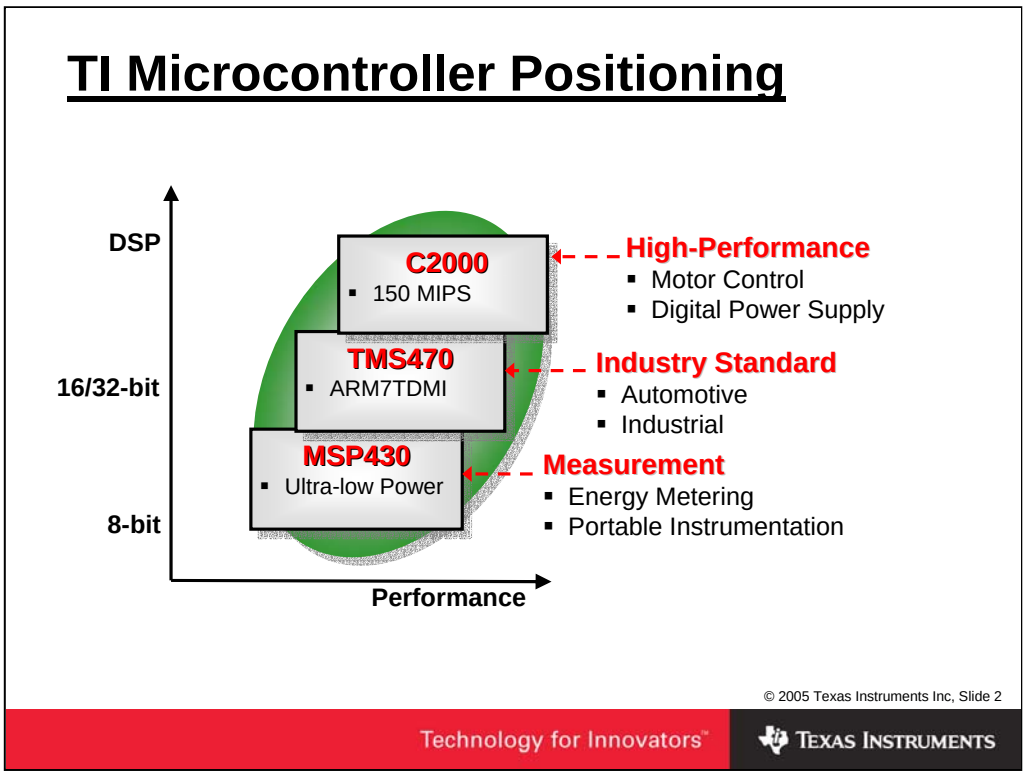
TMS470 Introduction



The overall embedded processor market TAM including DSP is approximately $20B.

AEC addresses the MCU portion of the TAM with three distinct product families optimized for each segment;

The C2000 family addresses the high end of the 32-bit MCU and low-end of the DSP market. The primary applications areas targeted are motor control and digital power supply. Up to 150 32-bit high-quality DSP MIPS are offered.

The TMS470 is based on and industry standard ARM7TDMI and was originally developed to target to 32/16-bit automotive applications. A focused subset of the product line is now available to the general purpose market for a wide range of applications that include industrial systems, medical instrumentation, consumer electronics, data processing and many other general-purpose embedded applications.

The MSP430 family targets 8-bit and low-end 16-bit ultra-low power applications. The product line offers end-equipment specific metering, instrumentation and sensing solutions and is also used throughout a wide range of applications. The product is noted specifically for lower power with devices starting as low as $0.49.

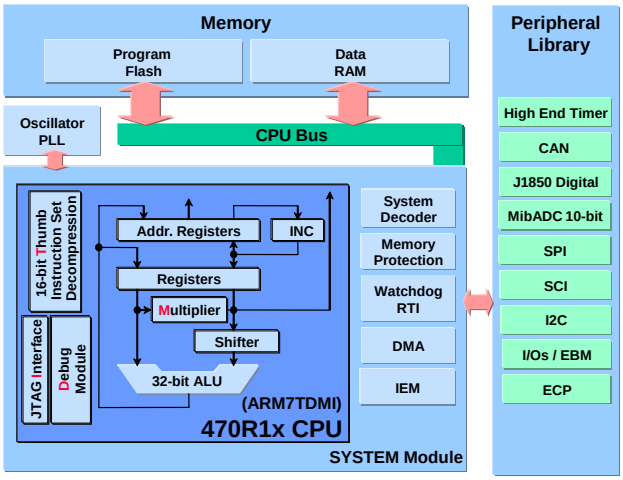
What Is The TMS470 MCU?

ARM7TDMI™

- Thumb extension
- Debug module
- Multiplier
- JTAG Interface

Performance

- Up to 60MHz
- 64K-to-1MB Flash
- 4-to-64KB RAM
- Peripheral library



Open Architecture with Full 3rd Party Support

The TMS470 combines peripherals and industry standard ARM7TDMI core operating at up to 60MHz.

Thumb Extension provides 16 bit instructions for high code density and allows for unlimited switches between instruction sets (16/32 bit). It includes run-time decompression.

Built-in Debug speeds application development and allow lower cost in-circuit emulation.

The multiplier allows high-performance signal processing.

JTAG Interface has access to all internal registers, execution of all CPU instructions, and memory access

To this standard ARM7TDMI core TI adds enhanced peripherals and a wide range of Flash memory options from 64kB to 1MB. As an industry standard architecture a wide range of third party tools and support is already in place.

Why ARM7TDMI?

