App Note 3521: Configuring the MAXQ3120 Mixed-Signal Microcontroller for Practical Applications

With a broad range of peripherals and an 8MIPS core, the MAXQ3120 is a perfect fit for a wide range of mixed-signal applications. In this article, four applications are presented that put the features and functions of the MAXQ3120 to use.

Designers and manufacturers of monitoring applications have repeatedly demanded a microcontroller versatile enough for typical daily applications. The activities could include electricity metering, automotive systems, data gathering, and sensor conditioning. The MAXQ3120 low-power, high-speed microcontroller was designed especially for versatile implementation. Here are the principal specifications garnering attention for this device.

- 16-bit, 8 million instructions per second (MIPS), one-cycle RISC core
- 32kB flash memory
- 512B RAM
- Two UARTs with independent baud-rate generators
- Three timers, one capable of PWM D/A
- Infrared communication capability
- LCD controller capable of driving 112 segments
- Battery-backed real-time clock with time-of-day and subsecond alarms
- 16 x 16-bit one-cycle multiplier with a 40-bit accumulator
- Two precision, 16-bit analog-to-digital converters (ADCs)

With all these impressive features, let us discuss what the MAXQ3120 can accomplish with analog I/O and DSP functions.

Voice Recorder Subsystem

The concept: Give a group of engineers an ADC, and someone will discover how to record voice with it. But the MAXQ3120 can do much more than simply record sound. Coupled with user-interface components and an inexpensive NAND flash memory, the MAX3120 becomes the core of a full-featured voice recorder subsystem.

The details: To perform audio I/O, one of the two ADCs and the PWM timer channel are used. The ADCs have a nominal input of +1V to -1V, and the built-in preamplifier has a programmable gain of up to 16X. In many cases, a condenser microphone cartridge with a built-in impedancematching amplifier can connect directly to the ADC input. If lower noise or more gain is required, Maxim makes a microphone preamplifier (the MAX4467) that provides the required bias to the microphone and an extremely low-power shutdown mode for battery-operated applications. On the output side, a single-stage amplifier drives the speaker and performs the modest antialiasing and PWM smoothing required.

Once the audio has been converted to a digital sample, it must be compressed and stored for subsequent playback. At 8MIPS, the MAXQ3120 has sufficient processor horsepower for many of the standard voice codecs in common use. The "gold standard" is the ITU G.711 codec, which operates at 64kbps, transmitting and receiving 8,000 8-bit samples per second. The G.711 codec can operate with two different transfer functions that map a 12-bit sample value to an 8-bit codeword; these functions are commonly known as A-law (used primarily in Europe) and µ-law (used primarily in the United States).

If higher compression is desired, the ITU G.726 codec is a possibility, at some sacrifice of voice quality. The G.726 codec uses adaptive delta pulse-code modulation (ADPCM) to more efficiently encode speech without storing the full value of each sample. This codec supports several bit rates down to 16kbps and, for most implementations, requires less than 3MIPS to operate. Both the G.711 and the G.726 codecs require very little RAM.

In the record phase, a timer goes off every 125µs (every 1,000 processor cycles at 8MHz), and the processor averages the samples received during the previous timer period (It is either two or three samples, as the ADC provides a new sample every 48µs.) to match the desired sampling rate of 8kHz. The 16-bit accumulated samples can then be encoded using the selected codec. For playback, the sample data is linearized and then passed to the PWM controller for presentation to the speaker.

Following compression, the audio samples are ready for storage. The MAXQ3120 has no nonvolatile storage other than its program flash, so some form of external storage must be found. The most cost-effective storage solution for this application is NAND flash, which comes in densities up to 8 gigabits. With a 16kbps codec, such a device would provide more than six days of voice storage! NAND flash is not perfect, however. First, most NAND flash devices come with a defect map that tells the application where "dead bits" reside within the memory array. Second, NAND flash cells, like those of all electrically erasable memory devices, tend to lose the ability to change state after extended use. Fortunately, in a voice application both of these types of errors cause fewer problems than they would if the memory were used, say, as a solid-state disk. Consequently, with NAND flash memory, defects of this type may not even be noticed, and if they are, only as a momentary glitch in the audio.

