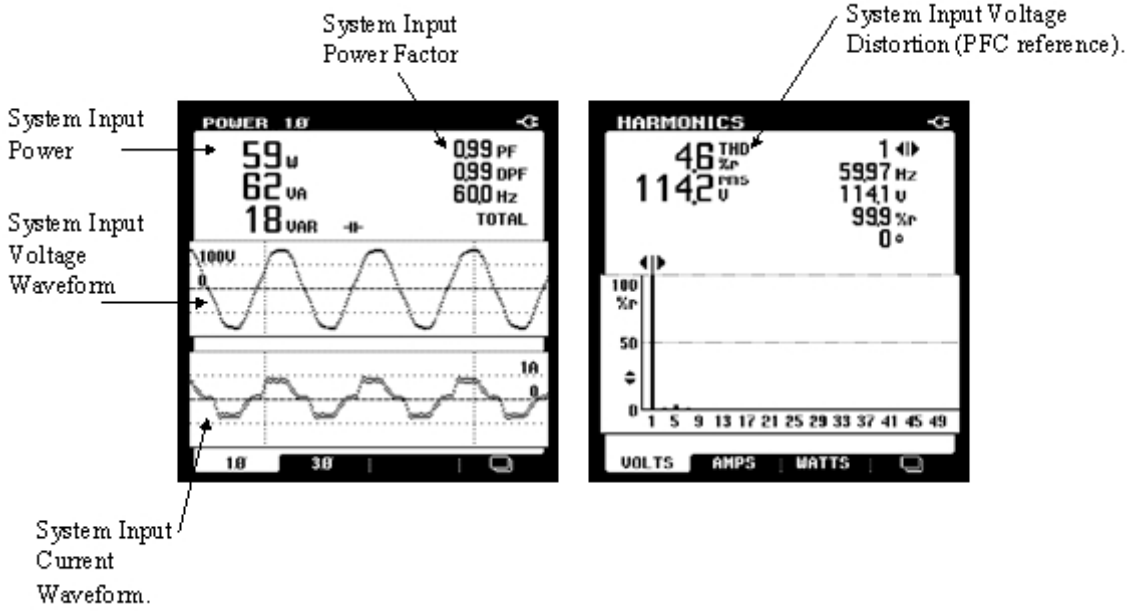


Figure 9-11. PFC Waveforms—Linear Load Conditions (59W)

