EM773 Energy Metering IC
Technical Training

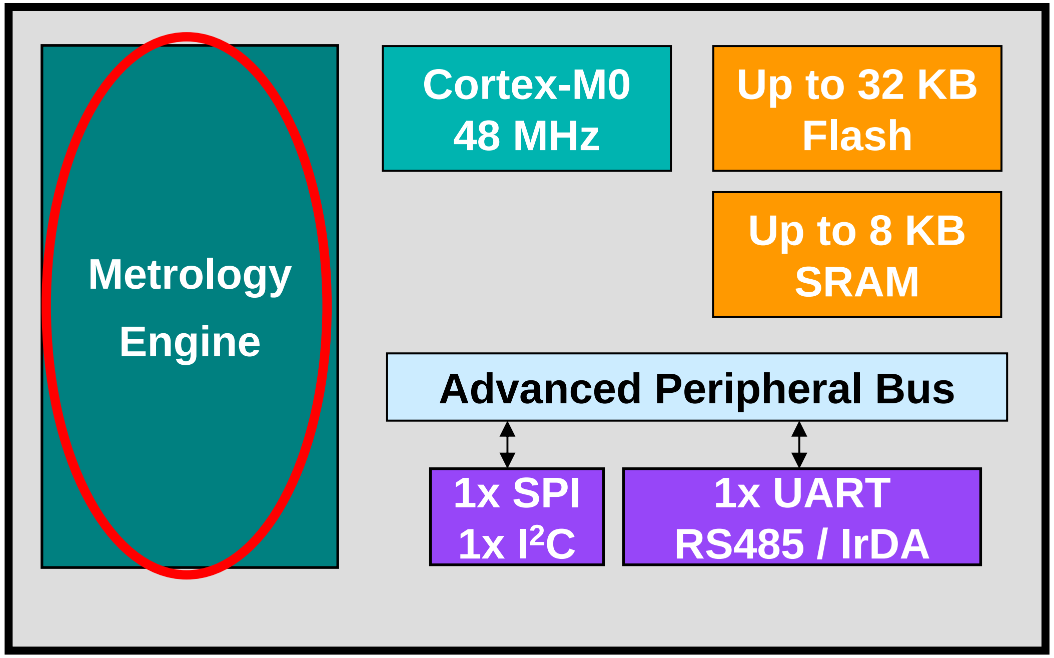
"The world's first ARM with
smart metering metrology hardware!"

Energy Metering IC for Non-Billing Meters EM773

- Perfect solution for non-billing metering apps
  - Plug meters
  - SMART appliances
  - Industrial & consumer sub-meters
- Exceeds market requirements with better than 1% metering accuracy
- No metering know-how required
- Product differentiation via application SW
- Wireless M-Bus demonstrator design
- Option for wired UART, SPI or I2C metrology output to local system or LCD display

Energy Metering IC for Non-Billing Meters EM773

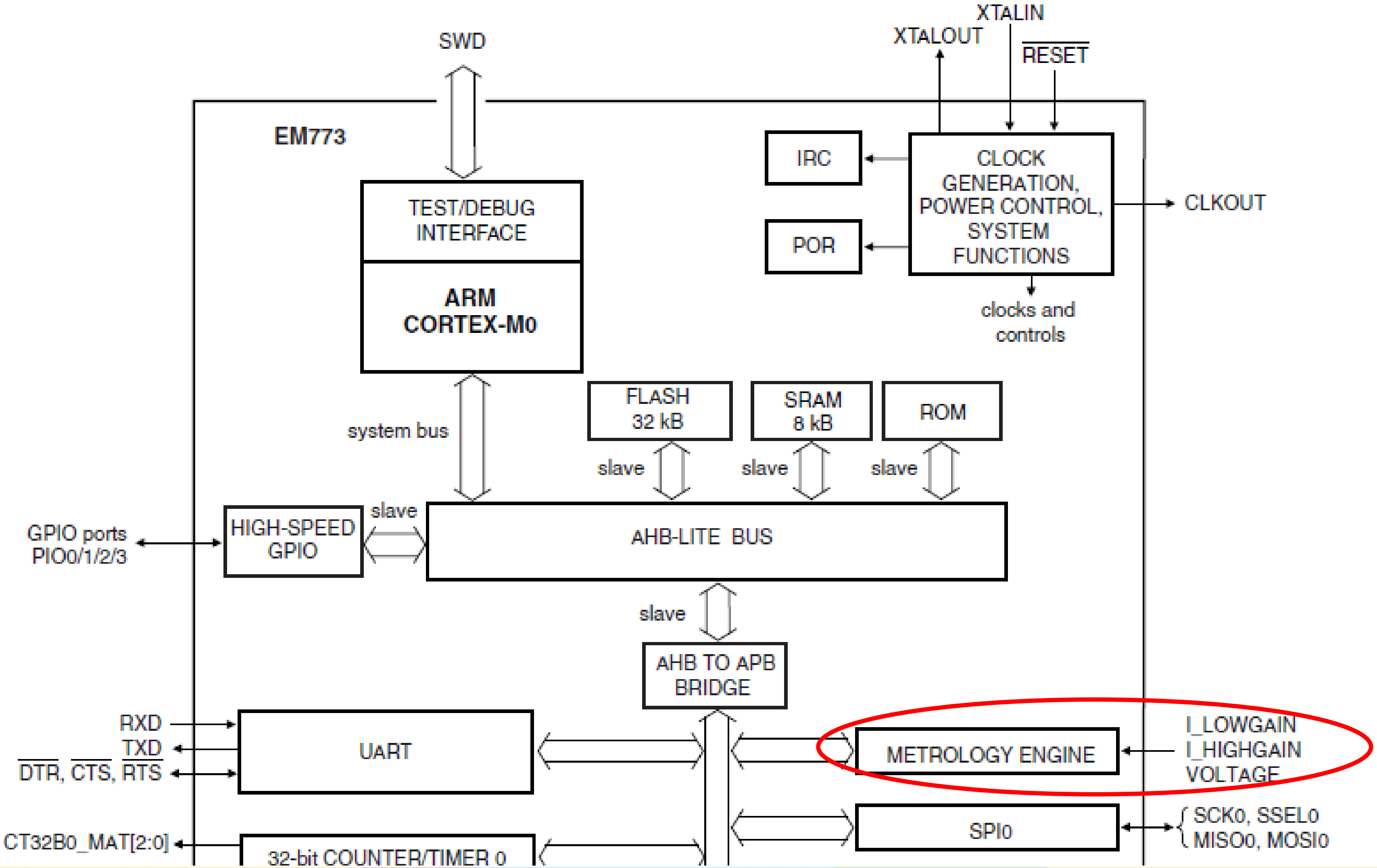
Optimized metrology inside with optional network connection



