AN1416

Low-Power Design Guide

Authors: *Brant Ivey*
Microchip Technology Inc.

INTRODUCTION

Low-power applications represent a significant portion of the future market for embedded systems. Every year, more designers are required to make designs portable, wireless and energy efficient. This document seeks to simplify the transition to low-power applications by providing a single location for the foundations of low-power design for embedded systems. The examples discussed in this document will focus on power consumption from the viewpoint of the microcontroller (MCU). As the brain of the application, the MCU typically consumes the most power and has the most control over the system power consumption.

As with all designs, it is important for the designer of a low-power embedded system to consider trade-offs between power consumption, and other factors, such as cost, size and complexity. While some low-power techniques can be used with no cost to the system, others may require trade-offs. This guide will give examples of these trade-offs where applicable. However, it is not feasible to discuss all possible trade-offs, so an embedded designer should keep in mind the possible system level impacts of power-saving techniques.

This design guide will refer to Low-Power modes available on PIC® MCUs, but will not go into detail about these features. For information about the Low-Power modes available on PIC MCU devices, refer to AN1267, *"nanoWatt and nanoWatt XLP™ Technologies: An Introduction to Microchip's Low-Power Devices"* (DS01267).

LOW-POWER BASICS

The definition of low power varies significantly from application to application. In some systems, there is plenty of energy available to run from, but the low-power designer is attempting to minimize operating costs or maximize efficiency. While in other applications, there may be a limited power supply, such as a coin cell battery, which determines the power consumption requirements of the system. These systems require different focuses to minimize power. It is important to consider and understand what causes power consumption and where to focus power minimization efforts to create an effective low-power system.

Main Sources of Power Consumption

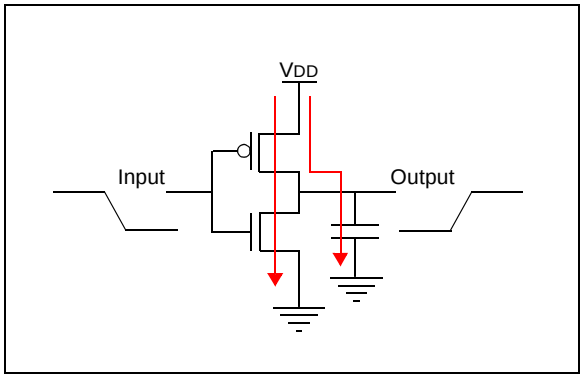
In CMOS devices, such as microcontrollers, the total power consumption can be broken down into two broad categories: dynamic power and static power. Dynamic power is the power consumed when the microcontroller is running and performing its programmed tasks. Static power is the power consumed, when not running code, that occurs simply by applying voltage to a device.

DYNAMIC POWER

Dynamic power consumption is the current which is consumed during the normal operation of an MCU. It includes the power lost in switching CMOS circuits and the bias currents for the active analog circuits of the device, such as A/Ds or oscillators.

To understand where switching losses originate from, consider a CMOS inverter, as shown in Figure 1.

FIGURE 1: CMOS INVERTER DYNAMIC POWER CONSUMPTION PATHS



This inverter will consume little to no power when the input is at VDD or VSS. However, when the signal switches from VDD to VSS, there is a transition period where the PMOS and NMOS will both be biased in the linear region, allowing current to flow from VDD to ground. Also note, in a real system there is some amount of load capacitance on the output bus. There is additional current consumption associated with the charging and discharging of this bus capacitance when the logic level changes.

The average power consumed by dynamic switching losses of a single gate can be defined by the following equation:

EQUATION 1: DYNAMIC POWER CONSUMPTION

$$P = V^2 \times f \times C$$

Where V is the system voltage, f is the switching frequency and C is the load capacitance.

Note that this equation is for a single CMOS device, not the entire MCU. When considering the entire MCU, this equation will be multiplied by a scaling factor (α), which varies depending on the switching frequency of all of the gates in the device.

Equation 1 reveals a few important points to consider about how to control dynamic power consumption. The first point to consider is that voltage is the most significant factor in dynamic power consumption because the voltage term is squared. Reducing the system operating voltage will have a significant impact on power consumption. Another major consideration is which of these components can be modified in a system. Every embedded system has different requirements which will limit the ability of a designer to adjust the voltage, frequency or load capacitance.

For example, the embedded system designer has limited control over C, the internal load capacitance. The capacitance is a function of the internal MCU layout and design. It is up to the MCU manufacturer to limit the switching of load capacitance by utilizing proper low-power IC design techniques, such as properly gating clock signals. The only control the system designer has over internal load capacitance is the ability to enable and disable MCU features individually. A savvy low-power designer should ensure that, at any point in a program, only the currently needed features of the MCU are enabled and all others are turned off.

