AVR465: Single-Phase Power/Energy Meter with Tamper Detection

Features

- **Cost-Effective and Flexible Single-Phase Energy Meter**
- **Fulfills IEC 61036 Accuracy Requirements for Class 1 Meters**
- **Detects, Signals and Continues to Measure Accurately Under At Least 20 Different Tamper Conditions**
- **Design Easily Downgrades to Fit Normal Single-Phase Energy Metering**
- **Compact Design With Internal Flash Memory, SRAM and EEPROM**
- **Includes USART and Programmable I/O**
- **LCD Is Easily Added By Migrating To Other AVR Microcontrollers**
- **Secure and Reprogrammable Flash Memory Enables Flexible Firmware Updates**
- **One-Time, Quick, and Accurate Digital Calibration Gives Added Benefits**
  - **Calibration Can Be Automated**
  - **No Need for Trimming Components**
  - **No Need for External EEPROM, as Calibration Coefficients Are Stored Internally**
- **Adjustable Energy Pulse Output Goes Beyond 10.000 Impulses / kWh**
- **Active Power, Voltage and Current Measurements are Easily Accessible Over USART Interface**
- **Design Easily Migrates to Any Other AVR Microcontroller**
- **Low-Power AVR Microcontroller Allows Operation Down To 1.8V**

Introduction

This application note describes a single-phase power/energy meter with tamper logic. The design measures active power, voltage, and current in a single-phase distribution environment. It differs from ordinary single-phase meters in that it uses two current transducers to measure active power in both live and neutral wires. This enables the meter to detect, signal, and continue to measure reliably even when subject to external attempts of tampering.

The heart of the meter is an AVR microcontroller. All measurements are carried out in the digital domain and measurement results are available in the form of frequency-modulated pulse outputs and as plain-text values, accessible over the USART interface. This enables the design to be used in cost-effective applications based on mechanical display counters. Alternatively, the design easily fits more computerized applications with features such as remote reading (AMR), demand recording, multiple tariffs, and other.

A prototype built for 230V and 10A operation showed better than 1% accuracy over a dynamic range of 500:1. With careful PCB design and following the guidelines given at the end of this document the accuracy can be further increased. The meter is easily configured to fit any other voltage and current settings.

Overview

Power meters are sometimes referred to as energy meters and vice versa. Per definition, (active) power is a measure of what is required (or consumed) in order to perform useful work. For example, a light bulb with a 100W rating consumes 100 watts of active power in order to create light (and heat). Energy, per definition, is the measure of how much work has been required over a known period of time. In the light bulb example, leaving the bulb on for an hour it will consume 100W * 3600s = 360000Ws (watt-seconds) = 100Wh (watt-hours) = 0.1kWh (kilowatt-hours) of energy.

The meter described in this application note can be referred to as a power meter, an energy meter or a kilowatt-hour meter. The Energy Pulse output (EP) is a ready indication of active power, as registered by the meter; the frequency of the pulse is directly proportional to active power. Integrating pulses over time gives active energy. For storage purposes, the meter includes two pulse outputs (DPP and DPN) to directly drive various display counters. All pulse outputs are easy to configure for any reasonable rate. The default is 10.000 impulses per kilowatt-hour for the EP output and 100 impulses per kilowatt-hour for the DPP/DPN pulses.

Not only pulse rates are readily adjusted; All measurement results can be calibrated in the digital domain, removing the need for any trimming components. This includes adjustments to compensate for phase delays in current transformers. The calibration event can be automated, removing the time-consuming manual trimming required in traditional, electromechanical meters. Digital calibration is fast and efficient, reducing the overall production time and cost. Calibration coefficients are safely stored in the internal EEPROM, further reducing the need for external devices.

The brain of the meter is the firmware, which is provided open source. Although it includes all the functionality required for a tamper-proof, single-phase meter, it can be modified and updated at any time. Even in the field. The firmware is entirely written in C, which makes modifications easy to implement. Integrity and intellectual property are yet easy to secure using Lock Bits of the AVR microcontroller.

Meter Hardware

The energy meter hardware consists of a power supply, an analog front end, a microcontroller section, and an interface section.

Warning: Shock Hazard

The meter described in this application note connects to high currents and high voltages. High currents and high voltages may be hazardous, even lethal. Hence, the meter should be operated by qualified technicians only. Atmel takes no responsibility for any consequence that may result from the use of this document or the application described herein.

Figure 1. Electric Shock Hazard.
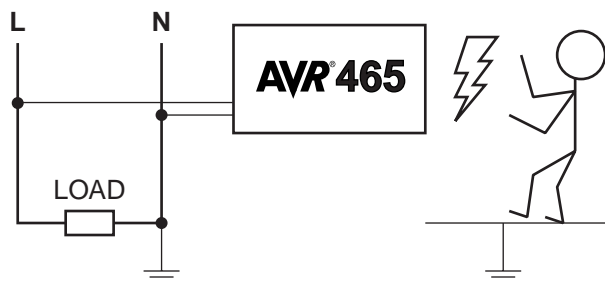
There is no insulation between line voltage and the meter. Hence, sections of the meter contain high voltages and even the low-voltage output of the power supply is connected to the mains. Use caution. The meter must be enclosed in a non-conductive casing to avoid accidental voltage shocks.

Power Supply

The power supply is a low-efficiency, but cost-effective and compact design. It is intended to provide just the amount of power needed by the meter at a cost as low as possible. If enhancements are made to the meter it may be necessary to derive new values for some of the components.

Power Budget

The following table summarizes typical current consumption of the main parts of the meter.

Table 1. Typical Current Consumption of Main Meter Sections.

| Section	Includes	Continuous	Peak
Front End	Operational amplifiers [1]	0.2 mA	0.2 mA
Microcontroller	AVR (Active Mode, 4 MHz) [2]	1.8 mA	3.5 mA
LED	All LED's (each about 1 mcd brightness)	None [3]	3.0 mA
Display	Display counter (400 Ω coil impedance)	None [3]	7.5 mA

Notes:
1. Typical consumption, according to LMV358 data sheet
2. According to ATmega88 data sheet (2545B–AVR–01/04)
3. LED's are mostly off and display counter is updated rarely

Peak currents are brief, mainly occurring when the display counter is updated. The worst-case scenario is when the display counter is updated and all LED's are lit. Typically, the power supply needs to be able to supply less than ten milliwatts (at 3 volts), but it must also be able to deliver the brief bursts of energy required to turn the coil of the counter.

Theory

The power supply is illustrated in Figure 2. The below schematic is identical to the power supply section of the meter, as illustrated in the schematic at the end of this document, but component numbers are not the same. Please note well the galvanic connection between live wires and meter ground!

Figure 2. Low-Cost, Transformerless Power Supply.
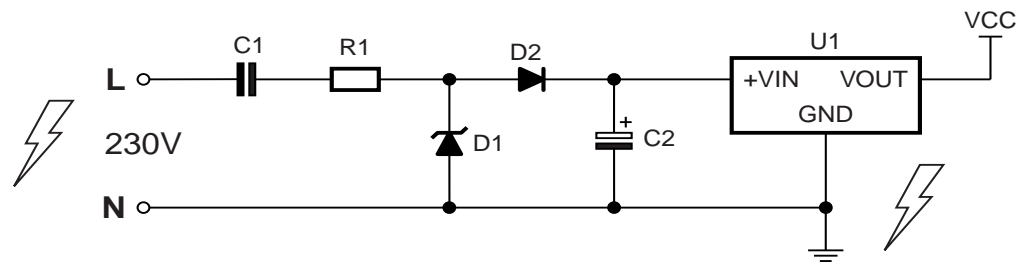
The power supply is based on halfway rectification. During negative half-waves capacitor C1 is charged and during positive half-waves the capacitor is drained. Zener diode D1 (minus the forward voltage of diode D2) dictates to which voltage C2 is charged. Voltage regulator U1 uses the energy stored in C2 to produce a stable output voltage. Resistor R1 controls the charge and discharge of C1 and also limits the current flow through zener diode D1.

Source of Noise

Please note that the power supply can be a source of noise, if poorly planned and realized. Ground connections are very important. For example, the current flow through the zener is rather large and if the same ground track is shared by the zener and, say, the operational amplifiers (see current front end) then measurement results will be greatly degraded. Typically, this can be seen as odd harmonics in the measured current signal.

It is recommended to use star topology in ground connections.

Component Values

The dropout voltage for the 3.3V regulator U1 is about 5V. When input voltage falls below the dropout level, the device ceases to regulate. The regulator input must be kept above this level, even at the end of the drain cycle and at worst-case current consumption.

As a starting point, the zener diode is specified to 15V. This leaves much headroom for capacitor C2 to discharge before reaching minimum input voltage of the regulator.