33-pin HVQFN

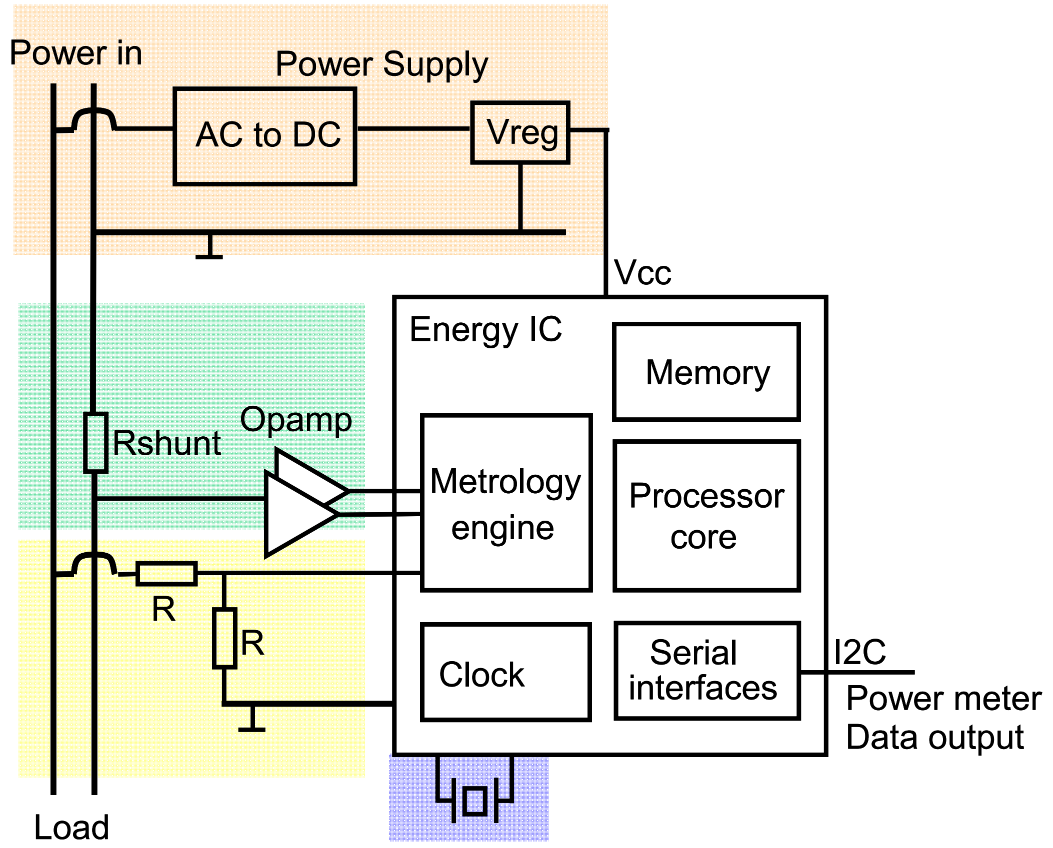
- Built-in metrology engine hardware and software
- Application programmable
- UART available for communications port
- Standard ARM support ecosystem available for easy development

EM773 Metrology Engine Inputs



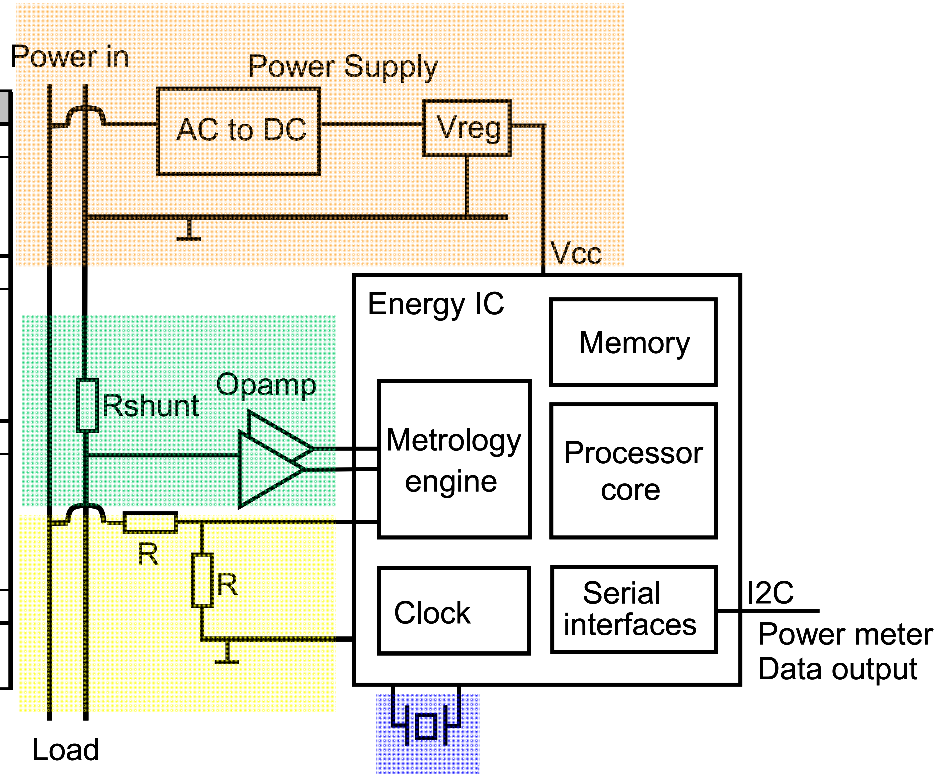
Energy Metering IC EM773
Principle block diagram

- Power supply
  - 3.3V as V_{DD} required
- Analog circuitry for current measurement
  - accuracy options
- Analog circuitry for voltage measurement
- Oscillator