With such a vast amount of voice storage, some way of managing the recordings is necessary. This is the job of the user interface, the heart of which is the LCD controller. Capable of driving 28 segment lines across four common planes, the MAXQ3120's LCD controller is compatible with a large number of existing 3V LCD glass modules. Plus, custom LCD modules can be designed and manufactured very cost effectively.

The user controls the recorder through pushbuttons connected to general-purpose I/O ports. There are four 8-bit ports available that are shared with other functions on the device.

What is left to do? The MAXQ3120 is the ideal microcontroller to use in advanced voice-recording systems. There are only a few engineering tasks left to the designer of such a system.

- *Design the user interface*: Select an LCD and decide how information is displayed, which buttons perform what function, and how recordings are organized.
- *Select a voice codec*: You can use one of the ITU codecs mentioned, select another proprietary codec, or store the samples raw if the memory is large enough. Many standard codecs can be purchased as C source code, with only small, low-level interface routines left to be coded.
- *Select a storage medium*: NAND flash is considered the best option, but there are other storage mechanisms that may be more attractive in your application. Removable storage (such as SD, SmartMedia® , or MMC Memory Cards), for example, is inexpensive and ubiquitous, and some manufacturers provide source code in C and development kits to help you design the storage card interfaces.
- *Battery management*: If the recorder is to be used in a battery-operated environment, some form of power management is required. Maxim has many low-cost battery-management solutions. Coupling those solutions with the very low-power stop and sleep modes available in the MAXQ3120 give the voice recorder an appealing battery life.

Doppler Radar Alternative

The concept: Law enforcement has used speed radar for longer than many of us have been driving. If the cost was low enough, this technology could be extended beyond ridding the streets of speeders. For example, Doppler radar could automatically alert the driver if it detected a stopped car ahead. How can the MAXQ3120 help here?

The details: The concept behind Doppler radar is simple. The radar set emits a continuous microwave beam at some known frequency. (In the United States, for example, it is increasingly the Ku band, or about 24.150GHz.) The beam then bounces off a moving object and is reflected back to the radar set, at a frequency slightly higher or lower than the transmitted frequency. The reflected signal is then mixed with the transmitted signal to produce a "beat note" with a frequency defined by:

$$f = [v * (f_0 / c)] * \cos\theta$$

where v is the velocity of the object measured, f_o is the nominal transmitter frequency, θ is the angle between the object's direction of travel and the radar set (see **Figure 1A**), and c is the speed of light. Notice that if the object is directly approaching the radar set, then θ is zero and cos θ is one. The velocity of the target then becomes:

$v = [f / (f_0 / c)]$

If, for example, a Ku band radar set produces a 1kHz signal, the target is approaching (or receding) at 12.4 meters per second, or about 28 miles per hour (or 45 kilometers per hour). Processing this audio frequency signal is the MAXQ3120's task. See **Figure 1B** for a block diagram.

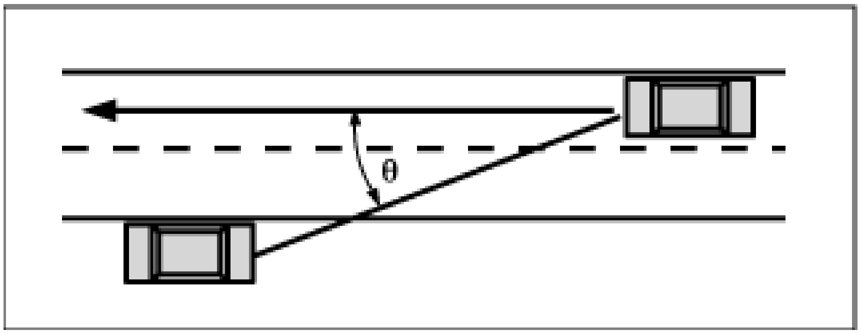

Figure 1A. The Doppler shift of the received signal is dependent on the target's speed and the angle between the direction of the target's travel and the radar set.