Next, the size of capacitor C2 is calculated. The minimum size is derived based on the general discharge function of the capacitor, as follows:

Equation 1. Equation To Calculate Size Of Charge Capacitor.

$$V = V_0 \times e^{-\frac{t}{RC}} \quad \Rightarrow \quad C = -\frac{t}{R \times \ln\left(\frac{V}{V_0}\right)}$$

Here t is the discharge time, V_0 is the initial voltage, V is the voltage after discharge and R is the load discharging the capacitor. If the worst-case current consumption is 14mA (see previous table), then the equivalent load resistance is R = 3V/14mA = 214 Ω. Worst-case current consumption takes place when driving the display counter. The length of the drive pulse is 100ms, by default. Assuming the voltage of the charged capacitor is allowed to drop to regulator minimum during the length of one display pulse, the smallest size of the capacitor is as follows:

Equation 2. Calculating Minimum Size of Capacitor C2.

$$C = -\frac{0.1s}{214\Omega \times \ln\left(\frac{5V}{15V}\right)} = 425.3\,\mu F \approx 470\,\mu F$$

Next, capacitor C1 is calculated. The size of the capacitor should be as small as possible, since it dictates how much power is drawn from the mains lines. Also, the larger the capacitor, the more expensive it is. The minimum size of the capacitor is derived from the basic functions of stored charge (Q = CU) and current (I = Q/t). For capacitor C1 it is no longer required to use the above worst-case current (14mA), since capacitor C2 will store energy enough to maintain the current briefly. Assuming 10mA continuous current, and that the capacitor is drained over one 50Hz half cycle, and that voltage is 80% of nominal, then the required minimum size of the capacitor is as follows:

Equation 3. Calculating Minimum Size of Capacitor C1.

$$C = \frac{I \times t}{U_{MAINS}} = \frac{0.01A \times 0.01s}{0.8 \times 230V} = 0.543\,\mu F \approx 680nF$$

The capacitor needs to be fully charged each half-cycle. The charge time is dictated by resistor R1, the size of which can be derived using the so-called 5RC rule of thumb. The 5RC rule says that for a step change in voltage the capacitor charges to within 1% of its final value in five time constants (RC). Specifying that the capacitor should be (almost) fully charged at the peak of the positive half-wave, the maximum size of resistor R1 can be estimated as follows:

Equation 4. Calculating Maximum Size of Charge Limit Resistor.

$$t = 5 \times R \times C \quad \Rightarrow \quad R = \frac{t}{5 \times C} = \frac{0.005s}{5 \times 680nF} = 1470\Omega$$

Another limitation on resistor R1 is that it must be small enough for capacitor C2 to charge enough during one half-cycle. The larger R1 is, the less C2 is charged each cycle. On the other hand, it is unreasonable to specify R1 such that C2 charges to, say, 99% during one half-cycle since this would make R1 very small and the power consumption in zener diode D1 very large. Instead, a decent charge level is selected and R1 is specified accordingly. For example, setting R1 = 470 Ω the meter works nicely (input voltage to regulator typically stays above 13V at all times).

Analog Front End

The analog front end is the part, which interfaces to the high voltage lines. It conditions high voltages and high currents down to a level where the signals cannot harm the more delicate electronics. It converts high voltages and high currents to voltages sufficiently small to be measured directly by the ADC of the microcontroller.

The nominal line voltage of the meter is 230V and the maximum rated current is 10A, both of which obviously are way too large signals to be fed directly to any microcontroller. The analog front end converts line voltage and line current to voltages with amplitudes of no more than 1V peak-to-peak. The front end is easy to configure for any other line voltage or current, as described in the following.

Voltage Front End

Line voltage is first downsized using a resistor ladder, then DC-filtered and finally DC-biased, as illustrated in Figure 3. Note that component numbers are not the same as in the full schematic at the end of this document.

Figure 3. Voltage Front End.
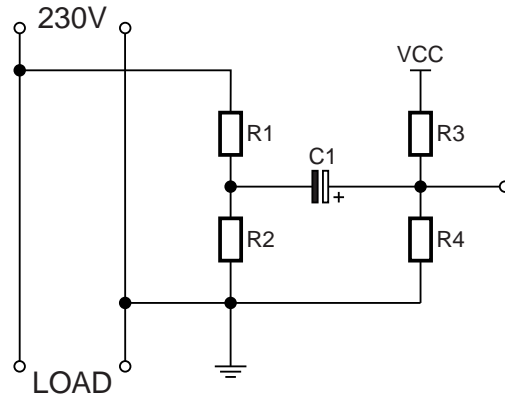
The resistor ladder R1-R2 by default produces a 1.1Vpp signal when the line voltage reaches 115% of nominal voltage, as follows:

Equation 5. Downsizing The Line Voltage.

$$U_{MAX} = U_{NOM} \times 1.15 \times \frac{R2}{R1+R2} = \frac{230V \times 1.15 \times 1k\Omega}{681k\Omega} = 0.388V = 1.099V_{PP}$$

The nominal voltage is 230V by default and the 265V limit is there to leave 15% headroom for overvoltages.

The DC bias ladder R3-R4 positions the AC signal halfway up the ADC voltage reference. By default, it has been sized to fit ATmega88 (1.1V reference) and 3V supply voltage, as follows:

Equation 6. DC Bias Level Of Downsized Signal.

$$U_{DC} = U_{VCC} \times \frac{R4}{R3+R4} = \frac{3V \times 680k\Omega}{3.68M\Omega} = 0.55V \approx \frac{U_{AREF}}{2}$$

Crosstalk

Please note that the voltage front end handles voltages of considerable amplitude, which makes it a potential source of noise. Disturbances are readily emitted into current measurement circuitry, where it will interfere with the actual signal to be measured. Typically, this shows as a non-linear error at small signal amplitudes and non-unity power factors. At unity power factor, voltage and current signals are in phase and crosstalk between voltage and current channels merely appears as a gain error, which can be calibrated. When voltage and current are not in phase crosstalk will have a non-linear effect on the measurements, which cannot be calibrated.

Crosstalk is minimized by means of good PCB planning and the proper use of filter components.

Current Front End

The current front end is a little bit more complex than the voltage front end. This is because line voltage remains constant at, say, 230V but line current varies with the load. Line current typically ranges from some milliamperes to ten amperes, or more. In order to achieve 1% measurement accuracy over such a wide range, the ADC would need to have a resolution of around 16 bits. Since the target device includes only a 10-bit A/D-converter the front end must amplify small-scale signals. The current front end therefore includes a programmable gain stage, which is controlled by the MCU.

The design criteria for the programmable gain stage are not very relaxed; the gain stage must amplify AC signals up to around 100x, but provide little or no DC amplification. This is because the input is a DC-biased AC signal and if the gain stage provides even a small DC amplification the output will saturate. In addition, the gain must be programmable by the MCU and the settling time must be considerably less than a second. Finally, the design must be cost-effective.

There are many school examples of how to realize the above, but most of them are sooner or later ruled out by at least one of the design criteria. A good starting point, however, is the operational amplifier; they are common, exist in a wide variety and can be very cost-effective.

A little experimentation soon shows that the non-inverting amplifier is not a viable topology for this design, mostly because of the requirements for high AC and low DC gains. Considering the frequency band of interest, AC-coupled, non-inverting amplifiers require very large (and expensive) capacitors for the DC decoupling. Also, a large DC decoupling capacitor leads to very long switching times when gain is altered. Since the gain needs to be variable the DC levels cannot be trimmed to zero.

A viable solution is found from inverting amplifier topology, although it still requires a rather (but not very) large capacitor to be used. Gain configuration resistors are readily toggled in and out using low-cost switches from 74HC-series logic, as shown in Figure 4. The gain stage shown has a fast switching time and allows high AC gain but a low DC gain.

Figure 4. Inverting Amplifier with Variable Gain Uses Bilateral Switches (74HC4066).
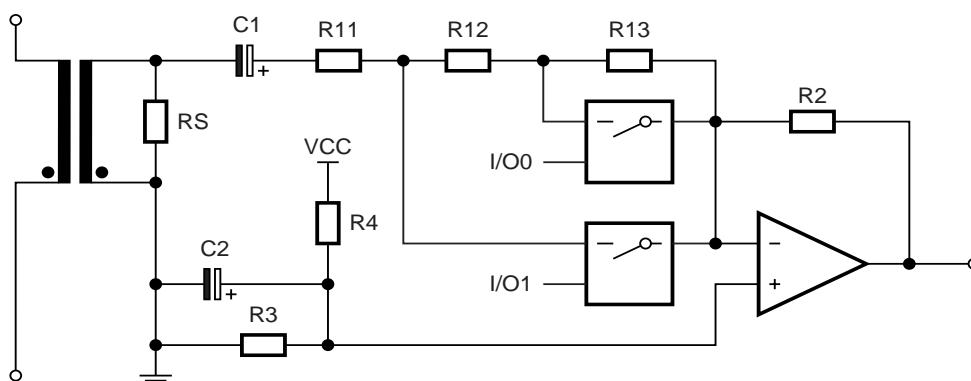
Calculating the Gain

The gain of the inverting amplifier is as follows:

Equation 7. Gain Of Inverting Amplifier.

$$A = -\frac{R2}{R1}$$

Here R1 consists of the series connection of R11, R12 and R13. Gain is adjusted by shorting out one of resistors R12 or R13. This is done using the bilateral switches, which are controlled by two I/O pins of the MCU, shown as I/O0 and I/O1 in the figure.

Gain adjustment resistors are dimensioned such that each range has an amplification of about eight times the previous. The number of switches may well be increased and the gain difference decreased, however, it is not recommended to have a larger gain difference than eight between two subsequent ranges. This is because gain

differences of around ten, and higher, cause the signal to degrade below 1% accuracy before it can be further amplified.

The table below illustrates how gain is adjusted from the MCU.

Table 2. Adjusting The Gain Of The Inverting Amplifier. R2 = 470 kΩ.

I/O0	I/O1	Range	R₁ Impedance	Gain
Low	Low	Low	R11+R12+R13 = 6.8k+39k+330k [2]	-(470/375.8) = -1.25
Low	High	Medium	R11 + R12 = 6.8k+39k	-(470/45.8) = -10.26
High	X [1]	High	R11 = 6.8k	-(470/6.8) = -69.11

Notes: 1. No effect, when I/O0 is high

2. All switches open; all resistors connected in series

Calculating Full-Scale

Shunt resistor RS and the current transformer are scaled such that a voltage signal of no more than 1V peak-to-peak is present at the amplifier output when maximum current flows through the primary of the current transformer and the amplifier is set to minimum gain. For example, using a 2500:1 current transformer, a 68 Ω resistor, and setting amplification to minimum (see above), the voltage at the output is as follows:

Equation 8. Full-Scale Voltage At Amplifier Output.

$$U_{MAX} = \frac{A \times RS \times I_{MAX}}{N} = \frac{1.25 \times 68\Omega \times 10A_{RMS}}{2500} = 0.34V_{RMS} \approx 0.96V_{PP}$$

DC Decoupling

The input signal of the amplifier must be DC decoupled. This is done using capacitor C1 in the previous figure. It should be noted that R1 + C1 form a high-pass filter (HPF) that may distort signals, especially at high gains. The corner frequency, or the –3dB point, of the HPF is calculated as follows:

Equation 9. Corner Frequency Of HPF.

$$f_{-3dB} = \frac{1}{2 \times \pi \times R1 \times C1}$$

The closer the corner frequency of the HPF is to the frequency band of the interest, the higher is the distortion of phase and amplitude. It is recommended to keep a distance of at least 100x between the two frequencies. At 50Hz line frequency and R1 minimum = R11 = 6.8 kΩ, C1 should not be less than 47 µF.