A designer does have control over the external load capacitance of a signal that is routed to an I/O pin. These capacitances can be much larger than the internal capacitance of the device and can cause significant losses. For this reason, it is important for a designer to review a design for stray capacitance on digital switching. Refer to the **"Hardware Design"** section for more details on I/O low-power design techniques.

Operating voltage is primarily defined by the process technology used in the manufacture of the MCU. As process geometries shrink, the operating voltage decreases and the device consumes lower dynamic power. An embedded system designer can utilize this knowledge by selecting MCUs which are capable of operating at lower voltages. However, if the minimum system voltage is defined by some other component of the system, such as a sensor or communications interface, this will require a cost versus power trade-off. This would require an additional voltage regulator for the MCU power, which would increase system cost. Interestingly, in some cases, it can be more power efficient to run at a higher voltage if it allows for the removal of a regulator. In this case, the power lost in the regulator is higher than the increase in dynamic current from operating at a higher voltage.

Frequency is typically the most variable of the factors contributing to dynamic power, and as such, is usually the component adjusted by embedded designers to actively control power consumption. The optimal operating frequency for a system is determined by a combination of factors:

- Communications or sampling speed requirements
- Processing performance
- Maximum peak current allowed

As the power equation indicates, lower frequencies will result in lower dynamic current. However, it is important to keep in mind that execution speed is also a factor in power consumption. In some cases, it may be optimal to run at a higher frequency and finish an operation more quickly to allow the system to return to Sleep for minimal power use. Also, consider that at low frequencies, dynamic switching current may no longer dominate system power consumption. Instead, the static power consumption used in biasing the analog circuits on the MCU will dominate. This can limit the effectiveness of reducing frequency as a power-saving technique. At this point, the designer should focus on techniques to reduce static power.

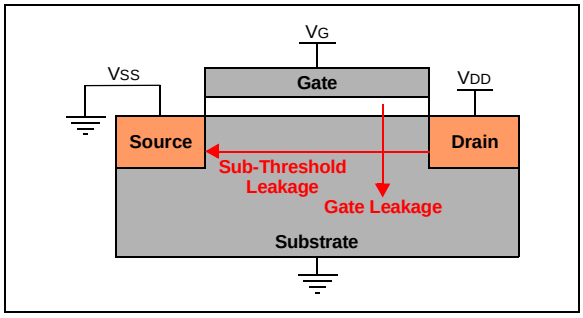
STATIC POWER

Static power consumption encompasses all of the power required to maintain proper system operation while code is not actively running. This typically includes bias currents for analog circuits, low-power timekeeping oscillators and leakage current. Static power is a major concern for battery-based systems, which spend significant portions of the application lifetime in Sleep mode.

Analog circuits, such as voltage regulators and Brown-out Resets (BOR), require a certain amount of bias current in order to maintain acceptable accuracy as temperature and voltage vary. In order to offset the current consumption of many of these modules, the best techniques are to utilize flexibility built into the MCU to enable and disable analog blocks, as necessary, or to utilize lower power and lower accuracy modes.

Timekeepers, such as the Watchdog Timers (WDT) and Real-Time Clock/Calendars (RTCC), are also usually considered part of the system's static power consumption. Even though these modules are actively switching and could be thought of as dynamic because they are always running at a constant frequency, and consume very little power, it makes more sense to include them in static power calculations. Nonetheless, as dynamic circuits, they follow the same rules for power optimization, as mentioned in the **"Dynamic Power"** section and are primarily affected by voltage, frequency and capacitance. By nature, these oscillators are low-frequency and low-power oscillators. For instance, 32 kHz crystal drivers are usually designed to allow the crystal to operate with as little peak-to-peak voltage as possible, while maintaining stable oscillation. Refer to the **"Low-Power Crystal Design"** section for more information on reducing crystal oscillator power consumption.

FIGURE 2: NMOS TRANSISTOR LEAKAGE CURRENTS



Leakage current is caused by the non-ideal operation of the MOSFETs used in CMOS devices. As process technologies shrink and transistors become smaller, there are several aspects of the FET which no longer behave as they would in an ideal system. Current begins to leak from the FET drain to the source, even when the gate is below the conducting threshold. This current is called sub-threshold leakage. Sub-threshold leakage occurs because the drain and source are physically closer in a narrower transistor. The narrower a transistor is, the larger this leakage becomes. Additionally, leakage is affected by temperature and voltage. For MCUs, using small processes at high temperatures and maximum voltage, leakage can amount to many μA of current.

Similar to dynamic power, some of the aspects of leakage power are outside of the control of the embedded system designer. Selecting an MCU with a larger process technology will reduce leakage, but with the trade-off of having higher dynamic current consumption. Temperature is also out of the designer's control, set by the requirements of the system. Generally, the best option for reducing leakage is to reduce voltage and power off unneeded circuits. Some PIC® MCUs provide features, such as Deep Sleep, which power down additional circuits in the device to reduce the power consumption below typical Sleep levels.