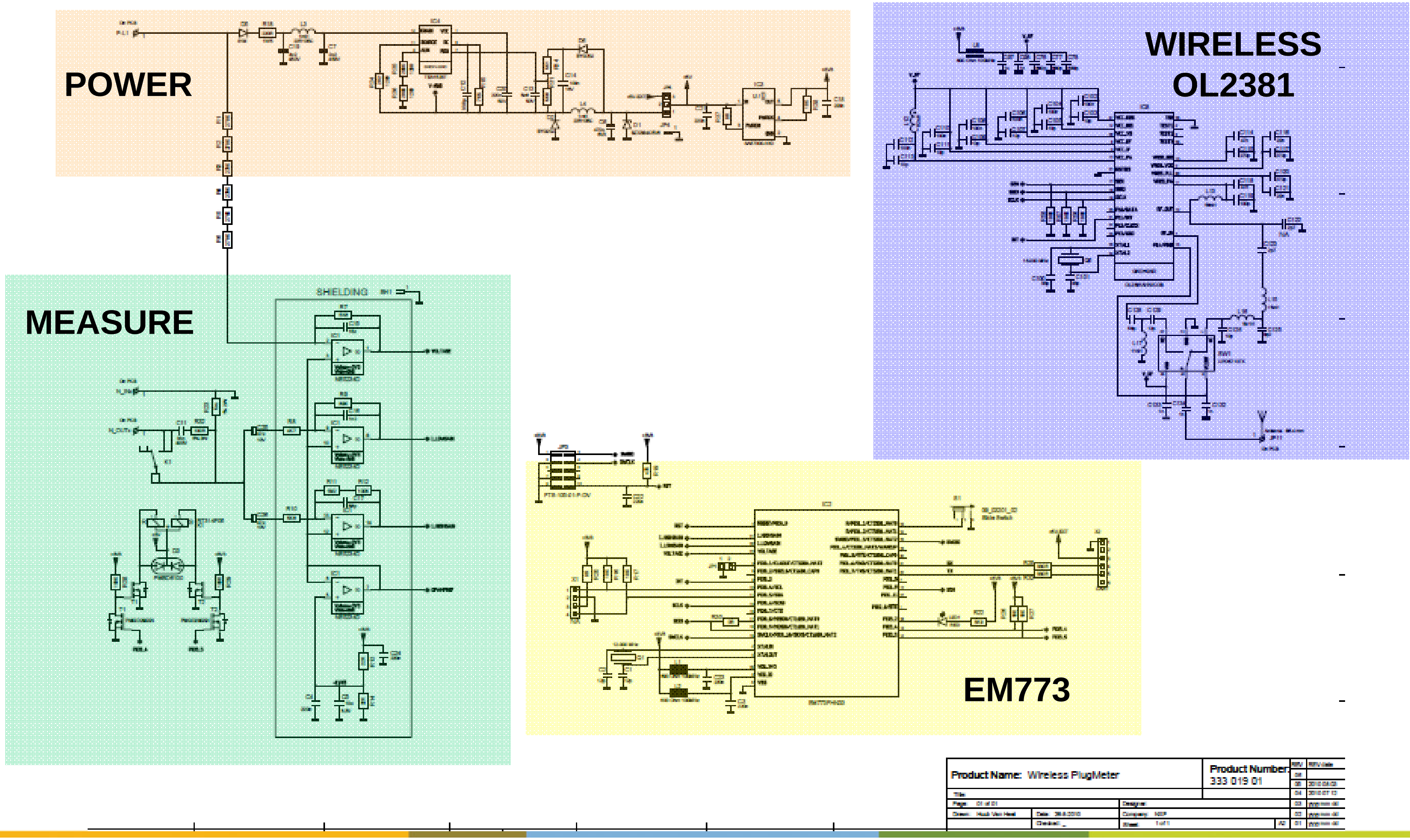


Energy Metering IC EM773
Key components of BOM

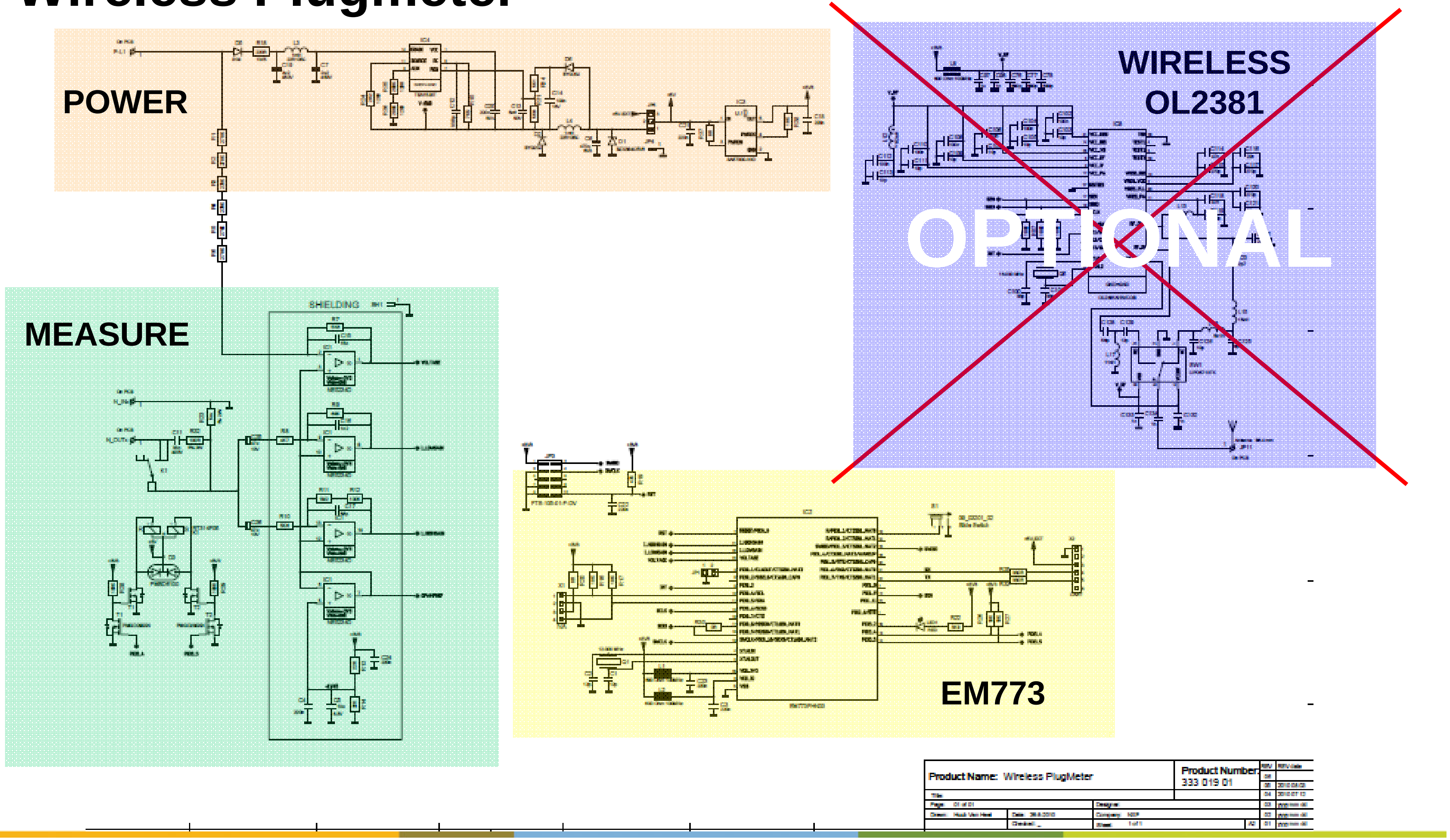
Circuitry	Device Type	Quantity	Remark
Power Supply (not needed if regulated 3.3V VDD is available)	AC to DC SMPS Voltage Reguilator	1 1	TEA1520T SA57000-33D
Current Measurement (2 current ranges)	Shunt Resistor Operational Amplifier	1 2	 NE5234 (4 inside single IC)
Voltage Measurement	Resistor Divider Operational Amplifier	2 1	 NE5234 (4 inside single IC)
Oscillator	Crystal	1	12 MHz



