



---

## Technological Arts Inc.

*Technological Arts*

# Adapt9S12DG128SM0 with Serial Monitor, min. config.

USD \$125.00



### Product Info

Adapt9S12D is a compact, modular implementation of Freescale's flagship 9S12D family of microcontroller chips. The HCS12 family is backward-compatible with 68HC12 (and 68HC11), and utilizes the latest process technology. By shrinking feature size, more functionality can be put on a chip, reducing power consumption and cost, while increasing operating speed. Adapt9S12D is an ideal low-cost platform that brings these advantages within easy reach of engineers, students, and hobbyists. The flexible design of the entire "Adapt" series microcontroller products accommodates all facets of training, evaluation, development, prototyping, and even volume production.

Implemented in an "AdaptS12" form factor board, Adapt9S12D is compatible with an entire range of application cards, prototyping cards, and backplanes, and usable with solderless breadboards ("proto boards"). The Minimal configuration includes all necessary support circuitry for the MCU, as well as a 5-Volt regulator and RS232 transceiver on-board.

### Product Details

**MCU Overview:** The MC9S12DG128 microcontroller unit (MCU) is a 16-bit device composed of standard

on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), 128K bytes of Flash, 8K bytes of RAM, 2K bytes of EEPROM, 2 asynchronous serial communications interfaces (SCI), three serial peripheral interfaces (SPI), an 8 channel IC/OC enhanced capture timer, two 8-channel, 10-bit analog-to-digital converters (ADC), an 8-channel pulse-width modulator (PWM), 89 discrete digital I/O channels (Port A, Port B, Port K and Port E), 20 discrete digital I/O lines with interrupt and wakeup capability, five CAN 2.0 A, B software compatible modules (MSCAN12), and an Inter-IC Bus.

### Module Features:

- surface-mounted 9S12DG128 MCU
- 2.30" x 3.25" AdaptS12 form-factor
- two 50-pin connectors bring out all I/O pins of the MCU
- all I/O pins on a 0.1" grid for easy interfacing to standard perfboard
- versatile connector design for use with solderless breadboards, prototyping cards, or embedding into your design
- RS232 transceivers provided for both SCI channels
- 16 MHz crystal, but internal bus can run up to 24MHz (using on-chip PLL)
- accommodates optional user-supplied oscillator
- Background Debug Mode (BDM) fully supported for loading and debugging user code
- universal 6-pin/10-pin BDM connectors support BDM pods from multiple vendors
- program in C, BASIC, Forth, assembler, etc.
- no special Flash programming voltage or switch required
- fast in-circuit programming
- **on-chip Serial Monitor** for quick loading of user programs via CodeWarrior or uBug12
- 500mA, automotive grade, low-dropout 5V regulator on-board
- includes low-voltage inhibit reset circuit + reset button
- accommodates optional precision voltage reference chip for analog-to-digital converters
- **has LOAD/RUN switch for Serial Monitor**
- **CAN and RS485 transceivers not populated**

### What's included:

- assembled, tested module, pre-programmed with the Freescale Serial Monitor (AN2548)
- red and black power connector wire (#PCJ1-8)
- printed schematic/pinout sheet
- data sheets, manuals, and all other resources for this product can be found by clicking on the Resources tab above

If you require mating connectors, browse Connectors in the Components category at the left. For power supplies, browse the Accessories category.

### **Before you order...**

**All of our [standard connector options](#)** are available for this board. The product photo above shows "RA1" connector on both H1 and H2. Please make your choice of connectors before adding this item to your shopping cart. Otherwise, the board will be shipped with the default option (no connectors on H1 and H2).

### **- Flash-based code development using on-chip Serial Monitor**

The Serial Monitor is supported by CodeWarrior, NoICE12, uBug12, and others.

## - Flash-based code development using a BDM pod

Use a BDM pod (such as our USBDMILT) to erase and load Flash with your program. The entire 128K Flash can be used, because the Flash-resident bootloader is no longer needed (therefore pseudo-vectors aren't required either).

Â

[Vendor Information](#)