DC Bias

The output signal of the gain stage is biased around the DC level present at the non-inverting input of the operational amplifier. Hence, this DC level should be exactly half of the reference voltage of the ADC in the MCU. Assuming a 1.1V reference, the voltage divider R3-R4 should produce a stable 0.55V. The voltage divider should have a large impedance to keep the current consumption low, since high impedance increases noise. For example, assuming 3V supply voltage, a suitable set of values is 3 MΩ + 560 kΩ.

The DC level is stabilized with one, or many, capacitors. It is recommended to include at least one capacitor in the µF-range.

Eliminating Noise

To make accurate measurements, the input signal must be as clean as possible, especially at low amplitudes. Input signals with low amplitude are amplified before being sampled and processed, which means any noise in the signal will be enlarged, too. At the low end of the measurement range input signals have amplitudes below

10mV, which means noise typically not visible on an oscilloscope (say, below 1mV) may distort the signal by as much as 10%.

Noise is eliminated using good PCB planning and properly sized and placed filter components.

Protection Circuitry

Energy meters are prone to operate in harsh environments; meters are often subject to over-voltages and current spikes. If such disturbances are not properly shielded they may traverse all the way to the MCU and drive it outside operating limits.

There are many techniques available for filtering out unwanted disturbances. Rapid voltage spikes are usually suppressed using varistors, which normally have very high impedance. At a given threshold the impedance very rapidly decreases, causing a short circuit, which then leads the excess energy to ground. Varistors must be picked such that they do not break down if the spike is too large or lasts too long (i.e. if the energy of the spike grows too large).

Much depending on the layout of the circuit board and connectors, some inputs may be prone to pick up radiated noise. In some cases, even the circuit board tracks can perform as antennas and pick up noise from the environment. Induced noise can be throttled using well-placed diodes, which clamp signals between ground and supply voltage.

EMC protection techniques are well covered in application note "AVR040: EMC Design Considerations".

Microcontroller Section

The microcontroller section includes the AVR and the clock source to run it. Also included are all digital input and output circuitry.

Signal LED's

In order to keep power consumption low all LED's are of low-current type. Low-current LED's typically produce around 2mcd of light power for 2mA current. The LED's have a forward voltage of about 2V and using a series resistor of 1 kΩ will therefore limit the current to about 1mA per LED. The resulting LED brightness is good for most applications.

Display Counter

For display purposes, this application uses what is sometimes referred to as a stepper motor counter. These types of counters are typically available in a price competitive range and tend to be rather common in designs like this. The counter is driven using a pulse waveform as illustrated in Figure 5.

Figure 5. Pulse Waveform To Drive Display Counter.
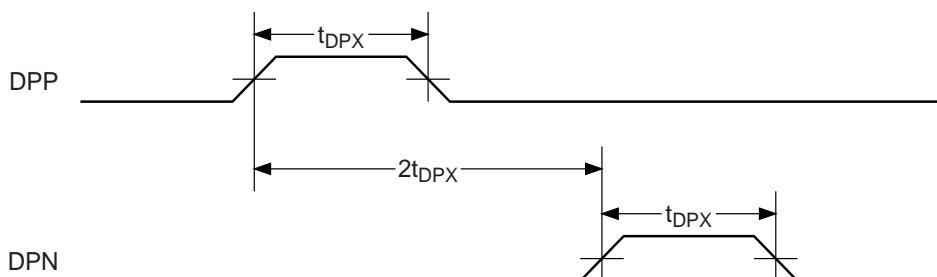
By default, pulse length t_{DPX} is 100ms, but this is readily adjusted from the firmware to any other value.

Naturally, the pulse outputs provided by the design can be used to drive standard impulse counters, as well.

Programming Interface

The programming interface is required for programming the microcontroller and for calibrating the meter. Using the serial programming interface (SPI) of the AVR, it is possible to access both Flash and EEPROM via the same connector. Complete system programming and calibration can be performed at any time.

The pin layout of the programming interface is shown in the table below.

Table 3. Pin Layout of SPI Connector

Pin	Name	Direction	Description
1	MISO	Out	Serial Data Out
2	VCC	N/A	Supply Voltage
3	SCK	In	Serial Clock
4	MOSI	In	Serial Data In
5	/RST	In	Device Reset
6	GND	N/A	Ground

Serial Interface

The application uses the USART (Universal Synchronous/Asynchronous Receiver/Transmitter) interface of the AVR to send measurement data. Any terminal software can be used for sending commands to the meter and for receiving measurement data. Measurement results, once calibrated, are given in units of watts, volts and amperes.

The following table shows the pin layout of the asynchronous interface connector.

Table 4. Pin Layout of Serial Interface

Pin	Name	Direction	Description
1	RxD	In	Asynchronous data in
2	TxD	Out	Asynchronous data out
3	GND	N/A	Ground
4	GND	N/A	Ground

Please note that the meter does not contain a level converter and the signals can therefore not be directly connected to the serial port of a computer. The RS-232 spare connector on STK500 contains a level converter and can therefore be used as a buffer between the meter and the computer.

Communication Noise

Asynchronous communication may introduce noise in measurement results, especially when power supply is unable to provide the amount of current required and when ground connections have not been properly laid out. By default, the power supply is capable of delivering enough current for the meter to run, but not necessarily enough to drive the USART interface, since it is typically not required after calibration. In addition, the serial interface cable may deliver disturbances to the meter.

Summary: Defaults

As a summary, some default values for the hardware are shown in the table below.

Table 5. Hardware Defaults.

Parameter	Value	Note
System clock	4MHz	External crystal by default
Maximum Current	10A	External amplifier chops above 10A
Nominal Voltage	230V	Varistor chops signals above 250V
Current Gain Ranges	1.25 / 10 / 69	

Meter Firmware

The firmware is written in C language and compiles on IAR Embedded Workbench, versions 3.10 and upward. Previous compiler versions do not support the ATmega88 and cannot be used as such. The language syntax is standard C, so the firmware should be easy to port to other C compilers of choice.

The firmware is interrupt-driven, which means the main program consists of an endless loop that is halted by interrupt requests on a regular basis. It is important that the interrupt requests arrive regularly since they are used as the time base for calculations. Variations in interrupt intervals will show in the accuracy of measurement results.

Interrupt requests are generated by the ADC, which is driven by a prescaled system clock.

Sampling & Timing

The Analog-to-Digital Converter (ADC) is set to operate in Free Running Mode, which means a new conversion will start immediately after the current conversion completes. At the end of a conversion, the ADC raises an interrupt request and the MCU starts executing the Interrupt Service Routine (ISR). The ISR reads, processes, and accumulates sampled data and then returns execution to the main program.

The main program is mostly idle and is activated only when enough data has been accumulated for the end-of-cycle calculations to start. Once started, the ISR may still continue to interrupt the main program, even when it is busy calculating.

ADC interrupt requests occur at sampling rate, which is by default:

Equation 10. Default Sample Rate.
$$f_S = \frac{f_{CLK}}{128 \times 13} = \frac{4000000Hz}{1664} = 2403.85Hz$$

DC Offset Removal

When the ISR is started its first task is to read sampled data from the ADC and store it in a location in SRAM where it cannot be overwritten by new data. The first task to perform on the data is then to remove any DC offset. This is carried out using a digital High-Pass Filter (HPF) of type Infinite Impulse Response (IIR). The transfer function of the filter can be written as follows:

Equation 11. High-Pass Filter, Infinite Impulse Response Type.
$$y[n] = 0.996 \times y[n-1] + 0.996 \times x[n] - 0.996 \times x[n-1]$$

This is a short – and simple – digital filter, but it removes DC sufficiently well and has a reasonable settling time. For a step change in input, the filter will settle to within 1% of its final value in some 1200 samples. This means that after power-on the meter should be allowed a few seconds to settle before measurements start. This is not a problem when the meter is running, as there are no large step changes in DC level.

It should be noted that the filter is realized with a bit length higher than the input or output data. If internal filter data is truncated, the filter will rather generate offset than remove it. Hence, the filter implementation uses 32-bit data types (signed long) for intermediate and output data.

Active Power Measurement

Active power is defined as the power used by a device to produce useful work. Mathematically it is the definite integral of voltage, u(t), times current, i(t), as follows:

Equation 12. Active Power Definition.

$$P \equiv \frac{1}{T} \int_{O}^{T} u(t) \times i(t) \, dt \equiv U \times I \times \cos(\varphi)$$

Here U and I are the respective voltage and current Root-Mean-Square (RMS) values and phi is the phase lag between current and voltage. The discrete time equivalent is:

Equation 13. Active Power Definition in Discrete Time.

$$P \equiv \frac{1}{N} \int_{n=0}^{N-1} u(n) \times i(n)$$

Here u(n) and i(n) are the sampled instances of u(t) and i(t) and N is the number of samples. Active power is calculated simply as the average of N voltage-current products. It can be shown that this method is valid for both sinusoidal and distorted waveforms.

The implementation uses 32-bit data types (signed long) for storing accumulated data. The result is stored as a floating-point number. When the meter has been calibrated, the active power results are in units of watts.