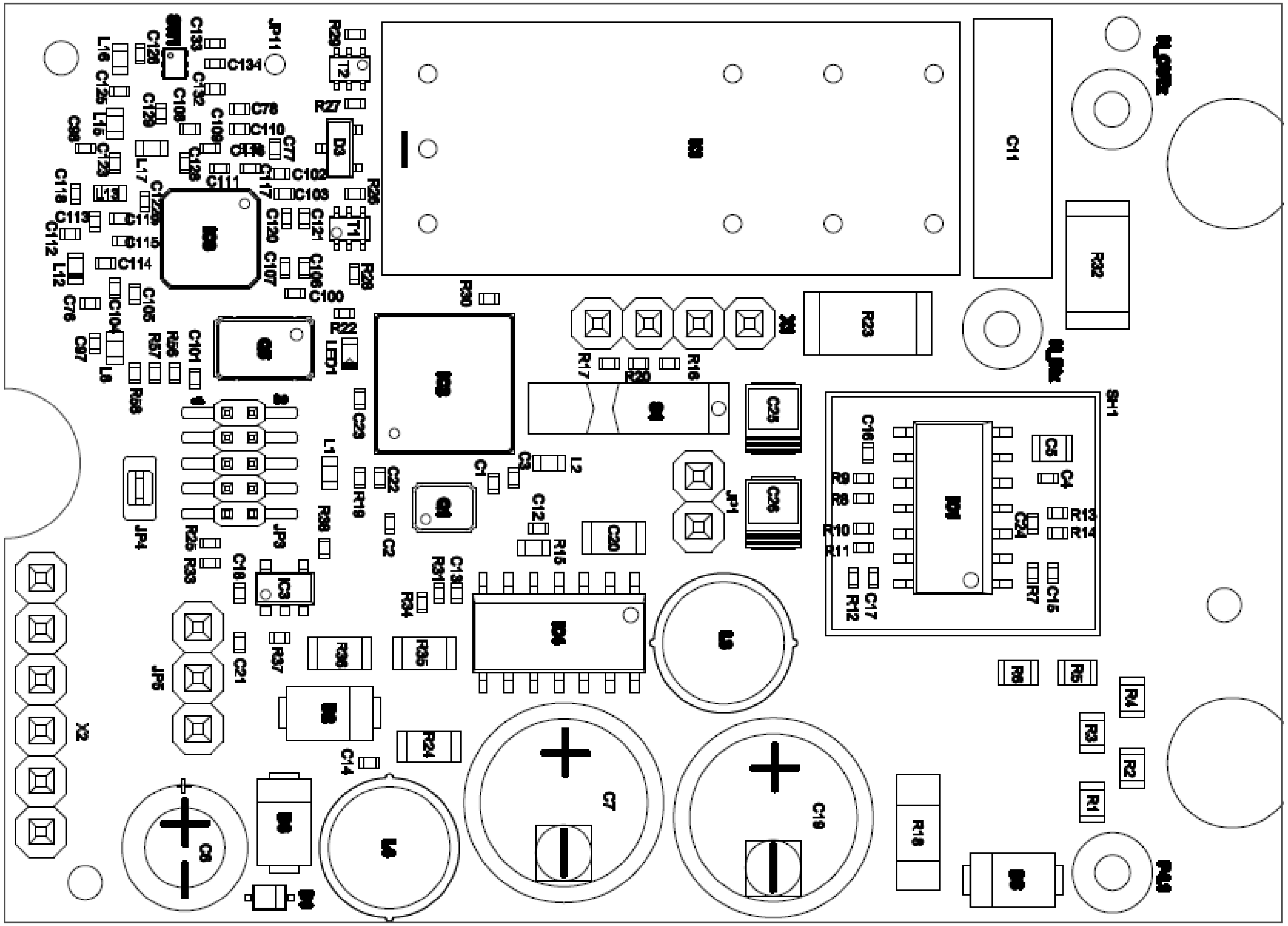
Energy Metering IC EM773: Circuit Example
Wireless Plugmeter



