MICROCONTROLLER SOLUTIONS FOR CAN NETWORKING

FOR COST-EFFECTIVE & HIGH-END INDUSTRIAL AND AUTOMOTIVE APPLICATIONS
Several times a day, most of us use embedded CAN applications without knowing it. The most popular applications are in:

- Automotive and Transportation Systems
- Medical and Agricultural Equipment
- Building Automation
- Household and Food Appliances
- Robotic and Factory Automation

This wide-spread success is mainly due to the high reliability offered by the CAN protocol.

To satisfy the growing demand for embedded networks, the Atmel CAN family has been specially designed for all applications needing easy and frequent code updates. The CAN family extends its 8051 offering with a 16 MIPS AVR® RISC microcontroller with up to 128 Kbytes of Flash program memory, providing customers with a large choice for their CAN systems.

**Performance range**

Based on either 8051 or AVR core, the CAN 8-bit Flash microcontrollers achieve 5 MIPS or 16 MIPS processing speed respectively.

**Powerful On-chip CAN Controller**

V2.0AV2.0B compliant – Handles independent message objects programmable on-the-fly.

**Easy Remote Programming and Field Upgrade**

Highly flexible self-programming via CAN, UART, SPI, JTAG

**Support Higher Layer Protocol Stacks**

CANopen, DeviceNet™, J939 and OSEK™

---

**Popular 8051 Architecture**

for Cost-effective Applications

- 5 MIPS at 30 MHz
- Up to 64 KB Flash Program Memory
- Flexible Self Programming at 3V and 5V via UART or CAN
- 6-clocks per cycle provides 1 Mbit/sec., using only 8 MHz crystal, thus reducing EMI

---

**Powerful AVR**

Architecture to Tackle the Most Demanding High-end Applications

- 16 MIPS at 16 MHz
- Self Programming
- Hardware Multiplier
- Optimized for High-level C Language
- On-chip Debug Through JTAG Interface
- Up to 256 KB Addressing Capability
Independent message objects can be dedicated dynamically, either in **2.0A** or in **2.0B mode**, to one of the following:
- Reception Channel
- Transmission Channel
- Receiver Buffer (multiple CAN frames buffer)

Each message object has its own masking and filtering registers, which significantly reduces the host processor load.

Indirect addressing allows easy access to all the Data & Control bytes of the CAN controller.

On the AT90CAN128/64/32, an interrupt accelerator engine informs the CPU of critical messages without running a software scan routine, minimizing the repercussions on real time event applications.

Also supported:
- Autobaud and Listening Modes
- Readable Error Counter
- Error Capture with Interrupt
- Time Triggered Communication & Time Stamp

**Example of**
T89C51CC01, AT89C51CC03  
or  AT90CAN32, AT90CAN64, AT90CAN128

**CAN Buffer Configuration**

**Message object 14**  
**Message object 13**  
**Message object 12**  
**Message object 11**  
**Message object 10**  
**Message object 9**  
**Message object 8**  
**Message object 7**  
**Message object 6**  
**Message object 5**  
**Message object 4**  
**Message object 3**  
**Message object 2**  
**Message object 1**  
**Message object 0**

**Efficient & Flexible CAN Controller**

**Typical CAN Application: A Complete Solution**

**High Level Protocol Partners**

Development Tools & Support

**Starter Kits**
- Atmel

**Reference Design**
- ESAcademy® (CANopen)

**Compilers**
- Crossware®, Hi-Tech, IAR™, Keil™, Raisonance, SDCC, Tasking®

**Emulators**
- Ceibo, Hitex, Metalink®, Nohau, Signum Phyton™

**In-system Programming**
- Atmel: PC-based Pilot FLIP,
  ESAcademy CANopen compliant

**Flash device programmers**
- Advantech®, BP Microsystems, Data I/O®, Dataman, Hi-Lo Systems®, ICE Technology™, Elenc®, etc.

---

**T O O L S F O R 8 0 5 1  M I C R O C O N T R O L L E R S**

**Development Tools & Support**

**Development Kit**
- Atmel ATDV90CAN1

**Starter Kits**
- Atmel STK500 + STK501
  + CAN transceiver with ATADAPCAN01

**Compilers**
- CodeVision, GCC-AVR, IAR™, ImageCraft

**Emulators Platforms**
- Atmel ICE 50, JTAGICE JTAGICE mkII

**In-system Programming**
- Atmel AVRISP, JTAGICE JTAGICE mkII
  STK500 + STK501

---

**T O O L S F O R A V R  M I C R O C O N T R O L L E R S**

**Integrated Development Environment for AVR**

(Includes Atmel Macro Assembler)