Energy Pulse Output

Most energy meters are required to provide a pulse, which frequency is proportional to the amount of active power measured. The pulse typically drives an LED, which is used for calibration or verification purposes. The frequency of the energy pulse, EP, and the amount of active power measured are related as follows:

Equation 14. Frequency of Energy Pulse Output.

$$f = \frac{P \times MC}{3\,600\,000}$$

Here P is active power in watts and MC is the meter constant in pulses/kilowatt-hour. The meter constant is part of the calibration coefficient set, and can be set to any integer value, ranging from one to 65535. Taking into account the system clock, f_{CLK}, and the timer/counter prescaler, PS, the energy pulse interval is derived as follows:

Equation 15. Interval of Energy Pulse Output.

$$T = \frac{f_{CLK} \times 3\,600\,000}{PS \times P \times MC}$$

It should be noted that after the meter constant has been set there is only one variable; active power, P. In order to save instruction cycles, the expression is reduced to T = N/P, where N is a pre-calculated, constant numerator. The numerator is calculated during firmware initialization, when MC is read from EEPROM. The default numerator is as follows:

Equation 16. Constant Numerator at Default Meter Constant.

$$N = \frac{f_{CLK} \times 3\,600\,000}{PS \times MC} = \frac{4\,000\,000 \times 3\,600\,000}{1024 \times 10\,000} = 1\,406\,250$$

After the required pulse interval has been calculated the timer/counter (T/C) is programmed to produce a pulse output with an interval as close as possible to the one derived. The 16-bit T/C is clocked by a prescaled system clock. The prescaler is chosen such that even the smallest pulse intervals can be accurately represented. The problem is to also be able to accurately produce pulses with very long intervals. For this purpose, the T/C is extended with a firmware counter.

Regular Mode T/C

The regular T/C mechanism is illustrated in Figure 6. In this mode of operation the energy pulse is set on each compare match.

Figure 6. Basic Timer/Counter Operation.
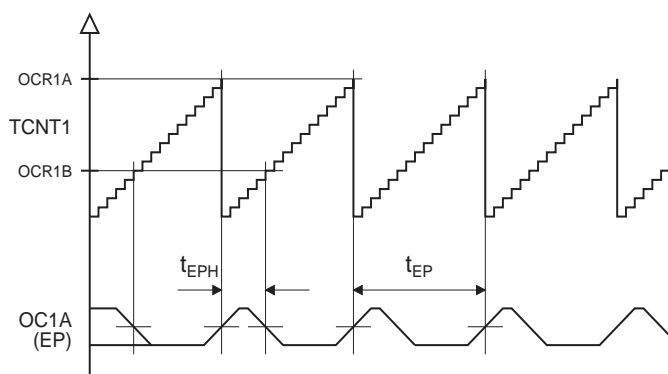
A compare match with OCR1A sets the OC1A output and a compare match with OCR1B clears the same. The pulse interval, t_{EP}, is controlled by OCR1A and the pulse width, t_{EPH}, by OCR1B.

Extended Mode T/C

When the calculated energy pulse interval exceeds 16 bits the firmware extended, 16-bit counter (TC1ext) is activated. The firmware counter value is increased each time the hardware counter reaches 16-bit maximum, effectively making the extended counter 32 bits in length. The extended counter mechanism is illustrated in Figure 7.

Figure 7. Extended Timer/Counter Operation.
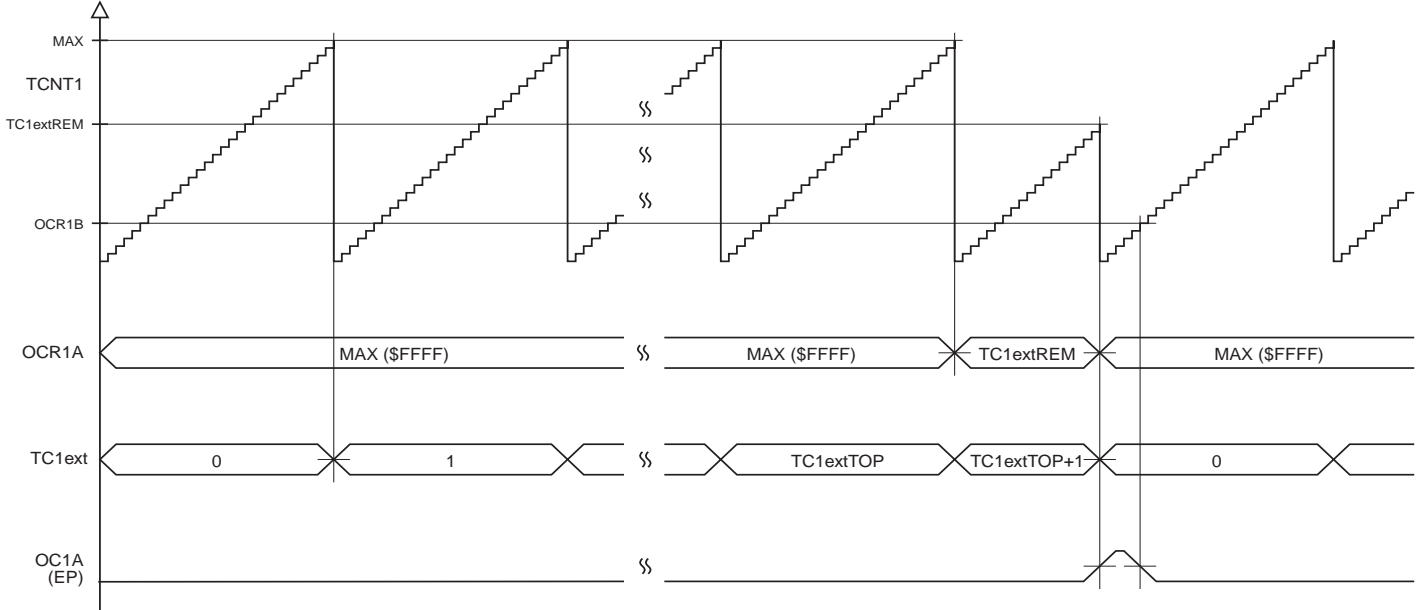
Display Pulse Outputs

For each energy pulse emitted, an internal pulse counter is increased. When the counter reaches a given threshold a display pulse sequence is initiated, as shown in Figure 8. The threshold can be set using calibration coefficients.

Figure 8. Generation of Display Pulses.
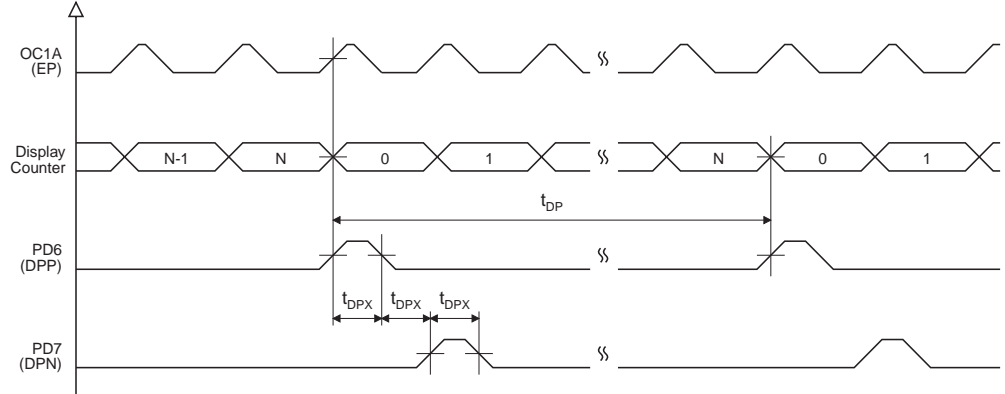
Pulse width and spacing, t_{DPX}, are firmware constants, which are easy to alter. The default setting generates 100ms long pulses.

Voltage and Current Measurement

The firmware calculates RMS (Root-Mean-Square) values of voltage and currents. An RMS value is defined as the square root of the mean value of the squares of the instantaneous values of a periodically varying quantity, averaged over one complete cycle. The discrete time equation for calculating voltage RMS is as follows:

2566A-AVR-07/04

Equation 17. Voltage RMS Calculation in Discrete Time Domain.

$$U_{RMS} = \sqrt{\frac{\sum_{n=0}^{N-1} u^2(n)}{N}}$$

Current RMS is calculated using the same equation, only substituting voltage samples, u(n), for current samples, i(n).

Accumulated data is stored 32 bits wide (signed long) and the calculation result is stored as a floating-point number. When properly calibrated, the resulting voltage measurement is in units of volts and current measurements in units of amperes.

Tamper Detection

A tamper condition enters when meter wiring is altered in a pilfering manner, typically with the intention to reduce electricity billing. The firmware detects, signals and continues to measure accurately under more than twenty known tamper conditions, including reversal of current and partial or whole earth reroutings.

Tamper detection is based on monitoring current flow in both live and neutral wires. Tamper indicators are set if any inconsistency is detected between the two currents. Actions are then taken to ensure measurement data is recorded correctly, regardless of the type of pilfering attempt.

Earth Fault

An earth fault means some or all loads have been connected to another ground potential and not the neutral wire. A partial earth fault is illustrated in Figure 9. Full current, I_{TOT}, only flows through one of the current transformers, since part of the return current, I_2, does not go through the meter. As a consequence, the current in the neutral wire, I_1, is less than that in the live wire, I_{TOT}. Alternatively, if live and neutral wires have been swapped, the current in the live wire is less than that in the neutral.