Energy Metering IC EM773: Circuit Example Wireless Plugmeter



Energy Metering IC EM773: Layout Example
Wireless Plugmeter



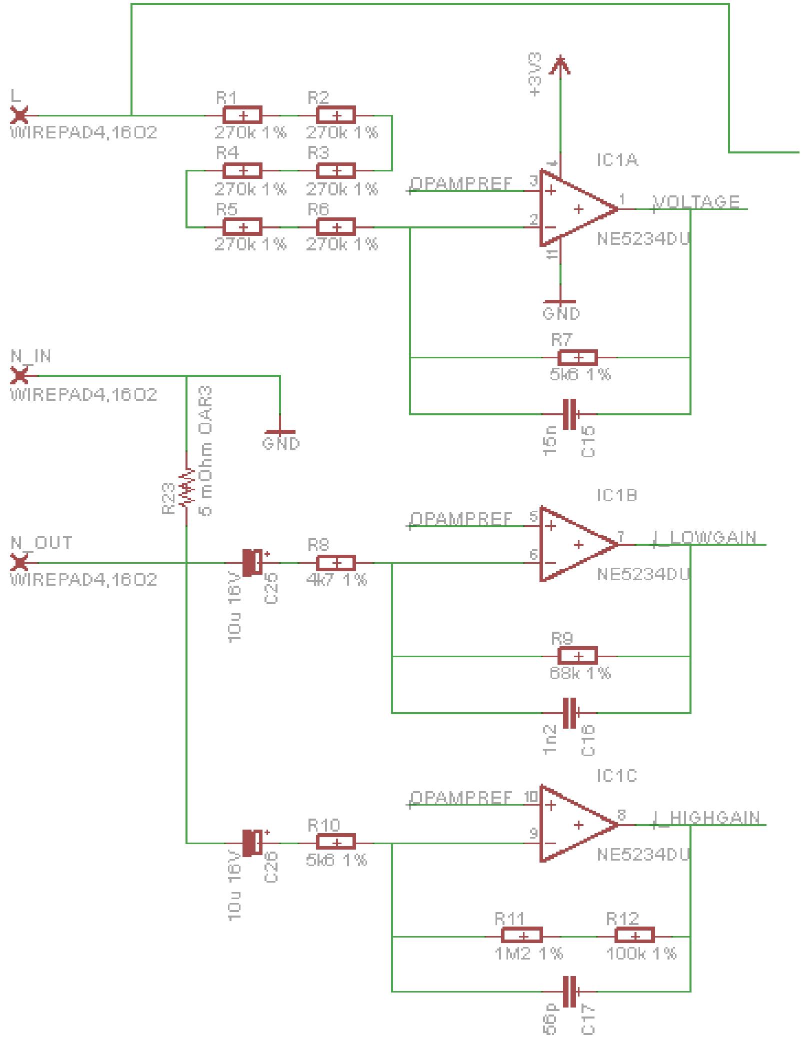
EM773 Analog Input Circuit Options

<u>Voltage measurement circuit</u>

- Voltage Divider
  - Cost efficient solution
- Transformer
  - Full galvanic isolation

<u>Current measurement circuit</u>

- Shunt Resistor
  - Series resistor with defined low resistance
  - Cost efficient solution
- Current Transformer
  - Low dissipation at high current
  - Standalone or PCB mounted



Energy Metering IC EM773
Definition of API calibration input values

- API Input: Voltage and Current
  - Start Metrology Engine with standard settings for Vpp, I1pp and I2pp derived from input circuits
  - Measure voltage and current from a calibrated source
  - Correct the voltage and current ranges Vpp, I1pp and I2pp with the relative difference

- API Input: Phase Correction
  - Measure the phase difference between voltage and current channels for two resistive loads (high current and low current) and enter this value Phi1 and Phi2 as the required phase correction

Energy Metering IC EM773
Calibration

- Real time calibration adjustments
  - The calibration parameters can be adjusted real-time by the application running on the Cortex M0 processor core.
  - This feature can for example be used to implement temperature compensation or to switch between different measurement inputs

EM773 Calibration API

17.4.1 Metrology ranges

The following structure is used to configure the voltage range, current ranges and the phase corrections for the I_HIGHGAIN and I_LOWGAIN gain current channels:

```
typedef struct metrology_ranges_tag
{
float Vpp;
float I1pp;
float I2pp;
float DeltaPhi1;
float DeltaPhi2;
} metrology_ranges_t;
```

EM773 Measurement Output API

```
typedef struct metrology_result_tag
{
float V;
float I;
float P;
float Q1;
float S;
float S1;
float PF;
float PF1;
float SN;
float N;
float THDI;
} metrology_result_t;
```

EM773 Calling Metrology Engine Driver (1/3)

```
1. Initialize metrology engine:
metrology_init(12000000, 50);

2. Set ranges for the metrology engine:
metrology_ranges_t metrology_ranges;
metrology_ranges.Vpp = (float)954.67;
metrology_ranges.I1pp = (float)2.84;
metrology_ranges.I2pp = (float)45.60;
metrology_ranges.DeltaPhi1 = (float)0.0;
metrology_ranges.DeltaPhi2 = (float)0.0;
metrology_set_ranges(&metrology_ranges);

3. Start the metrology engine:
metrology_start();
```

EM773 Calling Metrology Engine Driver (2/3)

```
4. Read the measured data:
while (running)
{
if (metrology_get_gainchannel() == CURRENT_CHANNEL1)
{
LED_ON(); /* signal measuring from I_HIGHGAIN */
}
else
{
LED_OFF();/* signal measuring from I_LOWGAIN */
}
if (metrology_read_data(&meter_result))
{
print_result(&meter_result);
}
ms_sleep(250);
}
```

EM773 Calling Metrology Engine Driver (3/3)

5. Stop the metrology engine:

```
metrology_stop();
```

EM773 Measurement Accuracy

RMS Voltage V	0.5 %
RMS Current I	0.5 % [1]
Active Power P	1.0 % [1,2]
Apparent Power S	1.0 % [1]
Non-active Power N	2.0 % [1]
Power Factor PF	2.0 % [1]
Fundamental Reactive Power Q1	2.0 % [1,3]
Fundamental Apparent Power S1	3.0 % [1]
Fundamental Power Factor PF1	4.0 % [1]
Non-fundamental Apparent Power SN	4.0 % [1]
Current Total Harmonic Distortion THDI	5.0 % [1,4]

