

THIS SPEC IS OBSOLETE

Spec No: 002-05181

Spec Title: AN205181 - F2MC-8FX Family MB95410H/470H Series One Phase Power Meter CS5464 Operation

Replaced by: None

AN205181

F²MC-8FX Family MB95410H/470H Series One Phase Power Meter CS5464 Operation

This application note describes how to use One Phase Power Meter (CS5464) solution's energy measure chip-CS5464. It is designed to accurately measure instantaneous current and voltage and calculate VRMS, IRMS, instantaneous power, active power, and reactive power for high-performance power measurement applications.

OBSOLETE

Contents

1 Introduction....1
2 Background....1
2.1 Overview....1
2.2 Features of CS5464....2
2.3 Function diagram....2
3 HW Diagram....3
3.1 The HW diagram of energy measure....3
4 HW Reference SCH....3
5 FW diagram....4
5.1 SPI Communication Process....4
5.2 Firmware System Diagram....5
6 FW Function List....5
6.1 API....5
6.2 HAL....7
7 Additional Information....7
Document History....8

1 Introduction

This application note describes how to use One Phase Power Meter (CS5464) solution's energy measure chip-CS5464.

Chapter 2 explains the background of CS5464.

Chapter 3 explains the HW diagram of energy measure function.

Chapter 4 explains the HW reference SCH.

Chapter 5 explains the FW diagram.

Chapter 6 explains the FW function list.

2 Background

Background of CS5464

2.1 Overview

The CS5464 is an integrated power measurement device which combines three $\Delta\Sigma$ analog-to-digital converters, power calculation engine, energy-to-frequency converter, and a serial interface on a single chip. It is designed to accurately measure instantaneous current and voltage and calculate VRMS, IRMS, instantaneous power, active power, apparent power, and reactive power for high-performance power measurement applications.

The CS5464 is optimized to interface to shunt resistors or current transformers for current measurement and to resistive dividers or potential transformers for voltage measurement.

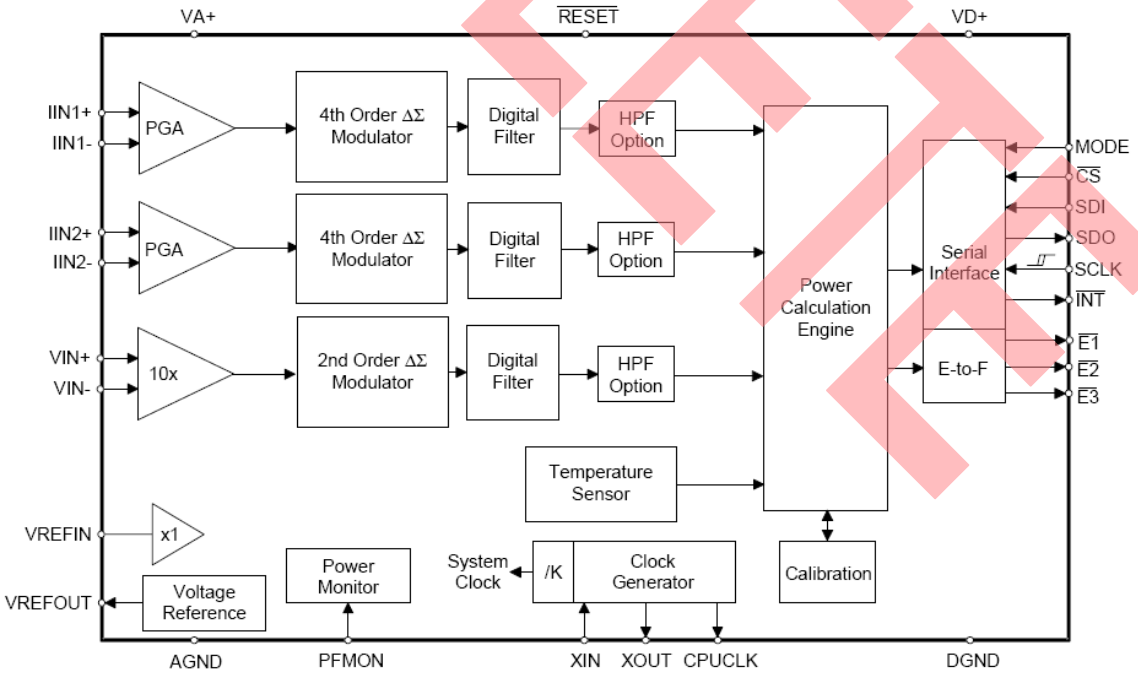
The CS5464 features a tamper detection scheme that uses the larger of the active power measurements to register energy. Additional features include system-level calibration, temperature sensor, voltage sag & current fault detection, and phase compensation.