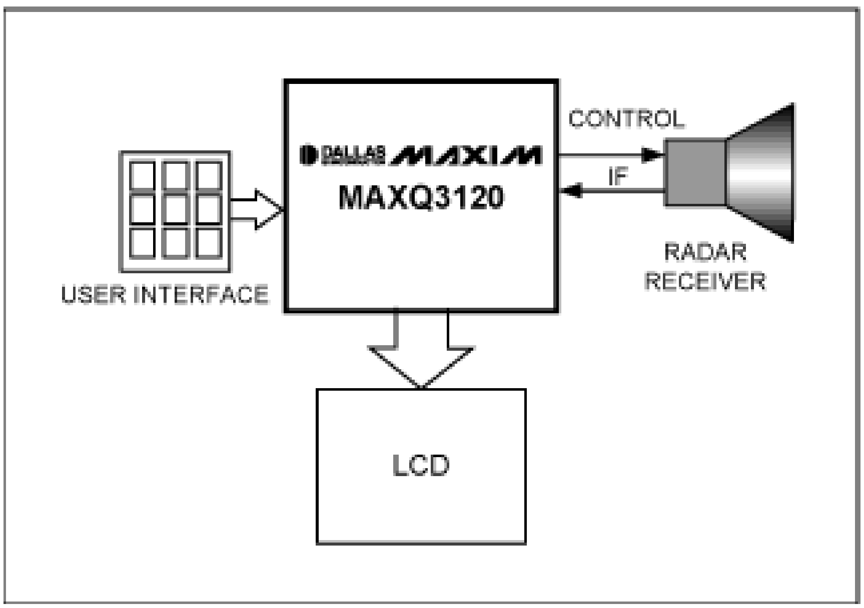

Figure 1B. In the Doppler radar application, the MAXQ3120 controls a radar head and recovers the "beat note" embedded in the intermediate frequency (IF) signal returned from the receiver.

Using one of the two ADC channels, the MAXQ3120 can sample the difference signal, extract the strongest frequency component, and scale that to a kilometers-per-hour (or miles-per-hour) reading. Additionally, using the multiply-accumulate unit, sophisticated filtering can isolate the strongest signal and possibly derive intelligence (such as the speed of the operator's own vehicle) from lower-level signals.

In some cases, the user interface is trivial—it may be nothing more than a logic level or contact closure to activate an audible alert. In other applications, the microcontroller may periodically record velocity and the time and date at which the measurement was made.

What is left to do? There are several manufacturers of Doppler radar modules, and most produce an IF in the audio band. For a simple velocity-measuring set, there is little else to do. For a set that performs more complex analysis, some engineering work is required to develop the algorithms for processing the signal. Fortunately, there are excellent tools that aid in the development of complex filtering and discrimination algorithms.

In some applications, it is desirable to know the direction of travel, i.e., is the target approaching or receding? A conventional Doppler radar cannot tell; the frequency shift is the same without regard to the direction of travel. But some manufacturers make a radar module that includes two quadrature outputs. By demodulating both and determining the phase difference between the two channels, the set can determine the direction of travel. As the MAXQ3120 has two ADCs, extending the application for this criterion becomes easy.

Telephone Nanny

The concept: You want to track telephone use—who is calling whom, and the time and duration of each call—but you do not want the expense or complexity of a full-featured call-accounting system. For example, parents want to track their children's telephone use. Professionals want to keep an automatic log of whom they call, who calls them, and when. Those who rent rooms by the night in a "bed-and-breakfast" would like to offer telephone use, but find it difficult to track guests' calls when a room telephone is an extension of the house line. Another MAXQ3120 project holds a solution: a device that monitors all the telephones on a phone line and tracks incoming and outgoing calls (see **Figure 2**).