1 For Ippmax/400 < Ipp < Ippmax

2 Crosstalk in P from Q1 < 0.1% of Q1

3 Crosstalk in Q1 from P < 0.1 % of P

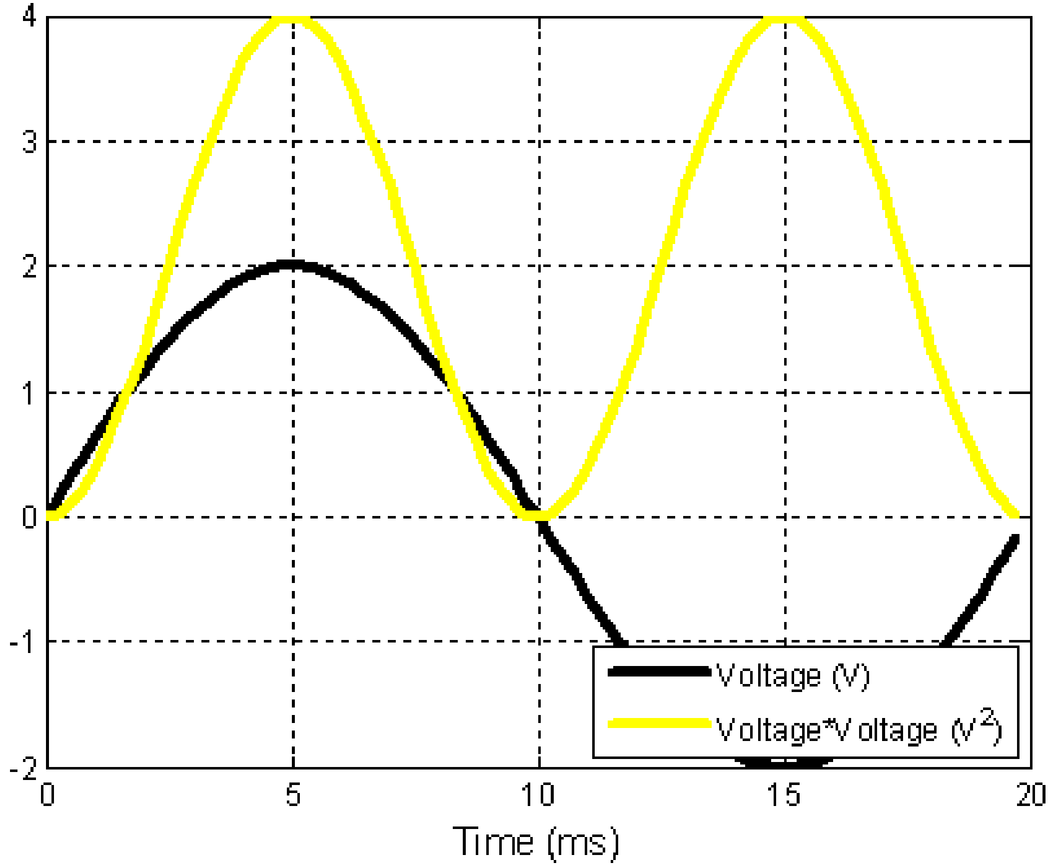
4 For THDV < 5 % and THDI > 40 %

Electricity Measurement Algorithms

Output for sinusoidal and non-sinusoidal voltage and current

1. RMS Voltage (V)

$$V = \sqrt{\frac{1}{kT}\int_{\tau}^{\tau+kT} v^2 dt}$$

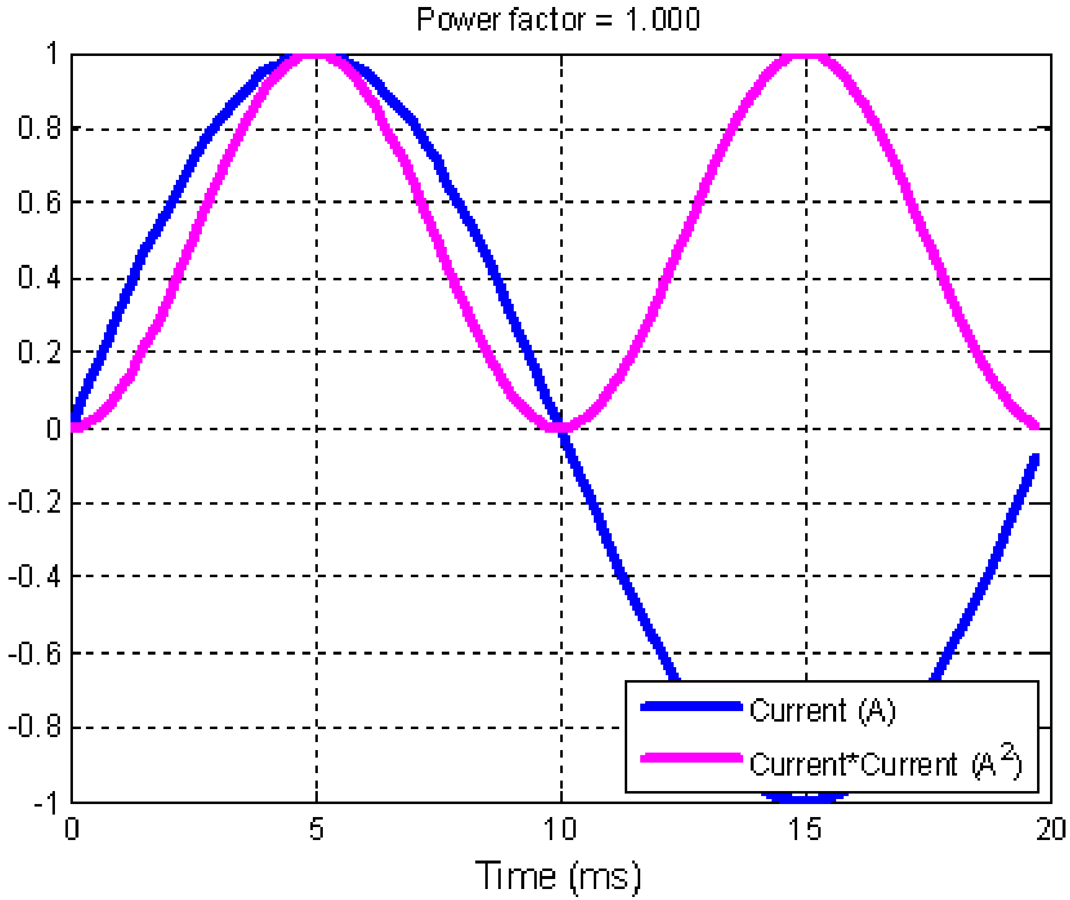


Electricity Measurement Algorithms

Output for sinusoidal and non-sinusoidal voltage and current

2. RMS Current (I)