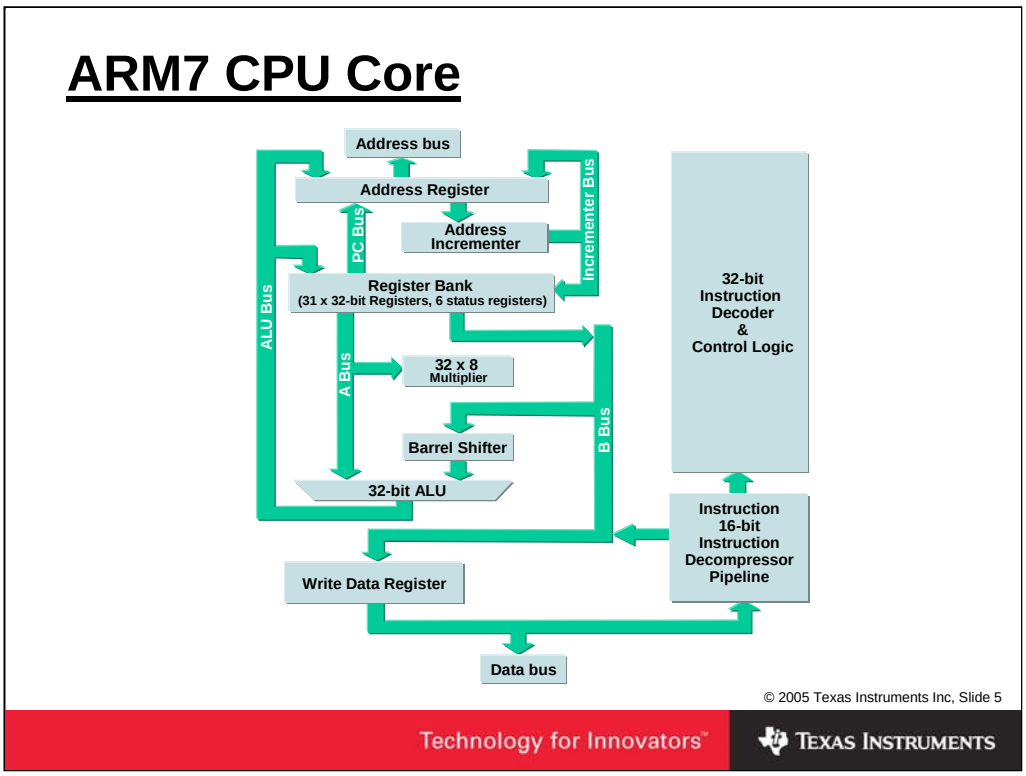
- **Industries most popular 32-bit MCU core**
  - Open Architecture with widest range of 3rd Party Support
  - Small ~32k gate core for cost effective integration
  - Flexibility - range of performance, peripherals, and power options offered
- **Two independent RISC instruction sets**
  - 32 bit instructions for fast execution
  - 16 bit instructions for high code density
- **Built-in Debug speeds application development**

ARM7 is an open architecture with the widest range of hardware and software tools support of any 32 bit MCU, including a wide range of all types of 3rd Party Support

The small ~32k gate core provides cost effective integration that enables production of ARM7TDMI MCUs from a variety of suppliers. This gives designers the opportunity to get 32 bit performance at close to 8 bit MCU prices

Design flexibility is greater than any proprietary MCU core because of its wide range of performance, peripherals, and power options.

Built-in debug speeds application development and allows lower cost in-circuit emulation.



The Texas Instruments TMS470R1x is the first of the TMS470 platforms of general-purpose 32-bit RISC microcontrollers. These microcontrollers offers high performance. High-end embedded control applications are demanding more performance from their controllers while still maintaining low costs. Many CISC (complex instruction set computer) cores are hitting their performance ceilings. Their large number of transistors tends to make them power-hungry, big, and expensive, as well as difficult to integrate, resulting in higher overall system cost. RISC (reduced instruction set computer) cores offer a potential solution to these problems. In the past, RISC processors often lost out to CISC processors because of poor code density, which required larger memory sizes and a consequence high system cost. The TMS470R1x RISC architecture offers the small die size and high performance needed in embedded applications. The code size problem has been addressed with extended architecture and 16-BIS, a new instruction set. Pipelining is employed so that all parts of the processing and memory systems can operate continuously. The Three Stage Pipeline has one instruction being executed, while its successor is being decoded, and a third instruction is being fetched from memory.

The 32/16-bit RISC architecture uses a 32-bit ARM instruction set for maximum performance and flexibility and 16-bit Thumb instruction set for increased code density. A 32-bit data bus carries both instructions and data. The three-stage pipeline, fetch – decode – execute, assure top execution speed. The 32-bit ALU arithmetic logic unit, performs all arithmetic computations, such as addition and all comparison operations. It sets and uses all conditional status flags. Multiplication is performed in the 32x8 multiplication unit. The barrel shifter is capable of shifting or rotating a data word by any number of bits in a single operation.

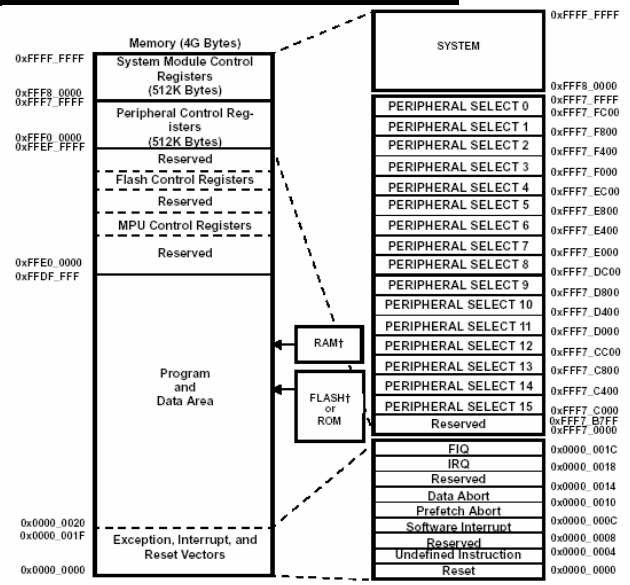
Two Instruction Set Advantage

- Speed versus code size optimization
- 16 BIS offers 25-50% code size reduction
- 32 BIS is 20-30% faster

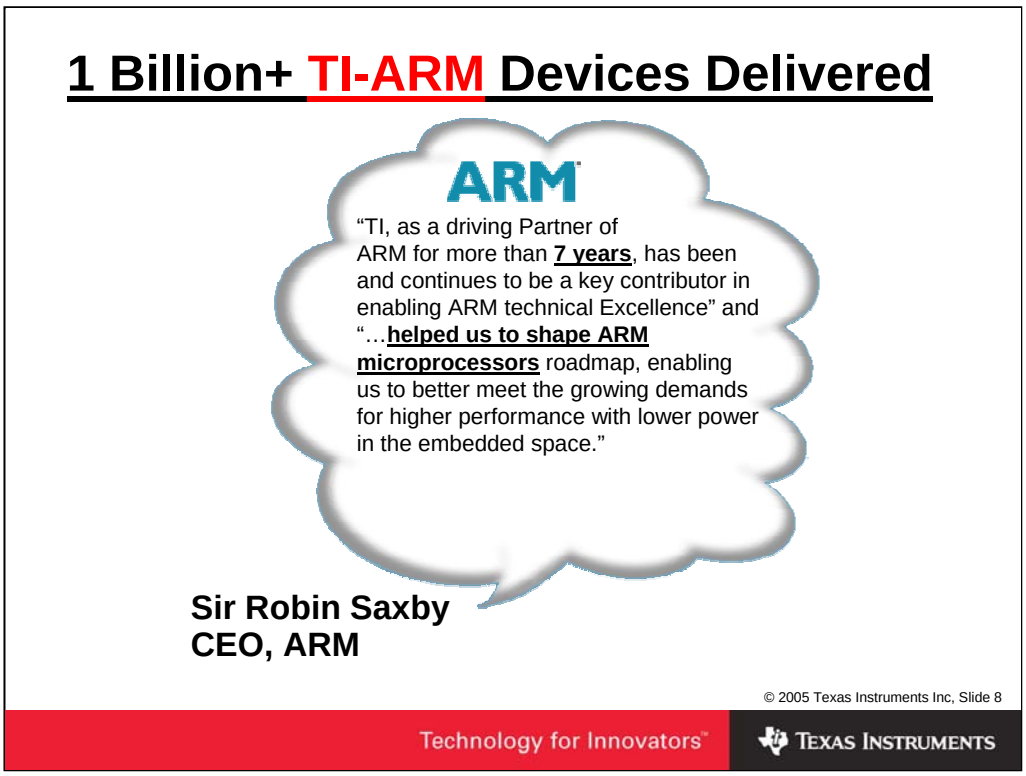
```
// In C code:
      int iabs(int x)
      {if (x>=0) return x;
      else return -x;}

; In ARM (12 bytes)
        CMP   R0,#0x0
        RSBMI R1,R1,#0x0
return BX    LR

; In Thumb (6 bytes)
        BPL   return
        NEG   R0, R0
return BX    LR

        Code optimized for size
```