Figure 9. Partial Earth Fault.
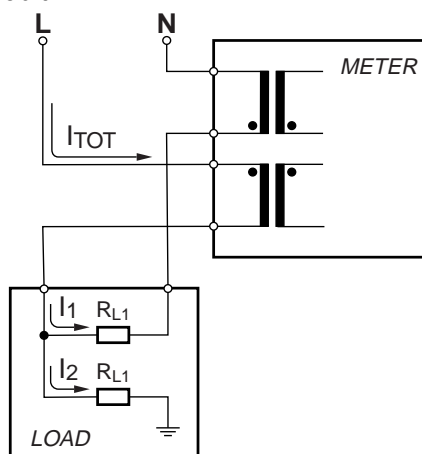
The firmware constantly monitors current flow in both wires and signals an earth fault if the magnitude difference between the two exceeds a given a threshold. The threshold has been hardwired to 5%, but is easily adjusted in the source code.

The earth fault indicator is set during initialization and will not be cleared until at least one set of valid readings have been found. This means that if the meter is powered on at a no-load condition the fail indicator will remain constantly set.

The firmware always uses the larger of the two readings for driving the energy pulse and an earth fault therefore has no effect on the accuracy.

Reversed Current

A reversed current condition means current in one or both wires is flowing in the wrong direction. As a consequence, active power readings will have the wrong sign. The firmware activates the reversed current indicator when any of the two currents has a sign opposite the one expected. Figure 10 illustrates how return current, I_2, has been reversed while live current, I_1, flows in the expected direction.

Figure 10. Reversed Return Current.
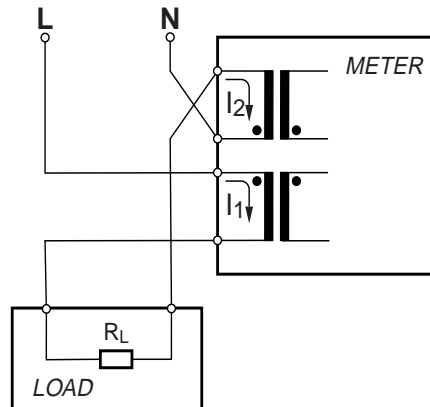
The reversed current indicator is set during initialization, but is soon cleared, provided currents in live and neutral wires flow in the correct direction.

The firmware always uses the absolute value of active power for driving the energy pulse and the direction of current therefore has no effect on accuracy.

Calibration

No two meters are alike and individual variations are to be expected. Typical tolerance figures for components used in the meter are 5%, which means the assembled meter has an inherent error of the same magnitude. Hence, each meter must be calibrated before accurate measurement result can be obtained.

Rather than populating the meter with trimming resistors and rely on slow, manual calibration the procedure is readily carried out in the digital domain. Calibration coefficients are first calculated for each meter individually, then stored in on-chip EEPROM and later retrieved during firmware initialization. The coefficients trim the calculations such that measurement results are accurate within limits.

Digital calibration is accurate and efficient, it is quick to perform, requires little or no manual intervention, and does not degrade over time. In addition, calibration data is safely stored in the internal EEPROM.

Phase

Phase displacements between signals are introduced by current transformers, input filters and by the multiplexing of input channels. Some DC immune transformers may well introduce phase displacements of more than 5 degrees while the multiplexing introduces a time difference between all channels, inversely proportional to the sampling frequency. At approximately 2400Hz sampling frequency, the time delay is 1/2400Hz = 0.42ms, which means that at 50Hz mains frequency there is a phase difference of 360 x (50Hz/2400Hz) = 7.5 degrees.

The phase displacements are adjusted using linear interpolation, which is fast, easy and sufficiently accurate. The limitation is that linear interpolation introduces a constant time delay to the signal, which means phase can be correctly adjusted for one frequency, only. Signal components above mains frequency, i.e. harmonics, cannot be simultaneously phase-adjusted using this approach. Typically, though, this is not a problem since most of the signal energy lies in the first harmonic.

The algorithm uses two subsequent samples to interpolate an intermediate point. This means that the higher the sampling frequency, the lower the phase adjustment margin. At approximately 2400Hz sampling rate (800Hz per channel) and 50Hz mains frequency, the highest phase delay that can be interpolated is 360 x (50Hz/800Hz) = 22.5 degrees.

The effect of the phase calibration coefficients is shown in the following equation.

Equation 18. Effect of Phase Calibration Coefficient.

$$Z = \frac{PCC}{65536} \times \frac{360° \times f_M \times 128 \times 13 \times 3}{f_{CLK}}$$

Here PCC is the phase calibration coefficient, f_M is the mains frequency and f_{CLK} the system clock frequency. There is one phase calibration coefficient for each input channel, i.e. three in total. Note, that the 16-bit phase calibration coefficients are treated unsigned.

Current Gain

Magnitude errors are introduced to all measurement results due to variations in discrete component values, but are easily corrected using a set of gain calibration coefficients. There are separate gain coefficients for voltage and each amplification range of both current channels, i.e. 1 + 2 x 3 = 7 in total.

Current samples are filtered, phase adjusted and then accumulated. For active power calculations, the current samples are first multiplied with voltage samples and then accumulated. At the end of each calculation cycle all accumulation registers are normalized and the results are then multiplied with corresponding gain coefficients. For each current channel, one of three gain coefficients is used, depending on the amplifier setting, i.e. different coefficients are used at low, medium and high amplification.

The effect of current gain coefficients is shown in the following equation.

Equation 19. Effect of Current Gain Coefficients.

$$I_{CAL} = \frac{I}{IG_N}$$

Here IG is the current gain coefficient and N denotes amplifier setting (N=1,2,3). The same equation holds for both live and neutral current measurements.

The 16-bit coefficients are treated unsigned.

Example

Assume meter is configured for 10A maximum current. At full-scale, sinusoidal input and lowest amplifier setting the peak-to-peak value of sampled data is 1023. After prescaling (x32) and DC removal, the amplitude of the sampled signal is:

Equation 20. Amplitude After HPF.

$$\hat{i} = \frac{255 \times 1023}{2} = \pm130432$$

After scaling (1/64) and taking the square of each sample, the amplitude is now +/-4153444. At 4MHz system clock, 128 ADC prescaler and 50Hz mains there are 16.026 samples per mains cycle. Accumulating over 25 x 16.026 = 401 samples, normalizing and taking the square root, the result before calibration is:

Equation 21. Result Before Calibration.

$$I = \sqrt{\frac{401 \times 4143444}{401 \times \sqrt{2}}} = 1713.755$$

Dividing the result with a calibration coefficient of 1713.755 / 10 = 171.3755 the result is:

Equation 22. Result After Calibration.

$$I_{CAL} = \frac{1713.755}{171.3755} = 10.000$$

Voltage Gain

Voltage is calibrated similarly as current, but since gain remains constant only one coefficient is required. The effect of the voltage gain coefficient, UG, is shown in the following equation.

Equation 23. Effect of Voltage Gain Coefficient.

$$U_{CAL} = \frac{U \times UG}{65536}$$

The 16-bit gain coefficient is treated unsigned.

Active Power Gain

Active power measurements do not require a dedicated gain coefficient, but use current and voltage gain coefficients. Measurement results are multiplied with the voltage coefficient and the corresponding current coefficient. The effect is shown in the following equation.

Equation 24. Effect of Voltage and Current Gain Coefficients on Active Power.

$$P_{CAL} = \frac{P}{UG \times IG_N}$$

Meter Constant

The meter constant dictates the relationship between amount of active power measured and the frequency of the Energy Pulse output. The larger the active power, the higher the frequency, f_{EP}, as illustrated in the equation below.

Equation 25. Effect of Meter Constant.

$$f_{EP} = \frac{P \times MC}{3\,600\,000}$$

Here P is active power in watts and MC is the meter constant in impulses/kilowatt-hour. Note, that the pulse frequency also depends on the system clock. The above equation assumes a system clock of 4MHz.

The calibration coefficient is treated as a 16-bit, unsigned integer.

Limitations

The Energy Pulse output is typically wired to an LED and used for calibration and verification purposes. The pulse must be sufficiently long to light the LED for an optical reader to register it, but also sufficiently short to allow a reasonably high output frequency. The default is 50ms active time, which limits the frequency to about 10Hz. At 4MHz system clock, 10A max current and 230V nominal voltage, this means the maximum meter constant is about 15000 imp/kWh. To increase the meter constant above this, the maximum current and the pulse length must be decreased.

Another limitation is the quantization of the pulse interval. At default system clock (4MHz) and default T/C prescaler (1024), the timer is updated once every 1024/4000000 = 0.256ms. Assuming highest default pulse rate (10Hz) the pulse interval is 100ms and the highest quantization inaccuracy is +/-(0.256/100) = +/-0.256%. For practical purposes it is recommended to scale system clock, maximum current and meter constant such that the timer/counter interval never goes below, say, 500 timer ticks. By default (4MHz, 10A, 230V, 10000imp/kWh) the pulse interval will not go below 500 ticks.

Display Constant

A pulse pair is provided to drive a stepper-motor counter. Typically, the required display pulse rate is much lower than the energy pulse rate. The pulse rate, f_{DP}, can be adjusted using the DPC calibration coefficient, as follows:

Equation 26. Effect of Display Pulse Constant.

$$f_{DP} = \frac{f_{EP}}{DPC}$$

Here f_{EP} is the energy pulse rate. The 16-bit display constant is treated unsigned.

Coefficient Layout

The EEPROM layout of calibration coefficients is shown in the table below. All coefficients are 16-bit wide and stored with high byte first, then low byte.

Table 6. Calibration Coefficient Layout in EEPROM.

ADDR.	+ 0x00	+ 0x01	+ 0x02	+ 0x03	+ 0x04	+ 0x05	+ 0x06	+ 0x07
0x00	PCC0		PCC1		PCC2		ILG0	
0x08	ILG1		ILG2		ING0		ING1	
0x10	ING2		UG		MC		DPC	
0x18					CRCW		CRC16	

PCC is the phase calibration coefficient and ILG and ING are the current gain coefficients for live and neutral wires, respectivelty. UG is the voltage gain coefficient, MC the meter constant and DPC the display pulse constant. CRC16 contains the checksum for the low 32 bytes of the EEPROM. If CRCW is set to to 0x4357 then the checksum is calculated by the firmware and written to CRC16.