2.2 Features of CS5464

- Energy Data Linearity: ±0.1% of Reading over 1000:1 Dynamic Range
- On-chip Functions:
  - Instantaneous Voltage, Current, and Power
  - IRMS and VRMS, Active, Reactive, and Apparent Power
  - Current Fault and Voltage Sag Detect
  - System Calibrations / Phase Compensation
  - Temperature Sensor
  - Energy-to-pulse Conversion
  - Positive-only Accumulation Mode
- Meets Accuracy Spec for IEC, ANSI, & JIS
- Low Power Consumption
- Tamper Detection in 2-Wire Distribution Systems
- GND-referenced Signals with Single Supply
- On-chip 2.5 V Reference (25 ppm/°C typ)
- Power Supply Monitor
- Simple Three-wire Digital Serial Interface
- “Auto-boot” Mode from Serial E2PROM
- Power Supply Configurations: VA+ = +5 V; AGND = 0 V; VD+ = +3.3 V to +5 V

2.3 Function diagram

Figure 1. The Function Diagram of CS5464



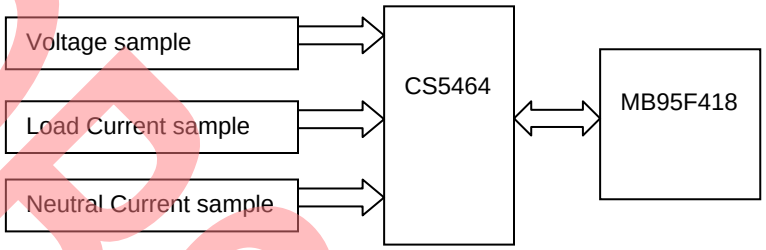
OBSOLETE

3 HW Diagram

Hardware diagram of energy measure system

3.1 The HW diagram of energy measure

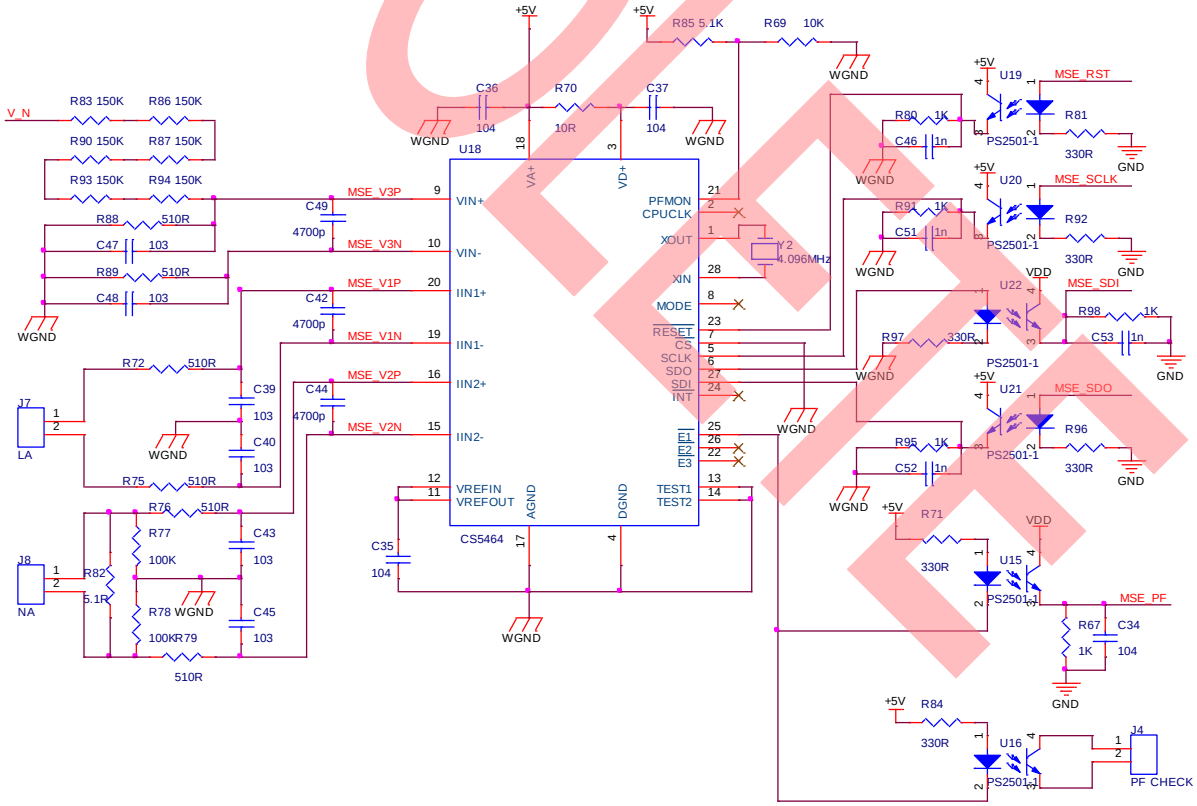
Figure 2. Hardware diagram



4 HW Reference SCH

Hardware reference SCH of energy measure system

Figure 3. RCS5464 Reference SCH



OBSOLETE

5 FW diagram

Firmware system diagram of energy measure system

5.1 SPI Communication Process

Figure 4. SPI Write Process

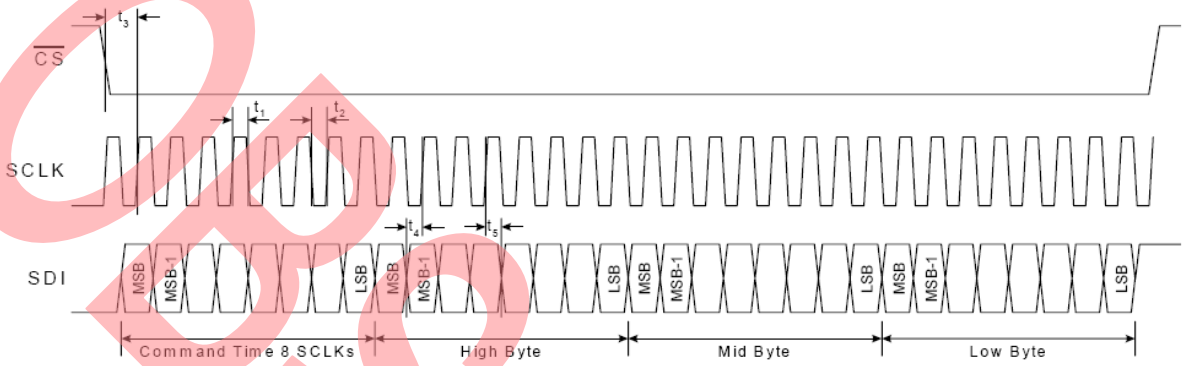
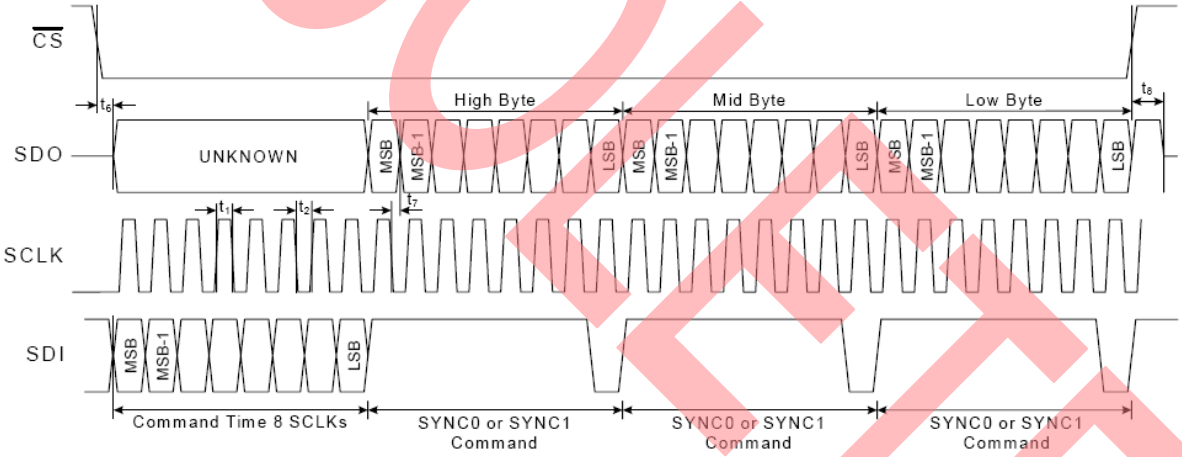