ARM7TDMI has two independent RISC instruction sets that provide speed versus code size optimization. The ARM 32 BIS (Bit Instruction Set) is for fast code execution, and Thumb 16 BIS is for high code density. Benchmarks show that the 16 BIS has approx. 25-50% code size reduction over 32 BIS and that the 32 BIS is 20-30% faster than 16 BIS. Keeping code size under control is a key concern for designers coming from the 8 Bit MCU design background.

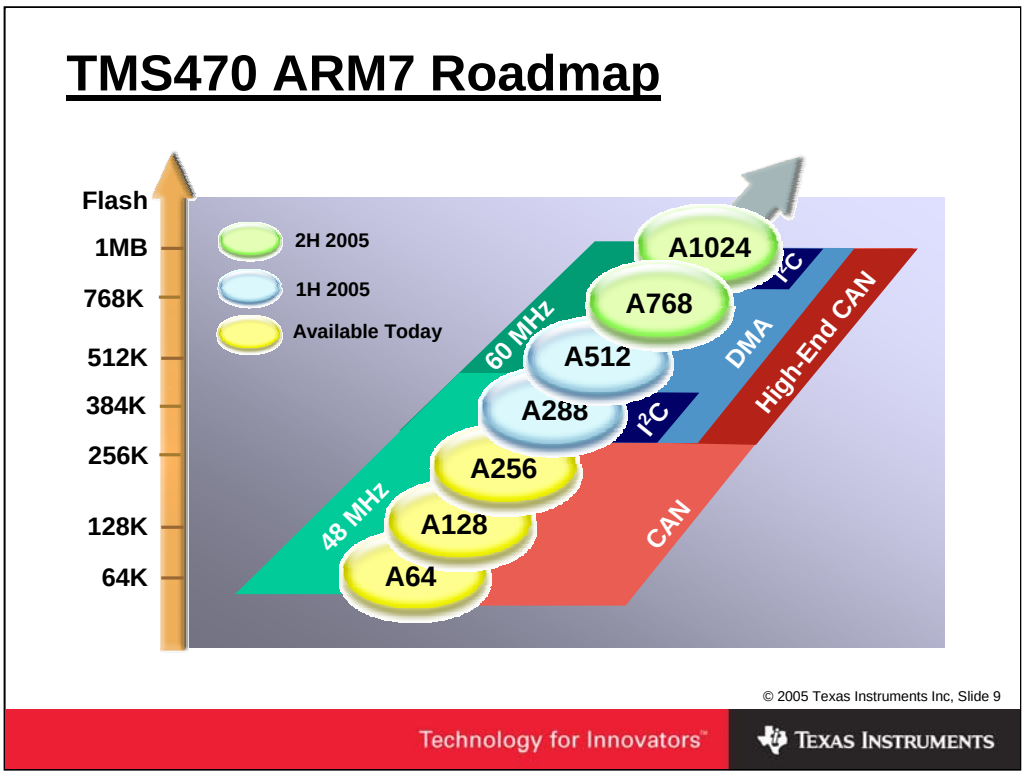
Unified 4GB Memory Map



The TMS470 provides a 4G byte Memory Map. The physical memory of the device (control registers, flash memory, ROM, and RAM) must be addressed within this memory map. The control registers are located at fixed addresses that are device specific. (See the device-specific data sheet for the mapping of peripherals.) 1K-byte sections of the memory map are reserved for each peripheral's control registers. The system module control registers and flash control registers are located in a reserved section of the upper 2M bytes of the memory map. System control registers (except for the MPU) are located from 0xFFFF_FFFF to 0xFFFF_FD00. In addition to the control registers, the exception vectors have a fixed map from 0x1F to 0x00. These vectors are shown above. These control registers and vector addresses are always valid. To access the peripheral control registers, the peripheral must be released from reset by setting PENABLE to 1 (PCR.0 = 1). There are exceptions that consist of reset, undefined instruction, software interrupt, prefetch, abort, data abort, IRQ, and FIQ. Each of these exceptions has its own vector.



TI has delivered more than 1 billion TI ARM-based devices in wireless, digital still camera, and automotive systems. The volume of products shipping and close relationship with ARM affords early next generation development. TI relationship with ARM Ltd. goes beyond the average partner. TI co-develops new ARM architectures with ARM Ltd. TI is driving new ARM technologies!



The scalable TMS470 ARM7 roadmap has memory options from 64kB to 1MB and a variety of integrated peripherals. The first devices launched in early 2005 include the A64, A128 and A256 operating at 48MHz. Later in 2005 additional 60MHz platform members will be introduced that include more memory and additional peripherals including I2C, DMA and an enhanced High-End CAN.

TMS470R1 Catalog Portfolio

TMS470	Flash KB	RAM KB	SPI/ SCI	Features	LQFP	1KU
A64	64	4	2/2	Timers, ADC, CAN	80	$4.95
A128	128	8	2/2	Timers, ADC, CAN	100	$5.95
A256	256	12	2/2	Timers, ADC, CAN	100	$6.85
A288	288	16	2/2	Timers, ADC, 2x CAN, I^2C, DMA, EBM	100/ 144	$8.95
A512	512	32	3/2	Timers, ADC, 2x High-end CAN, DMA	144	$9.95
A768	768	48	5/2	Timers, ADC, 3x High-end CAN, DMA	144	$13.95
A1024	1024	64	2/3	Timers, ADC, 2xSCC, High-end CAN (5) I^2C, DMA, EBM	144	To be announced

Available Today 1H 2005 2H 2005

The TMS470 offers the widest range of catalog ARM7TDMI Flash MCUs. All devices are fully software compatible.