19

Program Constants

Program constants are values that can be altered before the firmware is compiled, but not afterwards. Typical constants are various bit and flag definitions, but also included are some fundamental invariables, such as pulse lengths and sample rates. Some of the fundamental constants are discussed below.

AMP_LO

This constant defines the threshold when to increase amplification of current signal. Amplification is increased when the amplitude of the filtered current signal drops below this level. Should be below (255 x 1023) / current gain.

SAT_LO, SAT_HI

These are low and high levels, which define saturation limits for current samples. Amplification is decreased when unconditioned current samples go below low limit or above high limit. Recommended a few LSB's below max and above min.

GAIN_HOLD

This is the number of samples to wait before allowing a new gain switch to take place. By default, the amplifier settles in a few hundred sampling cycles, which means a value of 100 should be enough.

I_MIN

Starting current in amperes. Active power readings are cleared when current measurement drops below this limit. Should be set according to IEC 61036 standard.

OFFSET

Constant offset, which is added to all measurement results. Improves linearity at low amplitudes, if properly scaled. Should be around ½ LSB x 255 = 128.

NMAX

This is the number of samples to accumulate during each calculation cycle. Large values create more stable results but at longer intervals. Should be an integral multiple of number of samples per mains cycle.

NORM

This is the inverse of NMAX. It is a pre-calculated constant, which is used at the end of each accumulation cycle.

DP_ON

This is the length and spacing of display pulses, in units of sampling cycles.

Duty Cycle

The main program is mostly idle (looping) and is interrupted only when a fresh sample is available from the ADC or when the timer/counter requires service. The most frequently occurring event is the ADC interrupt service, which must be complete before the next service request arrives. In order to monitor that the ADC ISR does not consume too many instructions cycles the service routine sets and clears a pulse upon entry and exit, respectively.

By default, the duty cycle pulse is routed to the lowest bit of port B. The duty cycle is readily monitored with an oscilloscope connected to PB0. The closer the duty cycle is to 100%, the higher is the risk that there will not be enough clock cycles to process all data. A conservative duty cycle is below 50%.

**Summary:
Defaults**

As a summary, some firmware defaults are shown in the table below.

Table 7. Firmware Defaults.

Parameter	Value	Note
Compiler	IAR 3.10	
Device	ATmega88	8kB FLASH, 1024b RAM, 512b EEPROM
Starting current	2mA	See I_MIN constant
Calculation cycle	200ms	See system clock and NMAX constant
Energy pulse rate	10000 imp/kWh	See MC calibration coefficient
Energy pulse length	50ms	See system clock and OCR1B initialisation
Display pulse rate	100 imp/kWh	See DPC calibration coefficient
Display pulse length	100ms	See system clock and DP_ON constant
Terminal setting	38400 bauds	See USART initialisation

Setting Up the Meter

In stand-alone mode the meter only needs to be attached to the mains wires. During calibration and testing, the meter must be connected to a computer terminal via the serial interface and for programming purposes the meter must be connected to a computer via the In-System Programming (ISP) interface.

It is recommended to isolate the meter from other hardware using optocouplers.

**Warning:
Ground Precaution**

The meter must be operated with care when connected to mains (high voltage) lines. One common pitfall is to neglect ground wirings and nonchalantly proceed with connecting the meter to external hardware, such as a personal computer. The results can be destructive.

Figure 11. Short Circuit between Live and Neutral Wire.
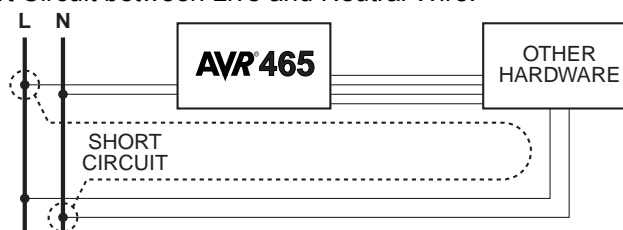
The meter has no galvanic isolation from live wires, which means even the apparently harmless meter ground is firmly bolted to the live voltage. Connecting the meter to another mains-powered hardware can lead to short-circuit between live and neutral wires, as shown in the figure above. Not a recommend thing to do.

Hardware Set-Up

For stand-alone operation, the meter is simply connected to the mains wires as illustrated in the schematic. The meter automatically starts to measure power and stores energy consumption data in the display counter. Note that the meter needs to be calibrated before reliable measurement data can be expected.

For programming, calibration and testing purposes the meter needs to be temporarily connected to external hardware, such as a computer or a microcontroller. The external hardware writes the firmware into the FLASH memory of the meter MCU. It then assesses the accuracy of the meter and writes calibration data into the EEPROM of the MCU. The meter is ready to run in stand-alone mode after programming and calibration has completed successfully.

A general illustration on how to connect the meter for programming, calibration and metering is shown in Figure 12. Please note that this is a generalized block schematic and does not include all recommended safety precautions.

Figure 12. Connecting the Meter to External Hardware.
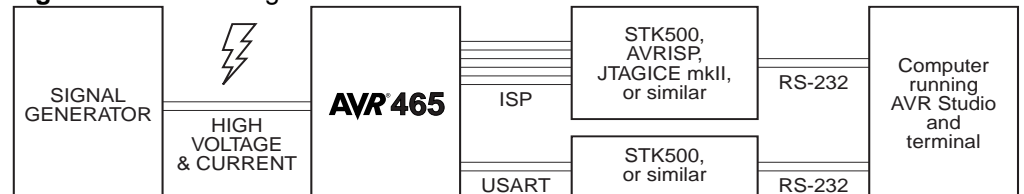
Warning: Galvanic Isolation

It is recommended to create a galvanic isolation barrier between the mains-coupled meter and the other hardware. This can be done using optocouplers at the ISP and USART interface or at the RS-232 interface. Optocoupling both ISP and USART makes the STK500 and debugger "safe", but requires many optocouplers. Alternatively, creating the galvanic barrier at the RS-232 interfaces requires less optocouplers but makes all devices except the PC hazardous.

Another alternative is to use an isolation transformer to create a floating front-end, which can then be bolted to the same ground as the rest of the electronics.

Programming

The meter AVR is programmed via the In-System Programming (ISP) interface, which means the device does not need to be removed for programming. The ISP connector (J1, see schematic) is connected to a programmer (STK500, ATAVRISP, or similar) or a debugger (ATJTAGICEmkII, or similar). Software such as AVR Studio can then be used for programming FLASH and EEPROM of the target AVR.

Calibration & Metering

The meter constantly sends measurement data via the asynchronous interface. To read measurement data (also during calibration), the asynchronous interface must be connected. The asynchronous connector (J2, see schematic) is wired to a microcontroller directly or to a computer via a RS-232 buffer. The asynchronous interface cannot be wired to a RS-232 port directly, since it is not buffered.

A RS-232 buffer is readily found on the STK500. Connect the asynchronous interface of the meter to STK500 pins labeled RS232 SPARE. Then connect the STK500 D-connector labeled RS232 SPARE to the serial port of the computer. Use a terminal software of choice to read data from the serial port.

A general illustration on how to connect the meter for programming, calibration and metering is shown in Figure 13. Also shown in the figure is a recommended optical isolation board.

Figure 13. Example Set-Up for Calibration and/or Metering.
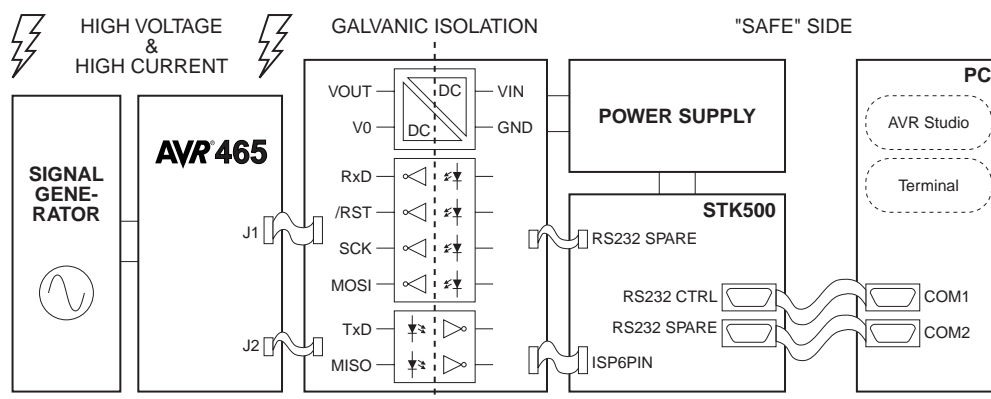
Firmware Set-Up

To download the firmware into the meter first open the compiler workspace, compile the project and use AVR Studio to send the compiled file to the target, as described above. Alternatively, use AVR Studio to send the precompiled firmware directly to the target.

Set up fuses to match hardware environment. The default is to use external crystal, with full output swing and long start-up time. This means fuses CKSEL are set to 0111 and fuses SUT to 11. Fuses CKDIV8 and CKOUT should be unprogrammed.

Calibration Example

There are many methods for calibrating the meter. Here is presented one. This method requires a programmable, high-voltage, high-current signal generator. Also required is hardware for programming the AVR in the meter. An STK500 can be used, but a debugger such as JTAGICE mkII is recommended, since it allows EEPROM to be edited more easily.

Set-Up

Connect the signal generator to the meter front-end. Then connect the meter to a PC using a RS-232 buffer circuit, such as the spare RS-232 interface on the STK500. Start a terminal software on the PC and configure it to receive data from the meter.