9.2.4 PFC—Transient Response to a Load Step

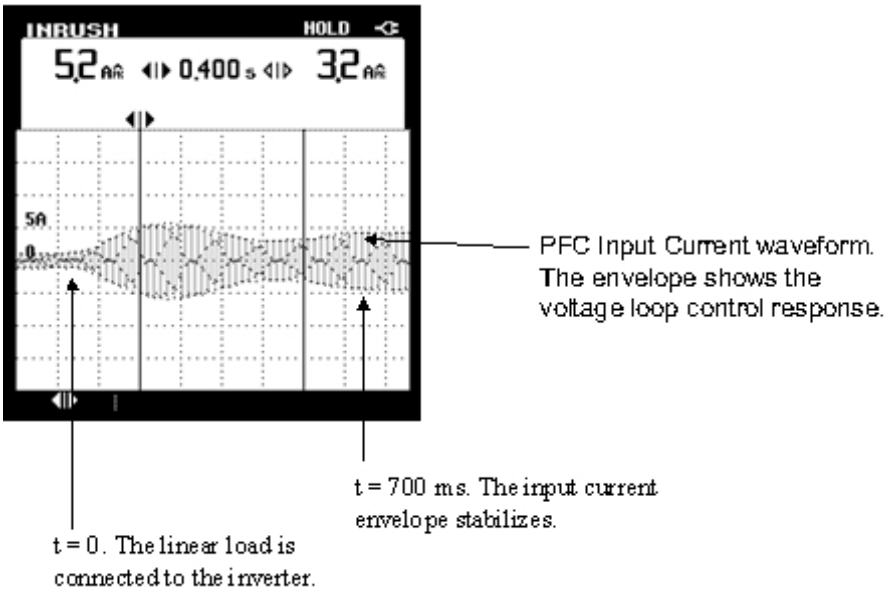


Figure 9-12. PFC Performance—Transient

9.3 Frequence Performance

Freerun Output Frequency: 60.0105Hz

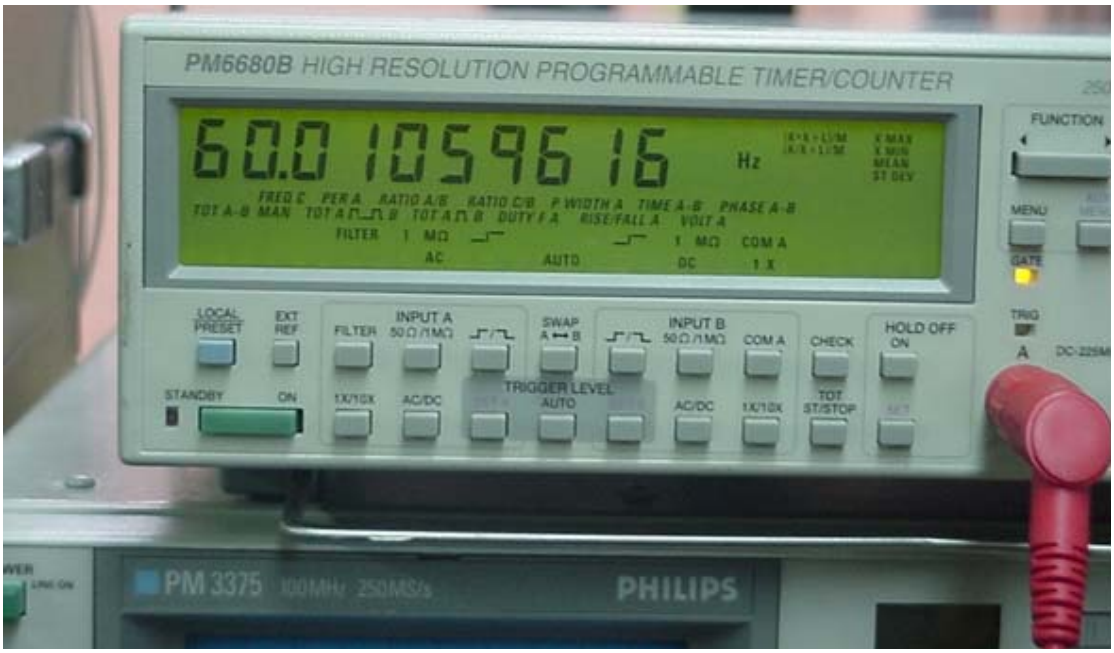


Figure 9-13. Freerun Frequency Measurement

9.4 Battery Charger Performance

9.4.1 Battery Charger Performance, Current and Voltage Waveforms

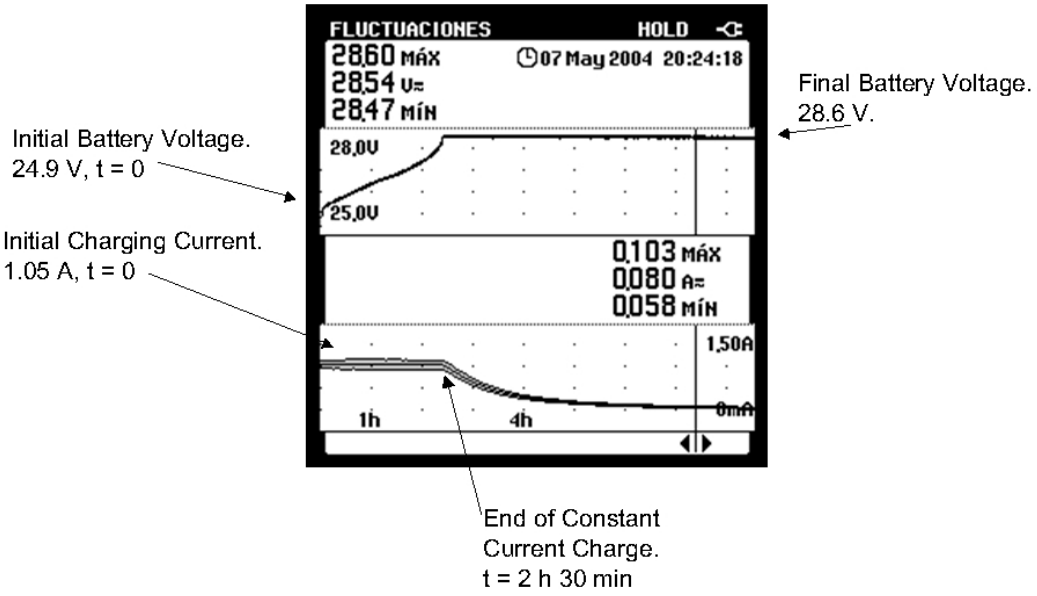


Figure 9-14. Battery Charger Performanc—Charging

9.4.2 Battery Charger—Floating Conditions

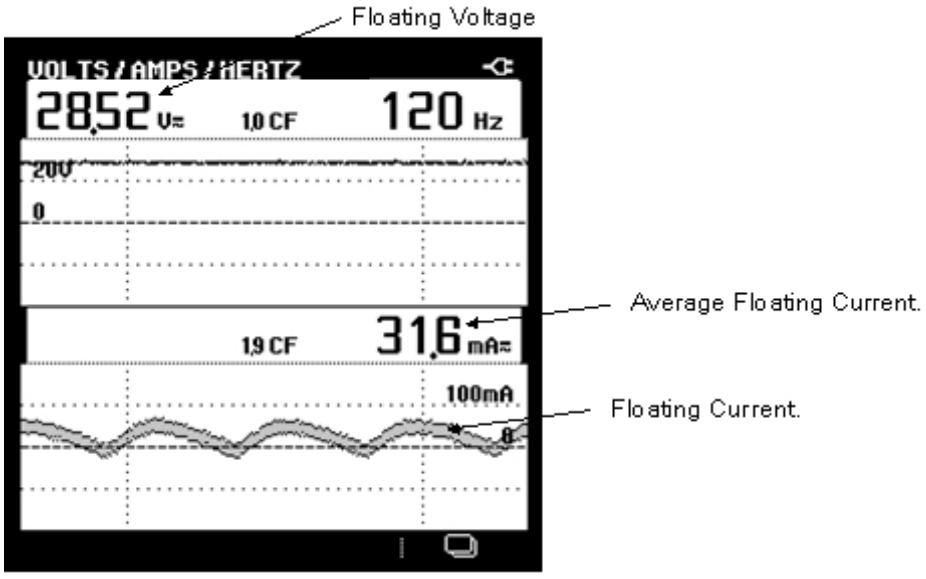