Peripheral Descriptions

System Module (SYS)	Address Manager, Memory, Interrupt, Clock Controller, Watchdog and RTI
High End Timer (HET)	Coprocessor with 32 I/O channels for compare/capture, function generation
Standard CAN Controller (SCC) **High End CAN Controller (HECC)**	CAN 2.0B with 16 Message Objects CAN 2.0B with 32 Message Objects
MultiBuffer ADC (MibADC)	10-bit, 16 channel, MultiBuffer FIFO
Serial Peripheral Interface (SPI)	Data length up to 16 bits, 12.5MHz
Serial Comm Interface (SCI)	Baud rates up to 1.25Mbps
J1850 Digital (C2SI)	SAE J-1850 Class II Multiplex Scheme
EBM (External Bus Module)	Provides External Bus or GPIO
Inter IC Connect (I2C)	Multi-master I2C communication with any slave or master I2C-compatible device.
ECP (External Clock Prescale)	Output a prescaled clock signal
Direct Memory Controller (DMA)	Transfer data without CPU intervention

The System module provides an interface from the ARM CPU to the TI TMS470 platform of devices. The module defines the CPU bus and the expansion bus. The system module is responsible for memory interface and protection, interrupt prioritization, reset generation, and clock synthesis.

The High-End Timer (HET) is a software-controlled timer with a dedicated 32 channel programmable micromachine coprocessor that includes it's own set of twenty-one instructions. The HET micro machine is connected to a port of I/O pins.

The Controller Area Network (CAN) uses established protocol to communicate serially with other CAN controllers in harsh environments.

The Multi-Buffered Analog-to-Digital Converter (MibADC) accepts an analog signal and converts the signal to a 10-bit digital value at 1.55 µSec. per conversion. The MibADC can function in two modes: compatibility mode or buffered mode.

The Serial Peripheral Interface (SPI) provides a convenient method of serial interaction for high-speed communication between similar shift-register type devices.

The Serial Communication Interface (SCI) is a full-duplex, serial I/O interface, intended for asynchronous communication between the CPU and other peripherals utilizing the standard Non-Return-to-Zero (NRZ) format.

J1850 Digital (C2SI) The Class II Serial Interface A (C2SIa) is a communication module used for transmitting and receiving data over a multi-master network. The C2SIa module is the interface from the digital logic of the TMS470R1 platform of microcontrollers to an external analog interface chip.

The Expansion Bus Module (EBM) is a stand-alone module providing bond out for both general-purpose input/output (GIO) pins and expansion bus interface pins. EBM supports the multiplexing of the GIO functions and the expansion bus interface. When the GIO functions are not used, the EBM can be used to interface 8- or 16-bit memories.

The Inter-Integrated Circuit (I2C) module is a multi-master communication module providing an interface between the TMS470 microcontroller and an I2C-compatible device via the I2C serial bus. This module will support any slave or master I2C-compatible device.

The External Clock Prescaler (ECP) provides the TMS470R1 platform of devices with an external output clock (ECLK).

The Direct Memory Access (DMA) controller transfers data between address ranges in the memory map without intervention by the CPU, maximizing system performance.

Enhanced Peripheral Library

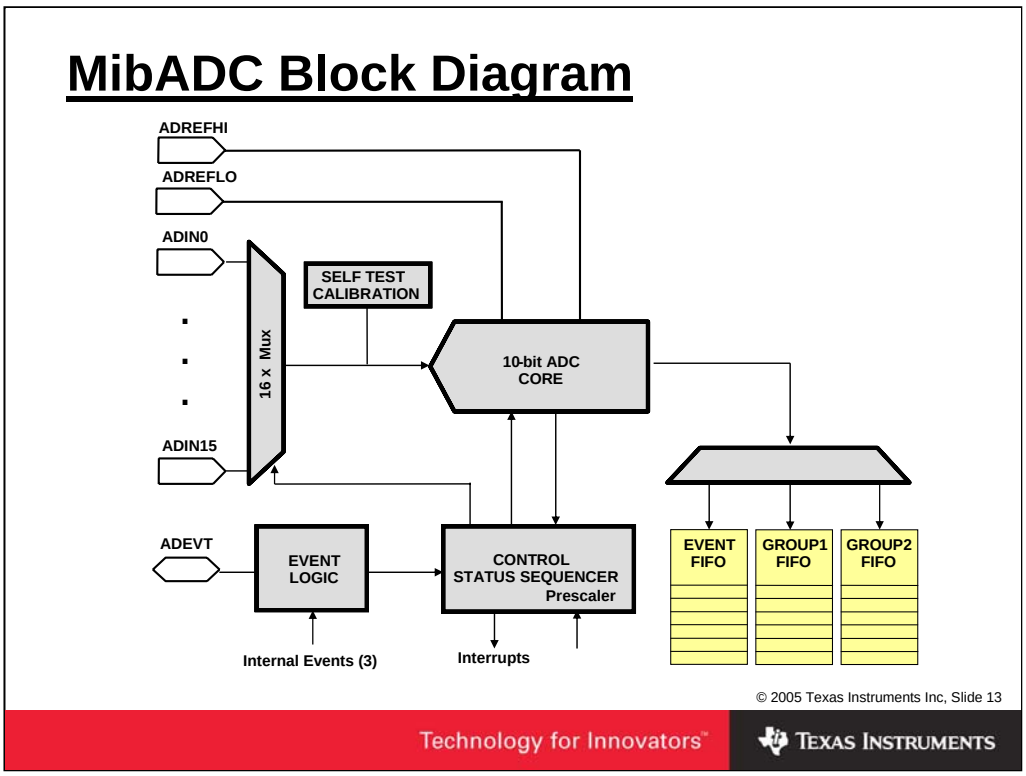
- **MibADC (MultiBuffer ADC)**
  - MultiBuffer FIFO RAM for results storage
  - 1.55µs, 10-bit resolution, up to 16 channels
- **High End Timer (HET)**
  - 32 channel programmable micromachine coprocessor
  - Capture, compare, PWM and complex functions generation
- **Controller Area Network (CAN)**
  - Standard CAN 2.0B Controller (SCC) with 16 message objects
  - High End CAN 2.0B Controller (HECC) with 32 message objects

- **Improve system performance - offload CPU**

The TMS470 complements the ARM7TDMI industry standard architecture with powerful enhanced peripherals. Specifically the peripherals reduce the loading on the CPU by automating common data handling activities.

The high-end timer for example provides a separate programmable micro machine that can generate automatically programmable complex wave forms in addition to common timing functions.

The MibADC integrates three independent FIFO's to automatically buffer conversion codes to reduce CPU overhead associated with servicing the ADC. Additionally the ADC offers external and internal conversion start triggering and programmable sample windows.



The MibADC is a 10-bit resolution ADC with 16 channels and conversions as fast as 1.55us. The MibADC includes features that reduce CPU load and increase overall system performance.