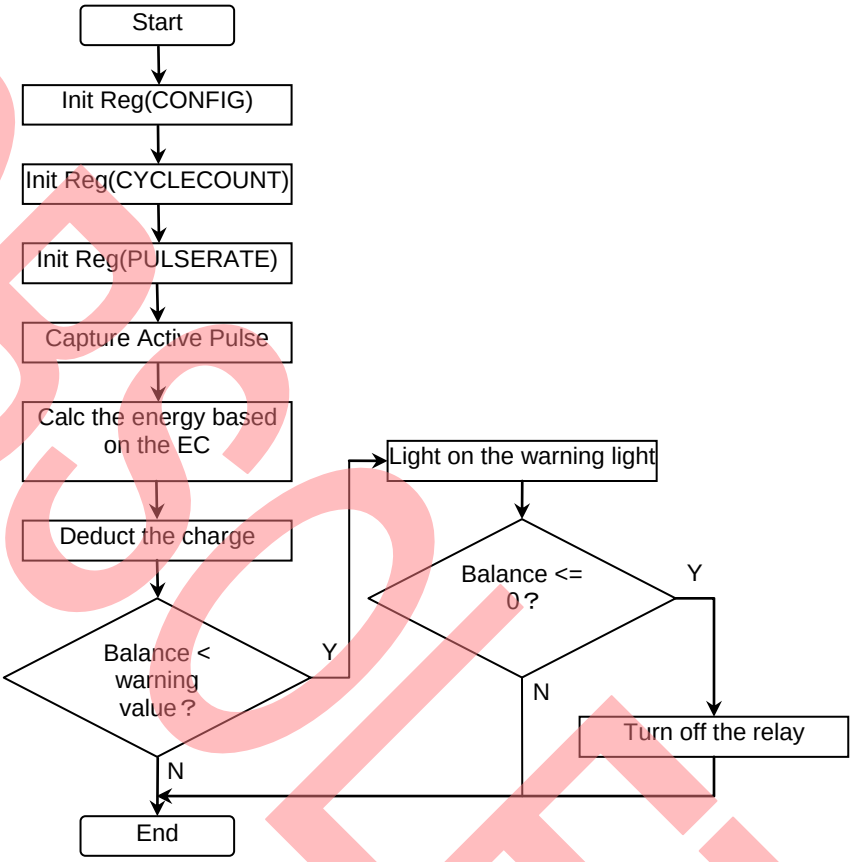
Figure 5. SPI Read Process



OBSOLETE

5.2 Firmware System Diagram

Figure 6. Firmware System Diagram



OBSOLETE

6 FW Function List

6.1 API

Table 1. FW API List

Function Prototype	Description
void CS5464_Reg_Init(void)	Initialize CS5464 register
void EnergyCount_Init(void)	Initialize for energy counting
void AfeRefreshEnable(void)	AFE refresh enable
void AfeRefreshTask(void)	Refresh AFE setup with a reset and re-configuration
void EnergyCount_Reset(void)	Reset all meter energy data
void CurEnergyCount_Reset(void)	Reset current meter energy data
void MeterKwhInc(void)	Accumulate meter KWH with each pulse received
void PowerEnergy(void)	Accumulate energy pulses & output calibration pulses
void GetMeterCurFwdEnergySum(INT8U *sumBuff)	Retrieve current sum of meter forward energy data