- Front End for Atmel Starter Kits, Programmers, and Emulators
- C and Assembly Source Level Debugging
- Supports Third Party Compilers
- Maintains Project Information
- Freely Available from http://www.atmel.com
<table>
<thead>
<tr>
<th></th>
<th>T89C51CC02</th>
<th>T89C51CC01</th>
<th>AT89C51CC03</th>
<th>AT90CAN32</th>
<th>AT90CAN64</th>
<th>AT90CAN128</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Speed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIPS</td>
<td>up to 5</td>
<td>up to 5</td>
<td>up to 5</td>
<td>up to 16</td>
<td>up to 16</td>
<td>up to 16</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flash Program/Boot</td>
<td>16 KB/2 KB</td>
<td>32 KB/2 KB</td>
<td>32 KB/64 KB</td>
<td>64 KB/16 KB</td>
<td>64 KB/16 KB</td>
<td>128 KB/16 KB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>2 KB</td>
<td>2 KB</td>
<td>2 KB</td>
<td>1 KB</td>
<td>2 KB</td>
<td>4 KB</td>
</tr>
<tr>
<td>RAM</td>
<td>0.5 KB</td>
<td>1.2 KB</td>
<td>2.2 KB</td>
<td>2 KB</td>
<td>4 KB</td>
<td>4 KB</td>
</tr>
<tr>
<td><strong>CAN Controller</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prog. Mess. Objects</td>
<td>4</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>CAN</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes(^{(2)})</td>
<td>Yes(^{(2)})</td>
<td>Yes(^{(2)})</td>
</tr>
<tr>
<td>UART</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes(^{(2)})</td>
<td>Yes(^{(2)})</td>
<td>Yes(^{(2)})</td>
</tr>
<tr>
<td>SPI</td>
<td>–</td>
<td>–</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>JTAG</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Self-programming Capability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Fail Detect</td>
<td>–</td>
<td>–</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>On-chip Peripherals</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16-bit Timers</td>
<td>0/1/2</td>
<td>0/1/2</td>
<td>0/1/2/3</td>
<td>0/1/2/3</td>
<td>0/1/2/3</td>
<td>0/1/2/3</td>
</tr>
<tr>
<td>PCA Channels</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Pulse Width Modulation</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>10-bit ADC</td>
<td>8 channels</td>
<td>8 channels</td>
<td>8 channels</td>
<td>8 channels</td>
<td>8 channels</td>
<td>8 channels</td>
</tr>
<tr>
<td>21-bit Watchdog</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>UART</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SPI</td>
<td>–</td>
<td>–</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Two-wire Interface</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Available Versions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ports</td>
<td>0/1/2/3</td>
<td>0/1/2/3</td>
<td>0/1/2/3</td>
<td>A/B/C/D/E/F/G</td>
<td>A/B/C/D/E/F/G</td>
<td>A/B/C/D/E/F/G</td>
</tr>
<tr>
<td>Max. Frequency</td>
<td>60 MHz</td>
<td>60 MHz</td>
<td>60 MHz</td>
<td>16 MHz</td>
<td>16 MHz</td>
<td>16 MHz</td>
</tr>
<tr>
<td>Power Supply (V)</td>
<td>3 - 5.5</td>
<td>3 - 5.5</td>
<td>3 - 5.5</td>
<td>2.7 - 5.5</td>
<td>2.7 - 5.5</td>
<td>2.7 - 5.5</td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>-40 to +125</td>
<td>-40 to +125</td>
<td>-40 to +125</td>
<td>-40 to +125</td>
<td>-40 to +125</td>
<td>-40 to +125</td>
</tr>
<tr>
<td>Automotive Graded Parts</td>
<td>SOIC24</td>
<td>TQFP44</td>
<td>TQFP44</td>
<td>TQFP64</td>
<td>TQFP64a</td>
<td>TQFP64</td>
</tr>
<tr>
<td>Packages</td>
<td>SOIC28</td>
<td>PLCC44</td>
<td>PLCC44</td>
<td>QFN64</td>
<td>QFN64</td>
<td>QFN64</td>
</tr>
<tr>
<td></td>
<td>PLCC28</td>
<td>BGA64</td>
<td>BGA64</td>
<td>BGA64(^{(2)})</td>
<td>BGA64(^{(2)})</td>
<td>BGA64(^{(2)})</td>
</tr>
<tr>
<td></td>
<td>TQFP32</td>
<td></td>
<td></td>
<td>TQFP64</td>
<td>TQFP64</td>
<td>TQFP64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PLCC52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td>Now</td>
<td>Now</td>
<td>Now</td>
<td>1Q/06</td>
<td>4Q/05</td>
<td>Now</td>
</tr>
</tbody>
</table>

\(^{(1)}\) ROM version available  
\(^{(2)}\) Using custom bootloader  
\(^{(3)}\) Check for availability  

Notes: Software Master/Slave (M/S) LIN drivers available. All products compatible with AT6660 Transceiver.