ADREFHI: Analog-to-digital converter external high reference. Any signal with a voltage higher than ADREFHI is recorded as 0x3FF

ADREFLO: Analog-to-digital converter external low reference. Any signal with a voltage lower than ADREFLO is recorded as 0x00

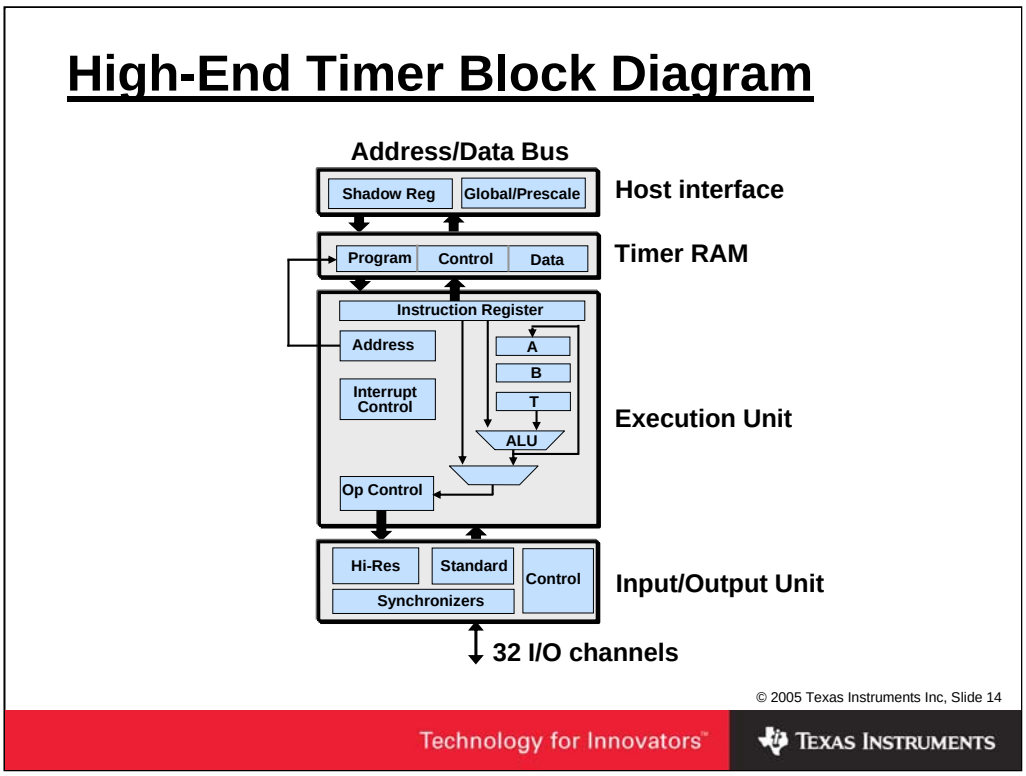
Calibration mode: A special active mode that allows an internal reference voltage to be connected to the ADC core input. While the calibration is selected, the input channel multiplexer is disabled and only the reference is available at the converter input.

Each buffer is readable from more that one register address (aliased) to allow effective use of the CPU capability to load multiple instructions to empty the buffer. For example, a read to any address between offset 0x20 and 0x3F pulls one conversion result from the event group buffer. If the CPU is used to read from a FIFO, the load multiple (LDM) instruction enables the loading of multiple registers from memory with back-to-back read operations.

Data can be read out of a FIFO either by the CPU or the DMA controller. Each FIFO may be read (either by the CPU or DMA) at the same time as a new conversion result is stored in it by the MibADC;

FIFO overrun occurs when a FIFO is full and the MibADC attempts to write data into the FIFO while no data is being read from the FIFO. If this occurs, the FIFO blocks the write (does not allow an overrun to occur) and goes into an overrun state. An interrupt request is generated, and the CPU must re-initialize the conversion group and its FIFO.

Sampling can be triggered by an external signal through the ADEVT pin



The HET is a third-generation Texas Instruments (TI) advanced intelligent timer module. This timer module provides sophisticated timing functions for real-time applications. The new High Resolution (HR) hardware channels allow greater accuracy for widely used timing functions such as period and pulse measurements, output compare, and PWMs. The HET has its own instruction set. This improves the versatility, resolution, and implementation of timer functions. The HET breakpoint feature, combined with various stop capabilities, facilitates easier software application debug. The timer consists of a specialized micromachine that executes instructions at the same speed as the system clock.

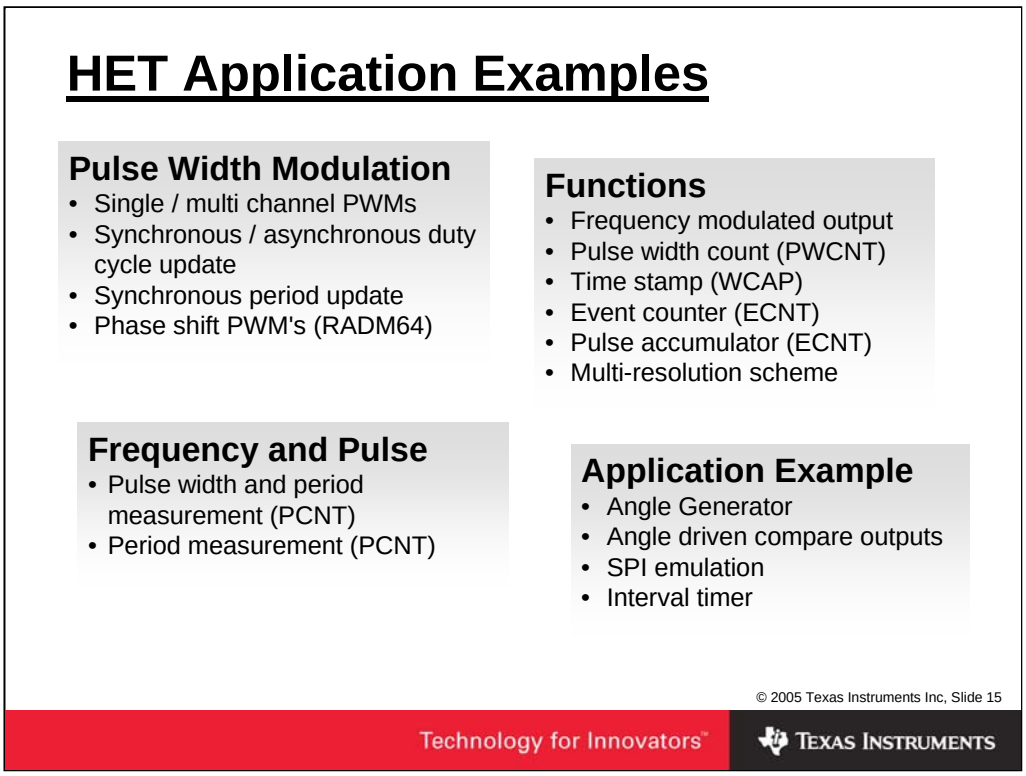
The HET is divided into 4 sections the Host Interface, Timer RAM, Execution Unit and Input/Output Unit:

The Host Interface controls all communications between timer-ram and CPU and is interfaced from the memory controller in the system module.