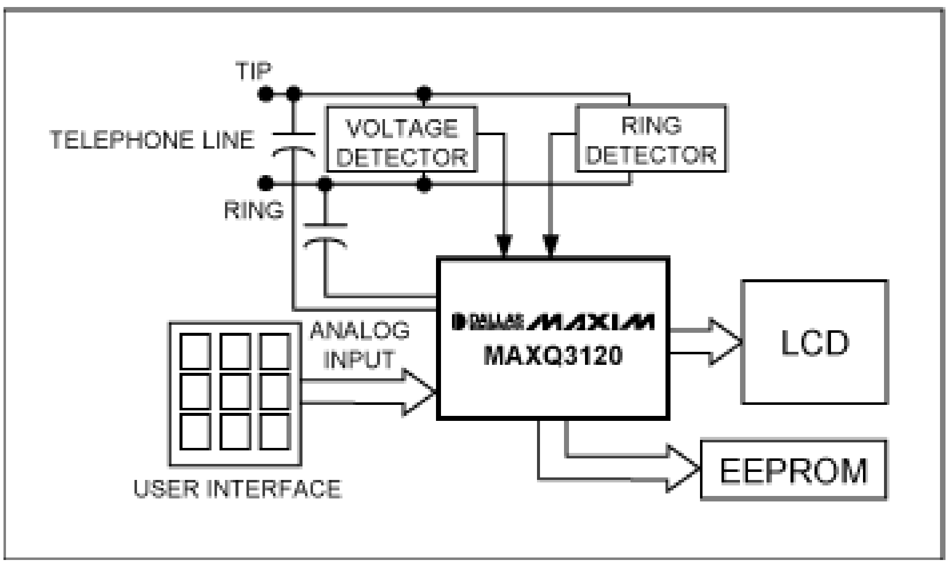

Figure 2. The telephone nanny monitors dialed digits, incoming ring, caller ID, and off-hook events to determine what numbers are called, what numbers are calling, and how long conversations last.

The details: The telephone nanny must monitor four types of input: the on-hook/off-hook state of the telephone line, incoming ring signals, dialed numbers, and caller ID signals. The easiest of these tasks is the off-hook signal. Whenever a phone goes off-hook, a voltage detector alerts the MAXQ3120 that the line voltage has changed from about 48V (on-hook) to less than 12V (off-hook).

Incoming ring is caused by a high-voltage AC signal placed on the telephone line by the phone company. To detect this signal, a capacitive-coupled opto-isolator can alert the processor while keeping it isolated from the telephone line. In general, a bidirectional optocoupler in series with 0.47µF and 4.7kΩ will reliably detect an incoming ring. To eliminate false alerts, a pair of back-to-back zener diodes can keep current from flowing in the opto-isolator until the voltage exceeds the breakdown potential.

Receiving dialed numbers is somewhat more complex, as there are two methods used to send digits to the phone company: pulse dialing and tone dialing. In pulse dialing, current is interrupted at 10 pulses per second in a pattern corresponding to each dialed digit. To detect pulse dialing, it is only necessary to time the low-to-high voltage transitions on the line; a series of transitions that occur 10 times per second is undoubtedly pulse dialing.

But fewer and fewer pulse-dial sets are used today. The currently accepted method for sending digits to the phone company is through dual-tone multifrequency (DTMF) signaling. In this scheme, the digits on a telephone are arranged in three columns and four rows, and pressing a key produces one tone corresponding to the row and one tone corresponding to the column on which the key appears. See **Figure 3** for details concerning DTMF frequency assignments. By detecting and decoding the tone combinations, the MAXQ3120 can determine what digits were dialed.

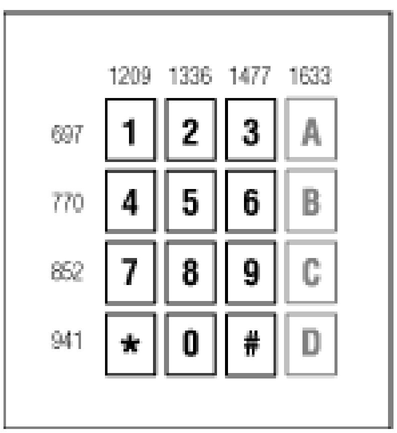

Figure 3. Dual-tone multifrequency (DTMF) signaling assigns one row tone and one column tone to each key on a keypad. Note that the right column (with the A, B, C, and D keys) is not used in customer applications, and is only present in central office equipment.

A simple, CPU-efficient mechanism for detecting tones is known as the Goertzel algorithm. This method is implemented as a two-pole filter and provides a clear, unambiguous indication of the tone presence in a potentially noisy channel. MAXQ implementations of the Goertzel algorithm have been coded and tested.

Determining the telephone number of incoming callers is easy with caller-ID service. Subscribers to this service receive a 1200bps signal in the gap between the first and second rings which contains the number that is calling, the name of the caller, and the time and date.