Figure 9-15. Battery Charger—Floating

9.5 Bypass Switch Performance

9.5.1 Inverter Bypass Switch Operation

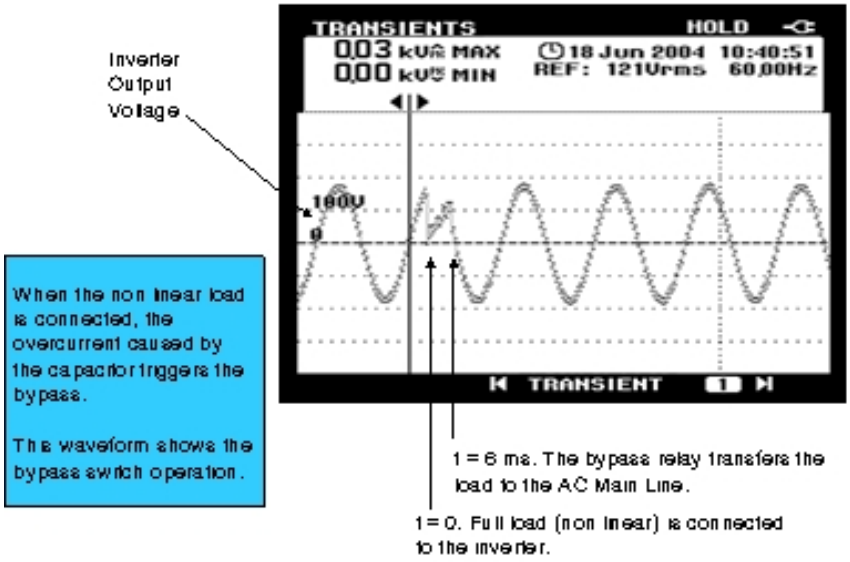


Figure 9-16. Bypass Switching Time

10. References

1. *56F8300 Peripheral User Manual*, MC56F8300UM
2. *56F8346 Data Sheet Preliminary Technical Data*, MC56F8346

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify that you have the latest information available, refer to **www.freescale.com** or contact your Freescale representative.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN3113
Rev. 1
07/2005