Function Prototype	Description
void GetMeterCurRevEnergySum(INT8U *sumBuff)	Retrieve current sum of meter reverse energy data
void GetMeterCurAllEnergySum(INT8U *sumBuff)	Retrieve current sum of meter total energy data
void ReadMeterRuntimeData(void)	Read meter energy data saved in EEPROM
void SaveMeterRuntimeData(void)	Save meter energy data to EEPROM
void ReadMeterConst(void)	Read meter constant from EEPROM
void ReadMeterClibData(void)	Read meter clibration data from EEPROM
void SaveMeterClibData(void)	Write meter calibration data to EEPROM
void LoadTimeSegInfo(void)	Load or reset time-seg parameter in EEPROM
void UpdateTimeSegTable(INT8U tableId, INT8U segCount, INT8U *segData)	Update time-seg table
INT8U FindTimeZoneSegTableSeriesId(INT8U idType)	Find a time-zone series ID base on current calendar
INT8U FindTimeZoneSegTableId(INT8U idType)	Find a time-zone-ID and time-seg-table-ID base on current calendar
INT8U FindTimeSegTariffId(INT8U hh, INT8U mm)	Find a time-seg tariff ID base on given time value hh(hour) & mm(minute)
void ValidateTimeZoneId(void)	Validate time-zone ID base on current calendar
void ValidateTimeSegTableId(void)	Validate time-seg table ID base on current calendar
void SaveLastMonthAllEnergyCounter(void)	Save last month's all energy counter value to EEPROM
void SaveLastMonthFwdEnergyCounter(void)	Save last month's forward energy counter value to EEPROM
void SaveLastMonthRevEnergyCounter(void)	Save last month's reverse energy counter value to EEPROM
void SaveCurrentEnergyCounter(void)	Save current energy counter value base on tariff ID
void LoadEnergyCounter(void)	Load meter energy counter base on energy direction state
void EnergyDirValidate(void)	Validate energy direction/state
void GetTimeSegFwdEnergySum(INT8U *sum, INT8U monthId, INT8U segId)	Get forward energy sum for a specific time segment
void GetTimeSegRevEnergySum(INT8U *sum, INT8U monthId, INT8U segId)	Get reverse energy sum for a specific time segment
void GetTimeSegAllEnergySum(INT8U *sum, INT8U monthId, INT8U segId)	Get energy forward+reversed sum for a specific time segment
void UpdateTimeSegTariffId(void)	Update tariff ID based on current time segment
INT32U GetVoltageRmsValue(void)	Get scaled voltage reading
INT32U GetLoadCurRmsValue(void)	Get scaled load current reading
INT32U GetNeutralCurRmsValue(void)	Get scaled neutral current reading
INT32U GetActivePowerValue(void)	Get scaled active power reading
INT32U GetPowerFactorValue(void)	Get scaled power factor reading
void ResetEnergyParameter(void)	Reset energy parameters
void Read_I_V_Energy(void)	Read AFE I/V energy data
void CS5464_StartConversion(INT8U mode)	CS5464 starts to conversion

OBSOLETE

6.2 HAL

Table 2. FW HAL List

Function Prototype	Description
void SPIPortInit(void)	Initialize SPI port
void BcdEnergySumIntInc(INT8U *bcd)	Increment a 4 byte BCD energy number by 1 from integer part(bcd[2])
void BcdEnergySumDecimalAcc(INT8U *bcd, INT8U accVal)	Accumulate a 4 byte BCD energy number
void CS5464_ReadRegister(INT8U page, INT8U regAddr, INT8U *buff)	Read CS5464 register value
INT8U SPI_ReadByte(void)	Byte in through SPI port
void CS5464_WriteRegister(INT8U page, INT8U regAddr, INT8U *buff)	Write CS5464 register value
void SPI_SendByte(INT8U outDat)	Byte out through SPI port
void CS5464_WaitForReady(void)	CS5464 ready-status wait
INT8U CS5464_IsReady(void)	CS5464 ready-status check
void CS5464_SetRegPage(INT8U page)	CS5464 set register page for read/write

7 Additional Information

For more Information on Cypress semiconductor products, visit the following website:

www.cypress.com/documentation/application-notes/mb95410h470h-one-phase-power-meter-cs5464-solution-s-energy-measure

OBSOLETE

Document History

Document Title: AN205181 - F²MC-8FX Family MB95410H/470H Series One Phase Power Meter CS5464 Operation

Document Number: 002-05181

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	HAUL	06/02/2011	Initial release.
*A	5259065	HAUL	06/28/2016	Migrated Spansion Application note from MCU-AN-500120-E-10 to Cypress format. Hardware no longer exist and this AN to be Obsolete.

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2011-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

OBSOLETE