$$I = \sqrt{\frac{1}{kT}\int_{\tau}^{\tau+kT} i^2 dt}$$

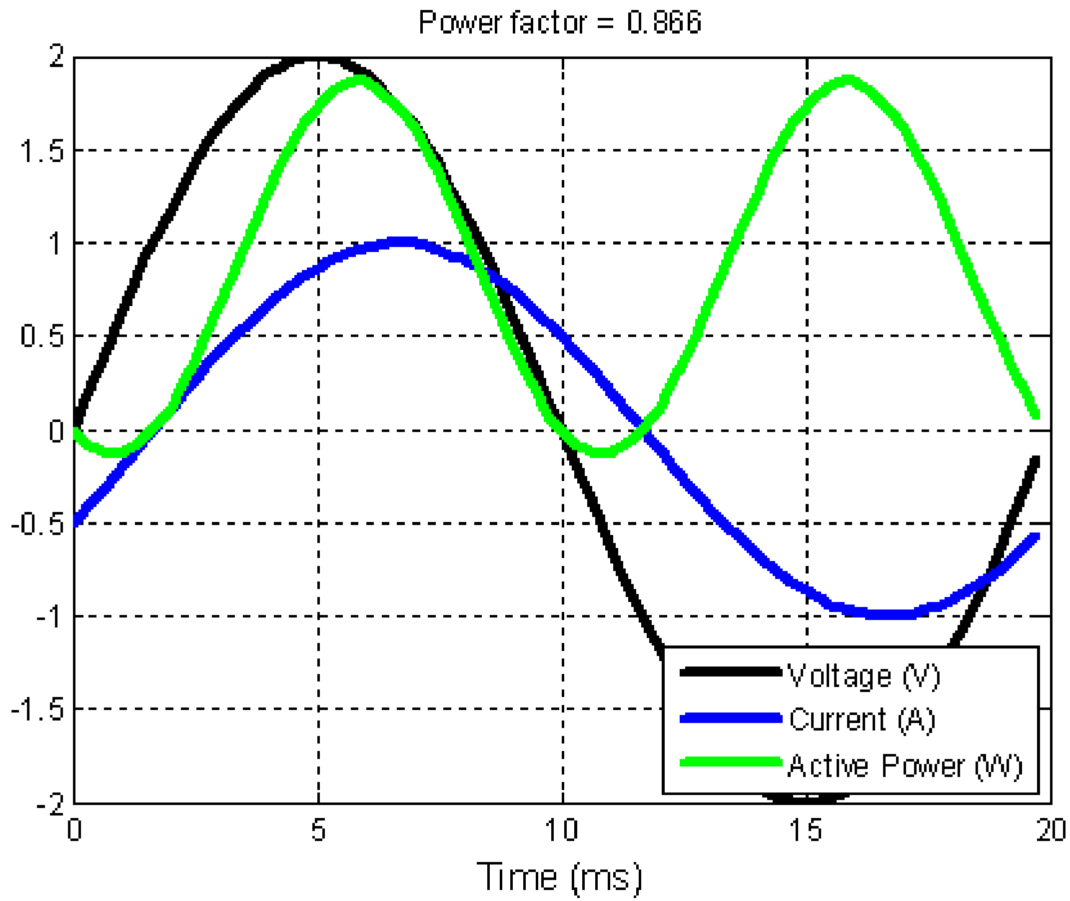


Electricity Measurement Algorithms

Output for sinusoidal and non-sinusoidal voltage and current

3. Active Power (P)

$$P = \frac{1}{kT} \int_{\tau}^{\tau + kT} vi\,dt$$



Electricity Measurement Algorithms

Output for sinusoidal and non-sinusoidal voltage and current

4. Apparent power (S) $$S = VI$$

5. Non-active power (N) $$N = \sqrt{S^2 - P^2}$$

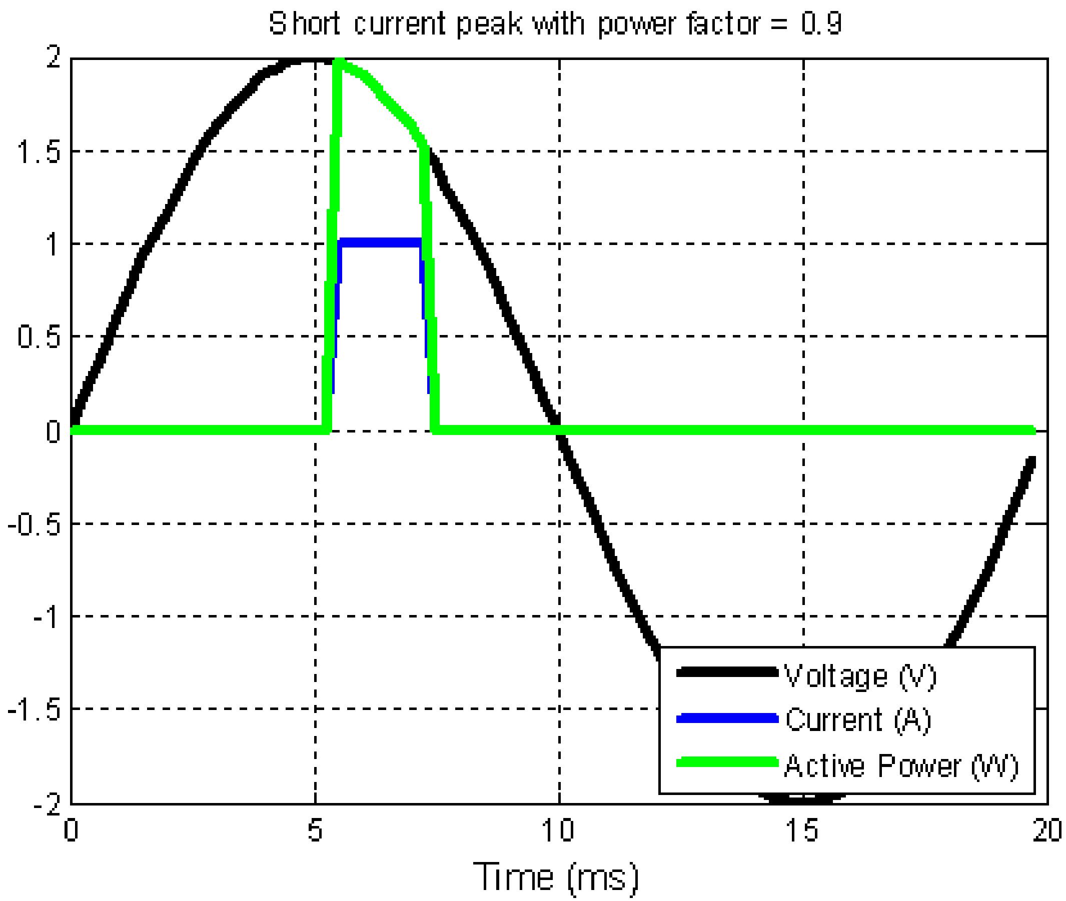
6. Power factor (PF) $$PF = \frac{P}{S}$$