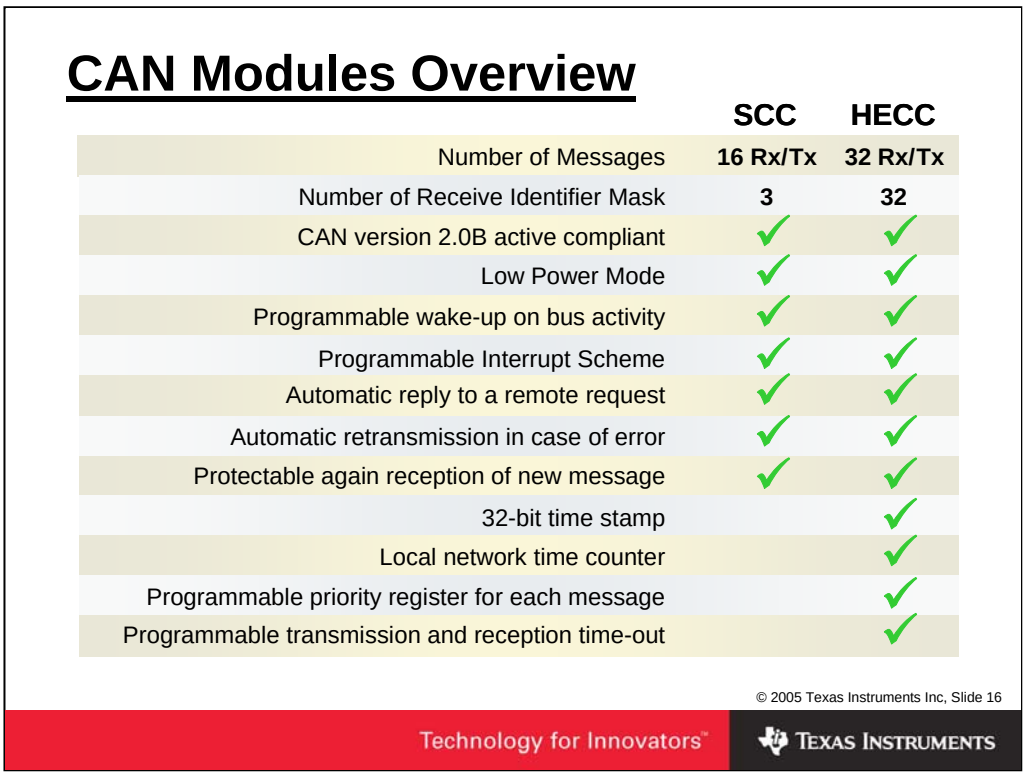
The HET RAM includes timer instructions (program and data). This memory is typically initialized by the CPU after reset before the timer starts execution. Once the timer program is loaded into the memory, the CPU starts the timer execution, and only data parameters can then be read or written into the timer memory.

The Execution Unit processes the instructions from the HET RAM. Operations performed in the ALU are additions (count), compares, and magnitude compares (higher or same). Each instruction includes an 8-bit field for specifying the address of the next instruction to be executed. An application program sequence is not controlled by a program counter (PC), but by the actual content of each instruction. This offers greater flexibility in going back and forth in the memory during program execution. System performance is improved by a wide instruction format (96 bits) that allows the CPU to fetch the operation code and data in one system cycle, thus increasing the speed at which data can be processed.

The Input/Output Unit interface is based on both HET RAM and control registers. The control registers include bits for selecting timer clock, configuring I/O pins, and controlling the timer module.



The HET has many advantages over timers found on many typical MCUs. In addition to classic time functions such as input capture or multiple PWMs, higher-level timer functions can be easily implemented in the timer program main loop. Higher-level timer functions include angle driven wave forms, angle and time-driven pulses, and input pulse width modulation (PWM) duty cycle measurement. Because of these high-level functions, data exchanges with the CPU are limited to the fundamental parameters of the application (periods, pulse widths, angle values, etc.); and the real-time constraints for parameter communication are dramatically minimized; for example, few interrupts are required and asynchronous parameter updates are allowed. The reduced instruction set and simple execution flow control make it simple and easy to develop and modify programs. Simple algorithms can embed all the flow control inside the HET program itself. More complex algorithms can take advantage of the CPU access to the HET RAM. With this, the CPU program can make calculations and can modify the timer program flow by changing the data and control fields of the HET RAM. CPU access to the HET RAM also improves the debug and development of timer programs. The CPU program can stop the HET and view the contents of the program, control, and data fields that reside in the HET RAM. Finally, the modular structure provides maximum flexibility to address a wide range of applications. The timer resolution can be selected from two cascaded prescalers to adjust the loop resolution and HR clocks. The 32 I/O pins can provide any combination of input, period or pulse capture, and output compare, including 24 HR channels. The standard memory structure allows module configuration from 64 words to 256 words of timer program memory.



The TMS470 CAN controller is available in two different implementations that are both fully compliant with the CAN protocol, version 2.0B. The two different CAN controller versions SCC (Standard CAN Controller) and HECC (High-End CAN Controller) use the same CAN protocol kernel module to perform the basic CAN protocol tasks according to the CAN protocol 2.0B. Only the message controller differs between the two CAN controller versions. The message controller is the interface between the CAN protocol kernel and the applications software.

The SCC internal memory supports 16, and the HECC internal memory supports 32 receive/transmit message objects. Each of these message objects contain message identifier information and up to eight data bytes. Three receive identifier masks are available in the SCC for standard CAN applications, and for more complex applications the HECC has 32. More available message objects and more identifier masks reduce the amount of CPU interaction required for an application.

Both SCC and HECC implement a low-power mode and can wake up the MCU on bus activity. A programmable interrupt scheme allows the application to react precisely to a wide variety of CAN controller and bus events, such as transmit acknowledge, abort acknowledge, received message pending, received message lost, wake-up, bus-off state and many others. The CAN controllers automatically reply to an incoming CAN RTR (Remote Transmission Request) message with a stored message object. If a message could not get delivered successfully, it will be automatically retransmitted until a proper acknowledge is received or a time-out event occurred (HECC only). Incoming messages can optionally be protected against being overwritten, thus keeping the older messages, should the CPU not be able to collect them in time.

Advanced timing features of the HECC improve usability in critical applications. Each of the 32 message objects gets a 32-bit time stamp on reception, thus making it easy to determine when a particular message was received, and the order of in which messages were received. The 32-bit time stamp gets derived from the local network time counter.

Also, with the HECC, priorities can be dynamically assigned to each transmit message object, allowing tight control of the order of the messages to be sent. A programmable transmission / reception timeout offloads the CPU and gives immediate feedback to the application about the communication status without requiring polling and implementing software counters.

Why Use CAN For Industrial Control?

	CAN	RS-485	Ethernet
Real Time Capable	Guaranteed in-time message delivery	No	No
Data Integrity	Guaranteed bus-wide error-free message delivery	No HW Error check	Not designed for harsh industrial environments
CPU Overhead	Low. Data integrity check & buffers in hardware	High software overhead for reliable communication	High Memory & MIPs required
Cost	Medium	Low	High
Throughput	1Mbit/s	10Mbit/s	1Gbit/s

The controller area network (CAN) uses a serial multimaster communication protocol that efficiently supports distributed real-time control, with a very high level of security, and a communication rate of up to 1 Mbps. The CAN bus is ideal for applications operating in noisy and harsh environments, such as in the automotive and other industrial fields that require reliable communication or multiplexed wiring. Prioritized messages of up to eight bytes in data length can be sent on a multimaster serial bus using an arbitration protocol and an error-detection mechanism for a high level of data integrity.