In the United States, caller ID is transmitted from the central office using the Bell 202 modem standard. (European standards call out ITU V.32 mode 2 (1300Hz mark and 2100Hz space. In both cases, data transmission is 1200bps. It is easy to build the application to support both standards, but for the purpose of this article we discuss the U.S standard.) In this standard, a "0" bit is represented by a 2200Hz tone, and a "1" bit by a 1200Hz tone. The MAXQ3120 can easily detect zero crossings, and thus determine the frequency of the incoming signal and thereby discriminate the bits. The data format is simple, asynchronous serial in the N81 (no parity, 8 bits with one stop bit) format.

Once the bits have been decoded, the message format must be interpreted. There are two types of caller ID. The first type of message conveys only the calling number, plus the date and time. It looks like this:

TYPE	LENGTH	MONTH		DAY		HOUR		MINUTE		PHONE NUMBER	CKSUM
04	12	30	39	32	33	30	39	35	34	39 30 33 35 35 35 31 32 31 32	49
—	—	09		23		09		54		9035551212	—

This above message type is transmitted if the customer subscribed to "number only" caller ID. Notice that all the characters are transmitted as ASCII except for the type, the length, and the checksum. If the phone number is not available, the letter O is sent instead. If the phone number has been blocked by the customer (or at the customer's request), the phone number field contains the letter P. The checksum is the two's complement of the modulo-256 sum of all the message's previous bytes.

Customers subscribing to "name and number" caller ID receive information that looks like this:

TYPE	LENGTH	DATA BLOCK 0	DATA BLOCK 1	...	DATA BLOCK N	CKSUM

The TYPE is always 0x80, and the LENGTH is the length of all data blocks. Each data block has the format:

BLOCK TYPE	BLOCK LENGTH	DATA

The BLOCK TYPE indicates what kind of data is being transmitted, and is selected from the following values:

VALUE	TYPE
1	Date and time
2	Phone number

4	Number not present
7	Name
8	Name not present

Once the data for a particular call is accumulated, it can be stored in an I^{2}C* EEPROM. These devices are inexpensive, reliable, and come in a variety of storage capacities. A 16kb EEPROM can store about 100 name-and-number caller-ID entries. An I^{2}C software implementation is available for the MAXQ family of processors.

What is left to do? There are several enhancements that can be considered. While the project presented here monitors all phones on a line, it cannot tell you which phone initiated or answered a call. To do this, a monitoring device would be needed on each phone station, but without the user-interface elements. The MAXQ3120 can use a current sensor to determine when the phone to which it is connected goes off-hook. It then can communicate this fact to the central phone nanny. To perform this communication task, the station microcontroller sends DTMF digits which identify the station that made or answered the call. In an on-hook state, the phone company would not even "see" the digits, and the phone wiring in the house makes a perfect conduit for these signals.

A second enhancement is automatic logging to a computer. The MAXQ3120 includes UART channels that can connect to the serial port of a PC, essentially turning the phone nanny into a complete call-accounting system in a tiny package. If you combine this project with the voice-recorder project discussed above, you have the core of very sophisticated answering machine or a telephone-call recorder.

Electricity Monitoring

The concept: Why is my electric bill so high? This is a common complaint heard by electricity providers. Part of the answer, unrelated to fuel prices, is that we have more and more equipment that stays on all the time.

Do not blame your refrigerator—it actually cycles on and off, switching on only when the interior warms above a preset limit. The real culprits for power usage are all around you. Consider the media equipment with glowing lights that tell you they are off, ready for your remote control to command them on. At one time, an "off" switch meant exactly that, as in this device is no longer operating in any fashion. Today, turning a television off simply puts it in standby mode, with much of the circuitry still actively consuming power. In fact, it is difficult to find modern electronic gear that has an honest, circuit-interrupting on-off switch.

Another covert consumer of power is your personal computer. In the era of always-on Internet, many people leave their PC on to catch downloads, get email, and perform other tasks while they are away. How much does that cost in electricity?

In this final section, the MAXQ3120 returns to its roots, but in the service of the electricity consumer instead of the electricity provider. **Figure 4** shows a block diagram for this application.

