

Replace Fixed-Function ICs with Low-Cost Microcontrollers

For about 25 cents, you can implement many simple analog and digital functions into applications ranging from pulse-width modulation to a stopwatch.

Most engineers have known for years that microcontrollers are the best choice in building new digital products. Legacy TTL and CMOS functional logic devices gave way to single-chip MCUs years ago. Yet some operations are still implemented with fixed-function devices because an MCU seems like expensive overkill in certain cases. Not anymore, though. When the price for a tiny MCU comes down to pennies, it's time to reconsider the use of fixed-function circuits. Here are some ideas to help you go in that direction.

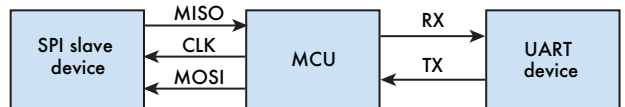
DESIGN EXAMPLES

A cheap MCU can replace more discrete logic and mixed-signal devices than you think. One example is a two-bit 8-bit microcontroller that's able to replace the still popular 555 timer IC. But that's not all. There are four basic areas where an inexpensive MCU can replace a fixed-function device: communications, system housekeeping, pulse-width modulation (PWM), and timing. The examples below illustrate how it can be done.

COMMUNICATIONS

In terms of communications, one example involves the need for two devices to talk to one another. One device uses a UART for external serial communications and another employs a serial peripheral interface (SPI). What's required is a bridge that acts as a translator for the two devices (Fig. 1). The MCU code translates the protocol of one serial method to another and provides data buffering and half-duplex data transfers. The ability to program the device lets you change the design to optimize it for your application.

Another communication application example concerns the use of a single bidirectional line to transfer data between the MCU and a master device. Commonly referred to as 1-wire

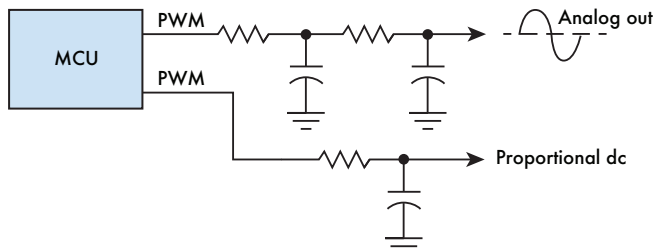


1. Here, the MCU acts as a UART-to-SPI bridge.

or SDQ single-wire serial interfaces, this communication peripheral reduces the number of physical hardware connections required while adhering to a protocol that's easily achievable with a simple MCU. This simple interface is used with external EEPROMs, SHA-1 authenticators, temperature sensors, digital switches, and battery-system monitors.

PULSE-WIDTH MODULATION

Most low-cost MCUs include PWM capability. PWM generates a variable pulse duration and duty cycle. These output pulses can be easily averaged into a proportional dc with external low-pass RC filters. The PWM function can also serve as a simple digital-to-analog converter (DAC). Among the many applications are LED dimming or color change, servo- or stepper-motor control, signal generation from a lookup table, or other analog-related functions. Figure 2 shows the necessary external RC filters for most uses.



2. External low-pass filters create analog and DC outputs from PWM signals.

HOUSEKEEPING

Many possibilities exist for system and housekeeping functions. Here are just a few examples:

- EEPROM emulation using the MCU's non-volatile memory
- Low-power hex keypad
- Multifunction reset controller
- Single-slope analog-to-digital conversion
- ADC Wake and Transmit on Threshold
- Hysteresis comparator
- Programmable frequency-locked loop
- Programmable clock source

Figure 3 shows how the non-volatile ferroelectric RAM in an inexpensive MCU can replace EEPROM that's often needed in certain applications. The EEPROM is used to store boot information, identification codes, backup data, calibration data, and other data that must be retained when power is off. The inexpensive FRAM MCU is connected via a SPI interface with traditional control signals.

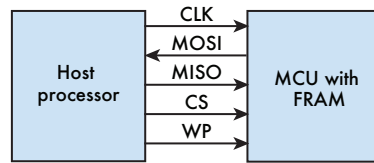
TIMING

Timing functions offer many other possibilities. Examples are simple kitchen timer, a stopwatch, a watchdog timer, or real-time clock. An internal 32-kHz reference oscillator, if available, can provide the timer accuracy, and an external crystal clock can be used for high-precision requirements. Internal timer counters accumulate the clock pulses.

Figure 4 shows a stopwatch example. Stop-start and reset buttons provide the external control. A 7-segment LED display shows the output. General-purpose digital I/O pins on the MCU accept the inputs and generate the outputs. External transistors are needed to control the multiplexed LED display.

What's discussed above represents only a few instances of how a cheap MCU can solve some interface problems. You know the value of programmability, and now you can apply that to simple functions when the cost is right.

Texas Instruments has produced an eBook that illustrates the many application possibilities of low-cost MCUs. The



3. An MCU with FRAM does a good job of emulating needed external EEPROM.

eBook includes brief application reports and links to code examples to get you started, as well as tips for code optimization.

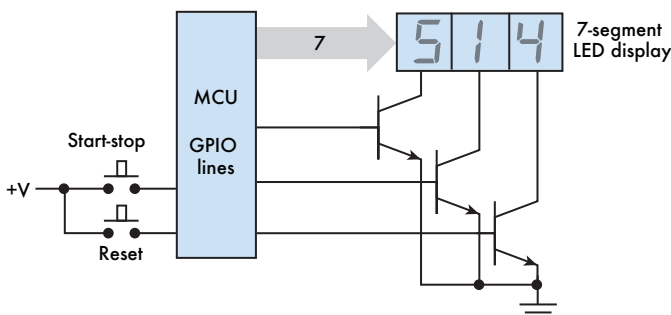
SELECTING THE RIGHT MCU

Literally dozens of inexpensive MCUs are available to implement external functions like those described here. One family of devices that not only meets the low-cost (\$0.25) target but also provides significant software support to make these MCUs easy to use is the ultra-low-power MSP430 microcontrollers (MCUs) from Texas Instruments.

These 16-bit MCUs feature a RISC architecture, on-chip RAM, and TI's ferroelectric non-volatile RAM. Clock rate can be up to 16 MHz, and I/O consists of GPIO pins, I²C, SPI, and UART interfaces. PWM capability and multiple timers add more versatility. All versions have a comparator with programmable hysteresis and a 10-bit ADC. The design is optimized for very low power operation. Multiple versions are available with 16- or 24-pin package options. A variety of evaluation boards and software development products provide design support.

MORE INFORMATION WAITING FOR YOU

Texas Instruments has plenty of additional data on the use of low cost MCUs. Check out their "25 functions for 25 cents" video series to see some examples of the simple analog and digital functions that can be replaced by low-cost MSP430 MCUs. Also, check out this blog that provides additional examples.



4. An inexpensive MCU makes a simple and inexpensive stopwatch.