RS-485 can be implemented with SCI (UART). It has a Multi-master bus with high data throughput of up to 10Mbit/s. It is inexpensive to implement in hardware. However, it has no hardware checksum and requires software overhead for reliable communication. It also lacks real time capability.

Ethernet has very high data throughput at up to 1Gbit/s. It requires an expensive physical layer interface and has high software and MIPS requirements. It was not designed for harsh industrial environments and lacks real time capability.

CAN has a multi-master bus with data throughput of up to 1Mbit/s. It is extremely reliable and robust communication (CRC-16, bit stuffing, frame check), and is ideal for noisy and harsh environments. It is real-time capable, guaranteeing bus-wide error-free message delivery. The hardware configuration is also very configurable.

Enhanced: Atomic I/O Manipulation

```
GIODOUTx = GIODOUTx|MASK; // R-M-W of GIODOUTx

GIODSETx = MASK;          // Set pin w/o reading GIODOUT
```

- GIODSET /GIOCLR allows atomic I/O manipulation
- Dangerous read-modified-write not needed
- Reduced from 8+ to 3 cycles

Atomic I/O Manipulation means separate set and clear registers for GIO functions. This eliminates the need for a read modify write operation the reducing the chance of timing issues from the RMW method. Additionally this saves code space and reduces execution time.

When the output buffer is enabled, the value in the data output register (GIODOUTx) specifies the voltage applied to the external pin. The GIO module provides three ways of communicating with the data output control register;

The control register bit can be written directly by moving an appropriate value to the data output register.

The data output register bit can be set to 1 by using the data set register (GIODSETx).

The data output register bit can be cleared to 0 by using the data clear register (GIODCLRx).

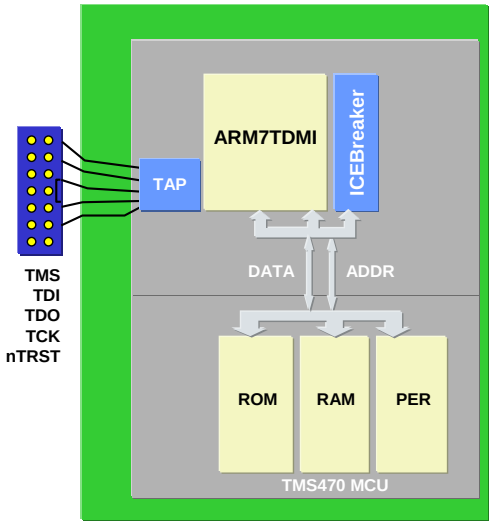
The GIODSETx and GIODCLRx registers allow improved handling of data minimizing the number of accesses to the peripheral to modify the output register and output pins. The data set and data clear registers remove any possibility of a read-modify-write (RMW) operation. RMW is possible when the CPU reads a register, performs some action (for example, an OR operation), and then writes the values back into the register. It is possible that the contents of the original register (GIODOUTx) can change (for example, an interrupt procedure) between the time when the CPU originally reads the register and the time when the CPU writes the new value. In this case, the new value written by the CPU overwrites the existing value, and the overwritten value is lost.

When the application needs to set or to reset n pins without changing the value of the others pins, the first possibility is to read GIODOUTx, modify the content (AND, OR, etc.), and write the result into GIODOUTx. This solution will take, in the best case, eight Cycles (LDR,AND/OR, STR), and could be interrupted by a function modifying the same register, which will result in a data coherency problem.

Using the GIOSETx or GIOCLRx registers the operation will take only three cycles (STR) and is not interruptible.

TDMI - Basic Debug Theory

- JTAG to control debug state
- Load/store registers/memory
- ICEBreaker:
  - Trigger CPU entry into debug state
  - Contains two watch point registers
  - Full address/data and status compare with mask
  - Hardware breakpoints / Single step
  - S/W breakpoints in RAM
  - Debug communications channel



The TMS470R1x-ICEBreaker module, referred to simply as ICEBreaker, provides integrated on-chip debug support for the TMS470R1x core. ICEBreaker is programmed in a serial fashion using the TMS470R1x TAP controller. It consists of two real-time watch point units, together with a control and status register. One or both of the watch point units can be programmed to halt the execution of instructions by the TMS470R1x core. Execution is halted when a match occurs between the values programmed into ICEBreaker and the values currently appearing on the address bus, data bus and various control signals.

The set of registers and comparators used to generate debug exceptions (e.g., breakpoints) Forces TMS470R1x into debug state after a breakpoint, watch point or debug-request has occurred. After an instruction has been breakpointed, the core does not enter debug state immediately. Instructions are marked as being breakpointed as they enter TMS470R1x's instruction pipeline. Thus TMS470R1x only enters debug state when (and if) the instruction reaches the pipeline’s execute stage. A breakpointed instruction may not cause TMS470R1x to enter debug state for one of two reasons:

A branch precedes the breakpointed instruction. When the branch is executed, the instruction pipeline is flushed and the breakpoint is cancelled.

An exception has occurred. Again, the instruction pipeline is flushed and the breakpoint is cancelled.

However, the normal way to exit from an exception is to branch back to the instruction that would have executed next. This involves refilling the pipeline, and so the breakpoint can be re-flagged. When a breakpointed conditional instruction reaches the execute stage of the pipeline, the breakpoint is always taken and TMS470R1x enters debug state, regardless of whether the condition was met. Breakpointed instructions do not get executed: instead, TMS470R1x enters debug state. Thus, when the internal state is examined, the state before the breakpointed instruction is seen. Once examination is complete, the breakpoint should be removed and program execution restarted from the previously breakpointed instruction.

The TMS470 Is Easy To Use

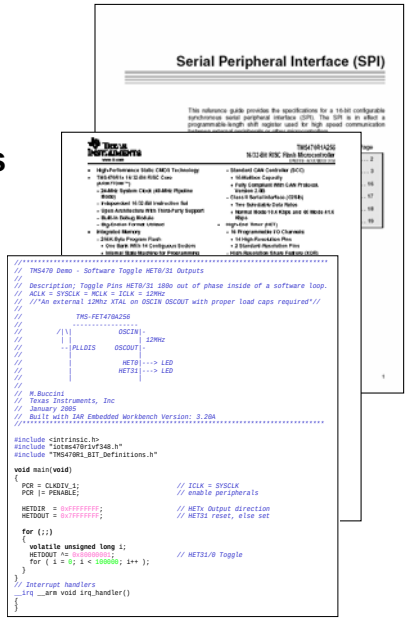
```
void main(void)
{
  PCR = CLKDIV_1;                 // ICLK = SYSCLK
  PCR |= PENABLE;                 // Enable peripherals
  HETDIR = 0xFFFFFFFF;            // HETx Output direction
  HETDOUT = 0x7FFFFFFF;           // HET31 reset, else set

  for (;;)
  {
    volatile unsigned long i;
    HETDOUT ^= 0x80000001;        // HET31/0 Toggle
    for ( i = 0; i < 100000; i++ );
  }
}
```


The TMS470 ARM7TDMI is not only powerful, it is easy to program as well. The example shown toggles two HET output pins in software. The HET pins are configured as outputs and software provides the time interval for flashing the LEDs.