Electricity Measurement Algorithms

Additional output for sinusoidal voltage

- Fundamental Active Power (P1)
- P1 = P
  (sinusoidal voltage)

$$P_1 = \frac{1}{kT} \int_{\tau}^{\tau+kT} vi\,dt$$



Electricity Measurement Algorithms
Additional output for sinusoidal voltage

7. Fundamental Reactive Power (Q1)

$$Q_1 = \frac{\omega}{kT} \int_{\tau}^{\tau+kT} i_1 \left[\int v_1 dt \right] dt$$

8. Fundamental apparent power (S1)

$$S_1 = \sqrt{P_1^{\,2} + Q_1^{\,2}}$$

9. Fundamental power factor (PF1)

$$PF_1 = \frac{P_1}{S_1}$$

10. Non-fundamental apparent power (SN)

$$S_N = \sqrt{S^2 - S_1^{\,2}}$$

11. Current total harmonic distortion (THDI)

$$THD_I = \frac{S_N}{S_1}$$

Electricity Measurement Algorithms

Energy calculations on EM773 CPU Core

12. Consumed Energy (T+)

$$T_{+} = \sum_{CalculationPeriod=1}^{M} P_{+} * T_{calculation}$$

13. Produced Energy (T-)

$$T_{-} = \sum_{CalculationPeriod=1}^{M} P_{-} * T_{calculation}$$

14. Total Energy (T)

$$T = \sum_{CalculationPeriod=1}^{M} P * T_{calculation}$$

EM773 Development Tools

EM773 Tool Highlights

- EM773 Starter Kit including:
  - Plug meter with EM773 and OL2381
  - USB transceiver with LPC1343 and OL2381
- EM773 FREE software examples
  - Downloadable from http://www.NXP.com/smartmetering
- IDEs Supporting EM773 Metrology Engine
  - IAR Embedded Workbench for ARM (NOW!!!)
  - Keil and Code Red (Coming soon!)
- SWD debuggers
  - All debuggers supporting Cortex-M0

EM773 Plug Meter

PC Energy Display

M-Bus USB Adapter

EM773 Online Community

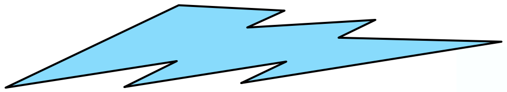
- NXP Smart Metering
  http://www.nxp.com/smartmetering
- EM773 Software Examples and Application Notes
  http://ics.nxp.com/support/design/microcontrollers/smart.metering/
- EM773 Online Support Forum
  http://knowledgebase.nxp.com/

EM773 FREE Software Examples in More Detail

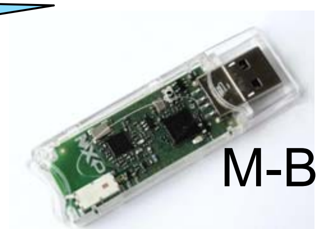
- FREE Software consists of 3 components
  1. EM773 project for plug meter examples
  2. LPC1343 project for USB wireless transceiver
  3. PC based application reading from USB receiver

Plug meters

Wireless M-Bus



PC Energy Display



M-Bus USB Adapter

EM773 FREE Software Examples in More Detail

1. EM773 project for plug meter examples

- Software Includes:
  - Metrology engine supported with closed source firmware driver
  - Open source ARM M0 application using metrology engine API
  - Open source wireless M-Bus for transmitting power data
  - Open source UART and I2C wired metrology output options
- Software Example Dependencies:
  - IAR Embedded Workbench 5.50+ (Keil and Code Red in roadmap)
  - FreeRTOS (Metrology Engine can be used independently)
- Application Ideas
  - Other communication options possible such as PLC, Zigbee
  - Local UART, I2C, SPI serial ports for data output and control
  - ARM Cortex M0 may be used for data encryption (AES, 3DES, etc)

Wireless Plug Meter Product Specifications

Voltage Range	90 – 270 V
Net Frequency	50 or 60 Hz
Maximum Current	16 A
Voltage Accuracy	Better than 0.5%*
Current Accuracy	Better than 0.5%*
Active Power Accuracy	Better than 1.0%*
Power consumption	< 0.45 W
Dynamic Range	400:1
RF output power	10 mW
Wireless range	Up to 300 meters line of sight

* Based on dynamic range of 400:1

EM773 FREE Software Examples in More Detail

2. LPC1343 project for USB wireless transceiver

- Software Includes:
  - ► Open source ARM M3 application using USB device driver
  - ► Open source wireless M-Bus for receiving power data from up to 50 plug meters simultaneously (can be configured for more)
  - ► All software included is open source c code

- Software Example Dependencies:
  - ► IAR Embedded Workbench 5.50+ (Keil and Code Red in roadmap)
  - ► FreeRTOS (USB and M-Bus can be used independently)

- Application Ideas
  - Bi-directional communication for sending commands to meters
  - Web based application bridge for metrology data

EM773 FREE Software Examples in More Detail

3. PC based application reading from USB receiver

- Software Includes:
  - Open source PC application using USB device driver
- Software Example Dependencies:
  - Microsoft Windows XP, Vista or Windows 7
  - Microsoft .NET 4.0 or newer
- Application Ideas
  - Improved graphical user interface (GUI) options
  - Web server for remote data access
  - Data logging and storage
  - List of ideas can go on and on…

Get Started Today!!!

1. Order an Energy Metering IC starter kit
2. Download FREE software examples and application notes
3. Run the demo application with a variety of different loads
4. Connect your debugger and IDE for software development
5. Use pin holes on the plug meter to evaluate custom components with the EM773
6. Design your own EM773 prototype and product