Step 1 - Initialisation

Start by writing the following set of calibration coefficients to EEPROM. This will set phase calibration to zero and gain to unity in all calculations.

Table 8. Initialization of Calibration Coefficients.

ADDR.	+ 0x00	+ 0x01	+ 0x02	+ 0x03	+ 0x04	+ 0x05	+ 0x06	+ 0x07
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01
0x08	0x00	0x01	0x00	0x01	0x00	0x01	0x00	0x01
0x10	0x00	0x01	0xFF	0xFF	0x27	0x10	0x00	0x64
0x18	0x00	0x00	0x00	0x00	0x43	0x57	0x00	0x00

If using a programmer, create a text file as shown below and load the contents of the file into the EEPROM. The file content is of type Intel Hex.

```
:1000000000000000000000010001000100010001EB
:100010000001FFFF271000640000000043570000AC
:00000001FF
```

Next, reset the meter. This will load the calibration coefficients into memory.

Step 2 - Gain

Apply nominal voltage and large – preferably full-scale – current to the meter. Voltage and current should be in phase, i.e. the power factor should be unity. Allow measurement results to settle for a few seconds and then record voltage and current readings. Calculate gain coefficients using the previously presented equations.

Table 9. Example Readings.

Input	Measured	Gain Error	Calibration Coefficient
230V 5A PF 1	$U = 1241.699$	5.3987	UG = 65536 / 5.3987 = 0x2F6B
	$I_L = 630.63784$	126.1276	ILG[0] = 0x007E
	$I_N = 626.58277$	125.3166	ING[0] = 0x007D

Using a programmer or a debugger of choice, load gain coefficients to EEPROM. In this example, the following coefficients are loaded.

Table 10. New Calibration Coefficients.

Address	Coefficient	Data
0x06	ILG0, High Byte	0x00
0x07	ILG0, Low Byte	0x7E
0x0C	ING0, High Byte	0x00
0x0D	ING0, Low Byte	0x7D
0x12	UG, High Byte	0x2F
0x13	UG, Low Byte	0x6B
0x1C	CRCW, High Byte	0x43
0x1D	CRCW, Low Byte	0x57

Reset the meter.

Step 3 - Phase

Continue to apply nominal voltage and a large current to the meter. Allow readings to stabilize and then record active power, voltage and current readings. Use voltage and current readings to calculate apparent power; S = UI. Compare with active power readings and evaluate phase error, as follows:

Equation 27. Phase Error.

$$\varphi = a\cos\left(\frac{|P|}{S}\right)$$

Derive phase calibration coefficient based on phase error, and add the result to the default phase calibration coefficient. An example is shown in the table below.

Table 11. Example Readings.

Input	Measured	Apparent Power	Phase Error	PCC
230V 5A PF 1	U = 230.038			PCC0 = 0
	I_L = 4.99541 P_L = 1131.194	S_L = 1149.1341	Phi_L = 10.138	PCC1 = 0x7359
	I_N = 5.00943 P_N = 1106.641	S_N = 1152.3593	Phi_N = 16.193	PCC2 = 0xB83E

Enter phase calibration coefficients into EEPROM (locations 0x00…0x05), leave other coefficients untouched, and write 0x4357 to CRCW. Reset the meter.

Step 4 - Final Gain

Still apply nominal voltage and high current to the meter. Record current measurements, and then reduce input current by a factor of about 8, so that the meter automatically switches to medium amplification range and that the sampled data has a high amplitude. Again record current and then again reduce input current by a factor of about 8. Record third set of current measurements. Then use the recorded data to calculate final gain coefficients for all current ranges, as shown in the table below.

Table 12. Example Readings.

Input	Measured	Gain Error	Calibration Coefficient
230V, 5A	I_L = 4.90191 I_N = 4.93608	0.9804 x 126.1276 = 123.66 0.9872 x 125.3166 = 123.71	ILG[0] = 0x007C ING[0] = 0x007B
230V, 1A	I_L = 1017.25948 I_N =1016.93559	1017.25948 / 1 = 1017 1017	ILG[1] = 0x03F9 ING[1] = 0x03F9
230V, 0.1A	I_L = 687.69546 I_N = 690.78431	687.69546 / 0.1 = 6877 6908	ILG[2] = 0x1ADD ING[2] = 0x1AFC

Note that the gain errors for low amplification are multiplied with the previously evaluated. This is because the previous coefficients were in effect during this set of measurement.

Write new current gain coefficients to EEPROM (locations 0x06…0x11), update CRCW to 0x4357 and reset meter.

Step 5 - Verify

The example meter was tested after the above calibration. The signal generator was connected to the front end of the meter and was programmed to perform a current sweep from 10mA to 10A. Active power readings were recorded from the Energy Pulse output. All measurement results were within 1%.

Measurement Results

Measurement results were recorded for a manually assembled prototype at room temperature.

Power Consumption

The following table shows typical current consumption figures for this application.

Table 13. Typical Current Consumption Figures.

Condition	Typical	Duration
Meter Running, All LED's Off, Display Pulses Low	3.0 mA	Continuous
Meter Running, All LED's On, Display Pulses Low	7.5 mA	Continuous
Meter Running, All LED's On, Display Counter Increment	12.5 mA	100ms

Accuracy

The measurement accuracy was tested on a calibrated energy meter. The reference was a programmable, three-phase signal generator (type Omicron, model CMC56). The signal generator was programmed to supply the meter with known voltages and currents at various power factors and then read back the frequency of the energy pulse (EP) at each setting.

For each measurement point, only one pulse interval was recorded, i.e. no averaging was done. The results from 57 measurement points were then gathered into three curves illustrating the dynamic accuracy for three given power factors.

Figure 14 illustrates how meter accuracy stays within 1% over a wide dynamic range. The current scale has been normalised, i.e. 1.000 corresponds to maximum current (10A, 20A, or whatever). The solid line represents the relative error at unity power factor, i.e. when voltage and current are in phase. The dotted line indicates relative error at power factor 0.8, capacitive, and the dashed line shows relative error at power factor 0.5, inductive.

Figure 14. Typical Measurement Accuracy.

Flow Charts

Main Program

Figure 15. Main Program.

Initialisation

Figure 16. Initialisation.

```
        ╭──────────────────────╮
        │      INITIALISE       │
        ╰──────────────────────╯
                   │
                   ▼
        ┌──╥────────────────────┐          Set up I/O pins.
        │  ║    Initialise()     │ ─ ─ ─ ─  Configure Timer /Counter.
        └──╨────────────────────┘          Configure USART & ADC.
                   │
                   ▼
        ┌──╥────────────────────┐          Read calibration coefficients
        │  ║  ReadCalibration()  │ ─ ─ ─ ─  from EEPROM.
        └──╨────────────────────┘
                   │
                   ▼
        ┌──╥────────────────────┐          Initialise variables that control
        │  ║  InitGainControl()  │ ─ ─ ─ ─  the gain.
        └──╨────────────────────┘
                   │
                   ▼
        ┌──╥────────────────────┐          Set low gain on both current
        │  ║    SetGain(LOW)     │ ─ ─ ─ ─  input channels.
        └──╨────────────────────┘
                   │
                   ▼
        ┌────────────────────────────┐
        │ Turn on Watchdog & Interrupts │
        └────────────────────────────┘
                   │
                   ▼
        ╭──────────────────────╮
        │         END          │
        ╰──────────────────────╯
```

Character Input Service

Figure 17. Character Input Service.

Calculation Cycle Service

Figure 18. Calculation Cycle Service.

```
                    ╭─────────────────────╮
                    │   CYCLE SERVICE     │
                    ╰─────────────────────╯
                              │
                              ▼
                    ┌─────────────────────┐
                    │ Clear CYCLE_FULL Flag│
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐          Make a copy of accumulated data,
                    │  Sum = Accumulator  │ - - - -  then clear accumulator to make
                    │  Accumulator = 0    │          way for new data.
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐          Increase or decrease gain of current
                    │   Adjust Gain &     │ - - - -  channels, according to control flags.
                    │Clear Gain Control Flags│       Then reset flags.
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐          Normalise data, then add a constant
                    │    Normalise &      │ - - - -  offset to it. Subtract, if accumulated
                    │ Add/Subtract Offset │          data is negative.
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐          Take square root of current and voltage
                    │ Calculate & Calibrate│ - - - -  data, then multiply accumulated data
                    └─────────────────────┘          with calibration coefficients.
                              │
                              ▼
                         ◇─────────◇          YES    ┌─────────────────────┐
                        ╱ RMS Current ╲ ─────────────▶│  Clear Active Power │
                        ╲Below Threshold?╱             └─────────────────────┘
                         ◇─────────◇                            │
                              │ NO                              │
                              ▼◀───────────────────────────────┘
                             ▽
                            CSS
                    (Continued on next page)
```

CSS

Active Power
Above Threshold? — YES

NO

P(L) within
5% of P(N) ? — NO → Signal Earth Fault

YES

SetPulse() ····· Calculate and set new pulse rate, based on active power measurements.

P(L) and P(N)
Both Positive ? — NO → Signal Reversed Current

YES

Display Measurements ····· Send measurement data over USART interface. Voltage in volts, current in amps and active power in watts.

END

SetPulse()

Figure 19. SetPulse().

Flowchart:

SetPulse(Power)

↓

Take Absolute of Power — — — — — Active power can be positive or negative, depending on the flow direction of current. Sign is irrelevant when calculting pulse interval.

↓

Power <> 0 ? — YES → $T = \dfrac{3600 \times f_{CLK}}{Power \times PRESC \times MC}$ — — — — f_{CLK} = Clock frequency; PRESC = T/C Prescaler; MC = Meter constant

NO ↓

Pulse Interval $T = T_{MAX}$

↓

$T < T_{MIN}$? — YES → $T = T_{MIN}$

NO ↓

$T > T_{MAX}$? — NO → Extended Counter = T / C_{MAX}; Counter = Remainder

YES ↓

Extended Counter = C_{MAX}; Counter = C_{MAX}

↓

Extended Counter TOP Value Set ? — YES → Set New Compare Match for T/C and New TOP Value for Extended Counter.