The IAR tools allow use of C, C++ and ARM Ltd. assembly code instructions. We provide Sample programs on the www.ti.com/tms470. We also give the programmers project templates that reduce workspace setup time and give predefined bit definitions that create easier to understand code.

www.ti.com/tms470

- **Datasheets**
- **Peripheral Reference Guides**
- **Code examples**
- **Errata**



The 470 has many resources including a dedicated TI website www.ti.com/msp470.

The most current TMS470 documentation is available at the website.

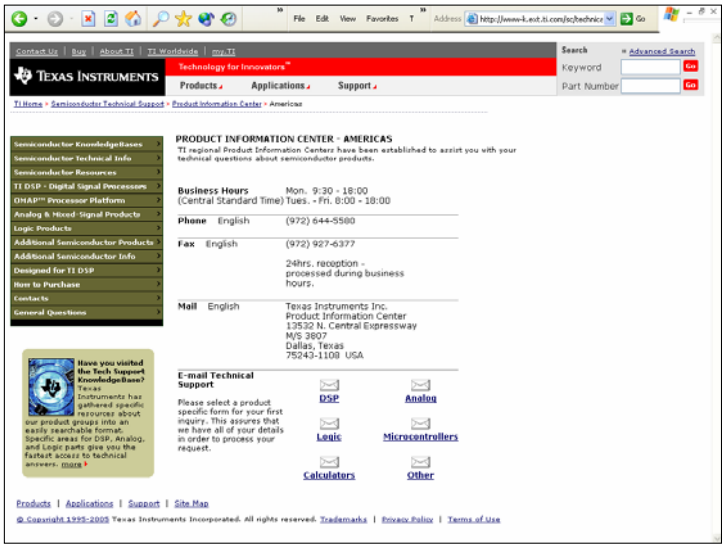
Peripheral reference guides provide detailed technical information on device peripherals.

Chip specific electrical, package, and memory configuration information is available in device-specific datasheets.

Many downloadable code examples are available also at the website.

Any known silicon errata is available.

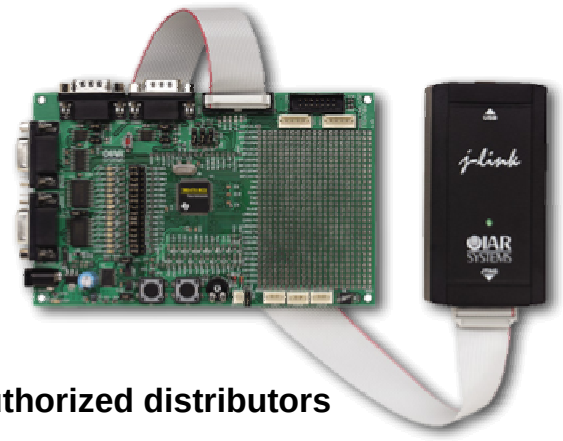
Product Information Center



TI regional Product Information Centers have been established to assist you with your technical questions about semiconductor products.

Get Started Now

- **TMS-FET470A256 Kickstart Development Kit**
  - Embedded Workbench
  - Assembler/linker
  - 16KB IAR compiler
  - USB JTAG debugger
  - TMS470R1A256
  - RS232 interfaces
  - CAN interface
  - 2 button inputs
  - ADC potentiometer
  - 16 LEDS
  - Voltage regulator



- **Available from TI authorized distributors**
- **$399 USD**

The TMS-FET470R1A256 Flash Emulation Tool – or FET – is a complete JTAG based real-time Integrated Development Environment (IDE).

The FET comes with an application target board with a device soldered on USB JTAG debugger, USB cables and all documentation on a CD-ROM.

A CD-ROM includes an IAR Kickstart embedded workbench IDE with 16kB Compiler.

The FET supports complete in-system development. Programming, assembler/C-source level debug, single stepping, multiple hardware breakpoints, full-speed operation and peripheral access are all fully supported in-system using JTAG.

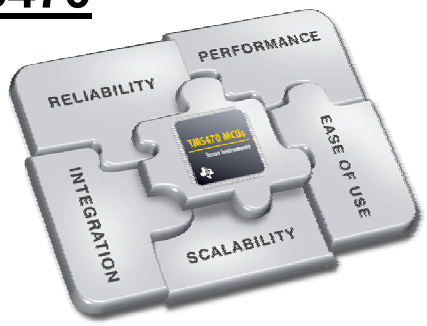
The FET comes complete with everything required to complete an entire project!

Customers wishing to purchase an unlimited C-compiler can do so from IAR; www.iar.com.

The FET is available through authorized TI distributors or on line at www.ti.com/tms470.

TI Delivers With TMS470

- **Complete ARM7 Solution**
  - Breadth of silicon offering
  - Enhanced peripheral library
  - Easy-to-use tools
- **Expert in ARM Technology**
  - Quality and reliability
  - Long-term supplier
  - Support
- **Flexibility**
  - Industry Standard Architecture
  - Roadmap
  - Scalable
- **Proven in Critical Applications**



Complete Solutions

TI is the only ARM7 MCU supplier with devices ranging all the way from 64K to 1M Flash. The TMS470 enhanced peripheral library empowers the ARM7 CPU by offloading many performance intensive tasks thus providing TMS470 with greater system thru-put. The TMS470 has easy to use, industry standard tools available now.

Expert in ARM Technology

The TMS470 foundation in automotive applications brings with it the highest standards of quality and reliability. TI's commitment to current TMS470 customers require TI to be a very long term supplier TI's has an excellent track record of being a long term supplier of DSPs and many other product families. TI's understanding of ARM systems design, with many years experience in Wireless, Imaging, and Industrial MCU system design allows TI to provide excellent support.

Flexibility

ARM Ltd.'s Industry Standard Architecture and roadmap give the TMS470 platform inherent flexibility.

TI's vast ARM experience provides a wealth of ARM knowledge, including core, peripherals, tools, and applications to draw from.

The TMS470 platform of 7 devices scheduled for release in 2005 already provides the greatest flexibility with regard to on chip Flash. Additional enhancements to TMS470 platform will also be coming…

Proven in Critical Applications

The TMS470 is an already proven solution in automotive applications that have shipped for many years. The TMS470 is shipping in high volume, safety-critical, and high-reliability automotive applications today. If you look under the hood… TI is silently there.