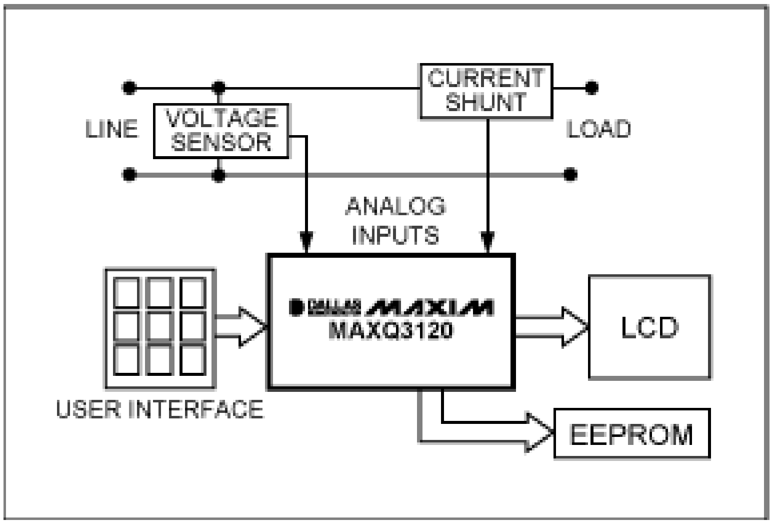

Figure 4. The power monitor can determine how much power is being used by an appliance and when. It also reports brownouts and voltage surges that can damage sensitive equipment.

The details: The MAXQ3120 was built to support an electricity meter application. Its two ADCs are designed to monitor a voltage channel and a current channel. In this project, the MAXQ3120 continuously monitors the voltage presented to and the current flowing through the device to which it is attached. It then reports the average power used by the device, the times and levels of any usage peaks, and, if needed, the power factor that the device presents to the line.

But how does it report? The simplest and most direct way is by a small LCD on the monitoring device. One or more pushbuttons can command the MAXQ3120 to cycle through its display modes (RMS volts, RMS amperes, watts, watt-hours of energy since last reset, etc.). A small LCD is surprisingly inexpensive and makes a compact, usable package when only one device is monitored.

If more than one device is to be monitored, having a central station to record the usage from all stations is preferred. The difficulty here is the poor quality of a typical power circuit used as a data-transmission medium. Transmitting data at any significant rate is beyond the ability of the inexpensive modules that we are trying to build.

But transmitting data at what might be considered an insignificant rate may be cost effective. At just over 20,000 samples per second, the ADCs on the MAXQ3120 cannot demodulate a carrier in the low 100kHz range (the range used by common power-line control systems), but they can demodulate a carrier in the audio range. And if the data is transmitted slowly enough, about 10bps, communication can be made arbitrarily reliable.

So in addition to monitoring power, the MAXQ3120's DSP functions would perform two additional functions: they would look for power in two narrow passbands to attempt to detect very low-speed frequency-shift keying (FSK) modulation; and when requested, they would generate tones in the 3kHz to 7kHz range for FSK detection by the central-monitoring station.

The central-monitoring station can be a stand-alone unit or a device that attaches to a personal computer through a serial port. The latter is attractive, as the PC has virtually unlimited storage and can perform much more complex reporting functions than any microcontroller.

What is left to be done? Not much. A complete power-meter reference design (ZIP, 76KB) is available that can be easily adapted to this project. For power-line communications, both tone generators and bandpass filters have been coded for the MAXQ3120, so building a low-data-rate FSK modem is a matter of plugging together components. In short, this is a project that can be readily assembled from existing hardware and software components into a working product.

Conclusion

As can be seen from its many application uses, the MAXQ3120 microcontroller is much more than the heart of a multifunction electricity meter. This strong, capable microcontroller offers many opportunities to expand on everyday applications, and may be ideal for your next mixed-signal project.

MAXQ is a trademark of Maxim Integrated Products, Inc.
SmartMedia is a registered trademark of Toshiba Corporation.
*Purchase of I^{2}C components from Maxim Integrated Products, Inc., or one of its sublicensed Associated Companies, conveys a license under the Philips I^{2}C Patent Rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.