NO ↓

Set Counter Value as New T/C Compare Match

↓

END

AVR465

Timer/Counter ISRs

Figure 20. Timer/Counter Compare A.

Figure 21. Timer/Counter Compare B.

```
        ┌─────────────────────────┐
        │      TC1_COMPB_ISR       │
        └─────────────────────────┘
                     │
                     ▼
        ┌─────────────────────────┐
        │     Configure T/C to     │
        │  Clear Output on Next Match │
        └─────────────────────────┘
                     │
                     ▼
        ┌─────────────────────────┐
        │   Force Compare Match    │
        └─────────────────────────┘
                     │
                     ▼
        ┌─────────────────────────┐
        │     Configure T/C to     │
        │  Set Output on Next Match │
        └─────────────────────────┘
                     │
                     ▼
        ┌─────────────────────────┐
        │           END            │
        └─────────────────────────┘
```

ADC ISR

Figure 22. ADC ISR.

```
                    ┌──────────────┐
                    │   ADC_ISR    │
                    └──────────────┘
                            │
                            ▼
                         ╱      ╲       NO      ┌──────────────┐          Current signals are inverted in
                     Voltage Sample?  ────────▶ │ Invert Sample│ ─ ─ ─ ─  the gain stage. Invert them back.
                         ╲      ╱               └──────────────┘
                            │ YES                       │
                            ▼◀──────────────────────────┘
                    ┌──────────────────┐
                    │ Apply High-Pass  │ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─  Remove DC Offset from Signal.
                    │     Filter       │
                    └──────────────────┘
                            │
                            ▼
              ╱                    ╲   YES      ╱                ╲   YES   ┌──────────────────────┐
          Hysteresis Counter > 0 ?  ────────▶  Sample > Max       ────────▶│ Active Gain Control   │
              ╲                    ╱            or Sample < Min ?           │   Flag LESSGAIN       │
                            │ NO                ╲                ╱          └──────────────────────┘
                            │                        │ NO                              │
                            ▼                        ▼◀──────────────────────────────┘
                    ┌──────────────────┐    ╱                ╲   YES   ┌──────────────────────┐
                    │ Reset Gain       │    Filtered Sample    ────────▶│ Clear Gain Control    │
                    │ Control Flags &  │    Above Threshold?            │   Flag MOREGAIN       │
                    │ Decrease Counter │    ╲                ╱          └──────────────────────┘
                    └──────────────────┘         │ NO                              │
                            │◀────────────────────┴──────────────────────────────┘
                            ▼
                    ┌──────────────────┐
                    │ Apply Phase      │
                    │ Calibration      │
                    └──────────────────┘
                            │
                            ▼
                    ┌──────────────────┐
                    │ Accumulate Data  │
                    │ and set          │
                    │ CYCLE_FULL Flag  │
                    │ when Full        │
                    └──────────────────┘
                            │
                            ▼
                    ┌──────────────────┐
                    │ Increase ADC     │
                    │ Multiplexer      │
                    └──────────────────┘
                            │
                            ▼
                    ┌──────────────────┐
                    │ Maintain DPP &   │
                    │ DPN Outputs      │
                    └──────────────────┘
                            │
                            ▼
                    ┌──────────────┐
                    │     END      │
                    └──────────────┘
```

Schematic

Figure 23. Schematic.

Bill of Material

Order codes are according to Farnell catalogue in 2004. Display counter and current transformers are not part of the inventory.

Table 14. Bill of Material.

Reference	Component	Value	Order Code
C10	Capacitor, X2, Metallised Polyester	0.68 uF	3549460
C11	Capacitor, Electrolytic, 16V	470 uF	3201508
C12, C31, C57	Capacitor, Electrolytic, 16V	10 uF	3201454
C20, C30, C40	Capacitor, Electrolytic, 16V	47 uF	3201478
C50 … C56	Capacitor, Ceramic	100 nF	896755
C58, C59	Capacitor, Ceramic	18 pF	236950
D10	Diode, Zener, 500mW, 15V	BZX55-C15	369081
D11	Diode, 1A, 400V	1N4004	251707
D50, D51	LED, Red, 5mm, Low Current	HLMP-4700	323135
D52	LED, Yellow, 5mm, Low Current	HLMP-4719	323147
J1	Straight Pin Header, 6-way	2x3	511079
J2	Straight Pin Header, 2-way	1x2	143187
R10	Resistor, 1W, 500V, 5%, Carbon Film	470 Ω	510476
R20, R23, R37	Resistor, 0.4W, 5%, Metal Film	680 kΩ	333037
R21, R24, R35, R45	Resistor, 0.4W, 5%, Metal Film	1 kΩ	332690
R22, R36	Resistor, 0.4W, 5%, Metal Film	3.3 MΩ	333116
R30, R40	Resistor, 0.4W, 5%, Metal Film	68 Ω	332550
R31, R41	Resistor, 0.4W, 5%, Metal Film	6.8 kΩ	332793
R32, R42	Resistor, 0.4W, 5%, Metal Film	39 kΩ	332884
R33, R43	Resistor, 0.4W, 5%, Metal Film	330 kΩ	332999
R34, R44	Resistor, 0.4W, 5%, Metal Film	470 kΩ	333013
R50, R51, R52	Resistor, 0.125W, 5%, Carbon Film	1 kΩ	477783
R53, R54	Resistor, 0.125W, 5%, Carbon Film	10 Ω	477667
T1, T2	Current Transformer, Tae Hwa Trans	TD76V	N/a
U1	8-Bit AVR® Microcontroller	Atmega88	N/a
U2	Dual, Low-Voltage, Rail-to-Rail Op Amp	LMV358	3555768
U3	Quad Bilateral Switch	CD4066BCN	380957
U4	Linear Regulator, 3.3V	LM1117MP-3.3	120765
V1	Varistor, 40J	250 V	318619
X1	Crystal, Quartz	4.000 MHz	492917
Z1	Display Counter		N/a

This Bill Of Materials does not include any price estimates, since local variations can be large.

Developing the Meter Further

This application has been created as a cost-effective alternative for tamper-proof, single-phase metering. Not all customers may base their decision on the same assumptions as we have and therefore some customers may experience an urge to redesign some parts of the meter. We do not only encourage the customer to do so, but also provide some discussion and suggestions on how to get started with it.

Analog Front End

The current front end includes operational amplifiers providing input signals for the A/D inputs of the AVR. It should be noted that the ADC input contains a S/H (sample and hold) capacitor and that operational amplifiers are not very good at driving capacitive loads (capacitive loads introduce phase lags into the output signal, which reduces phase margin in the amplifier). Hence, there needs to be a series resistor between the amplifier output and the ADC input. Typically, a 1-kohm resistor will do nicely, but this needs to be verified against the data sheet of the manufacturer of the operational amplifier.

There is no reason why additional gain ranges could not be implemented, using the same ideology as illustrated in this design. Naturally, there is a limit on how much the input signals can be amplified before noise starts to degrade measurement accuracy, but it is possible to increase the dynamic range above the present. Alternatively, adding more gain ranges and reducing the range difference from current eight to, say, two or four may provide a more linear transfer function and increase the absolute accuracy to better than 1%. Both alternatives are included at little extra cost (only switches and a few resistors required).

Power Supply

The power supply uses half-wave rectification. The efficiency of the power supply can be increased by using full-wave rectification. This means that at a cost of a few extra diodes, the power supply will be able to deliver more current to the meter without drawing more energy from the mains lines.

The power consumption of the supply is around 10VA, which may proove too high for some applications. For example, IEC60136 standard limits the maximum power consumption to 10VA. With the component values given in this document, power consumption is just around 10VA. To lower the consumption, reduce the size of C10 (see schematic at end of document) and recalculate necessary component values, as described earlier. If C10 is made too small, the input voltage to the regulator will not rise high enough for regulation to work properly.

Power consumption of the microcontroller section can be decreased by putting the device into sleep at the end of each sampling cycle and waking it up again when a new data conversion is ready.

Microcontroller Section

The microcontroller can be configured to run from internal RC clock, as described in the data sheet. Using the internal oscillator removes the need for an external crystal and lowers the overall price of the meter. The drawback is the internal oscillator (just like any RC oscillator) is sensitive to variations in temperature. This means the frequency changes with temperature and that measurement results will change accordingly. In addition, the bit error rate of asynchronous communication will increase if the system clock is not stable enough.

This application note is written with the ATmega88 in mind, but the design is easy to migrate to almost any other AVR. Alternative design ideas include:

- Remove tamper detection logic and turn the design into a standard single-phase meter. Firmware should then fit into 4kB and the microcontroller can be replaced by an ATmega48, for example.
- Add functions such as remote reading, demand recording or multiple tariffs. Upgrade to ATmega168, for example, if the firmware exceeds 8kB.
- Add integrated LCD support simply by migrating to ATmega169.

Meter Firmware

No software is ever complete and there are probably as many potential improvements as there are readers of this document. The good news is the firmware is open source and can be edited to fit any application.

Some recommended starting points for software enhancements include:

- Put the MCU to sleep as often as possible.
- Add apparent power, S, to measurement set. Hint: S = UI.
- Add reactive power, Q, to measurement set. Hint: $S^2 = P^2 + Q^2$.
- Consider alternate reactive power approach. Hint: If u(t) is delayed 90 degrees then reactive power can be calculated the same way as active power, i.e. Q = Int{u(t-90) x i(t)}. Note: academic discussion on this method's validity under non-sinusoidal conditions has been recorded since 1927 (see C. I. Budeanu, "Puissances reactives et fictives", Instytut Romain de l'Energie).

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

2566A-AVR-07/04