PROCESS TECHNOLOGY TRADE-OFF CONSIDERATIONS

A critical trade-off becomes apparent when trying to optimize both dynamic and static power. Smaller process technologies have considerably lower dynamic power, but at the cost of much higher leakage current. Table 1 shows a comparison of some commonly used process technologies, and the relative static and dynamic power consumption for each technology. In some cases, as a result of this trade-off, it can be difficult for a designer to determine which is more important to reducing power consumption for a system. In order to select the correct MCU, which will minimize power for a particular system, it is important for the designer to know whether dynamic or static power has a more significant impact on the system. This process is called 'Power Budgeting' and is discussed in detail in the **"Measuring Power Consumption"** section. This trade-off between dynamic and static power is what makes MCU design optimizations matter. MCU manufacturers are constantly searching for ways to maintain low leakage as process technologies continue to shrink.

TABLE 1: COMPARISON OF PROCESS TECHNOLOGY POWER CONSUMPTION

Process Technology	Leakage Power (Normalized)	Dynamic Power (Normalized)	Core Voltage
0.35 μm	0.5	2.8	3.0V
0.25 μm	.75	2	2.5V
0.18 μm	1	1	1.8V
130 nm	1.5	.75	1.5V
90 nm	2	0.44	1.2V

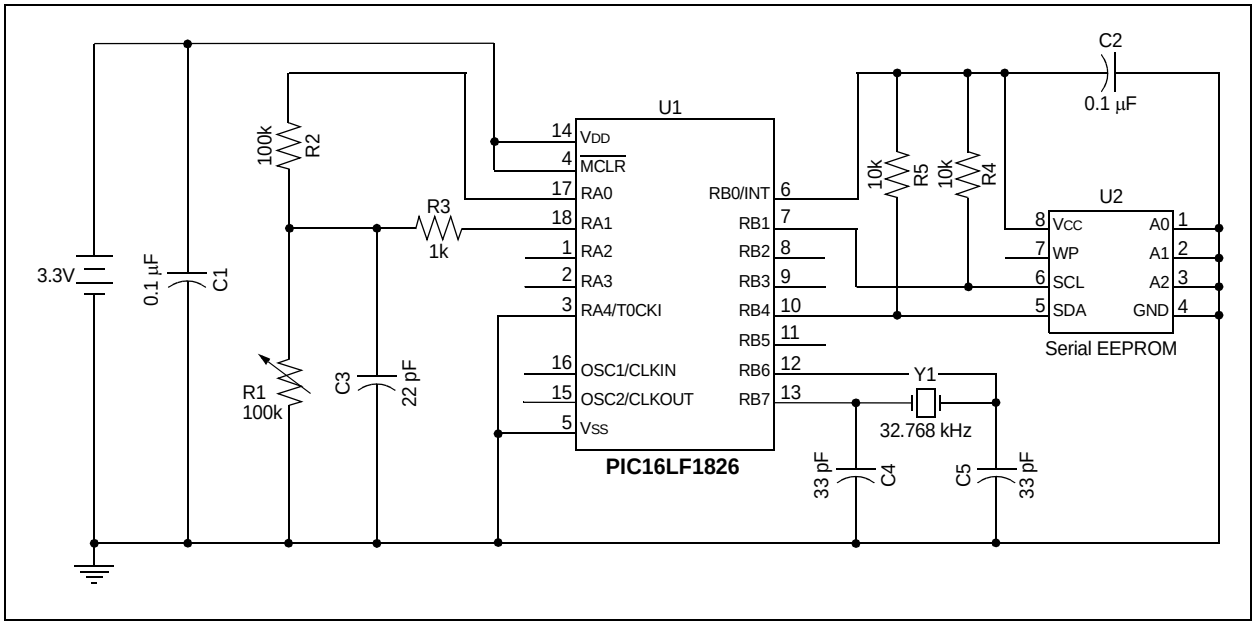
WHAT IS LOW POWER?

Low power means different things for different systems. Some applications focus on dynamic power consumption as they must remain running constantly, such as a power supply. For these applications, the only concern is how to reduce dynamic power as much as possible to improve efficiency. On the other hand, a battery-powered application would typically be more concerned with the Sleep mode power consumption, as it tends to spend most of its time in Sleep mode.

However, these Run and Sleep mode power consumptions are not the only aspects that are important to power consumption. For example, wake-up time can have a crucial impact on systems with a low active duty cycle. Consider a system which only remains awake for 10 µs to read and store an input before returning to Sleep. In this example, it is not feasible to start up a crystal oscillator which could take milliseconds before it is stable. The MCU must be able to wake-up and operate in a few microseconds to effectively manage the application.

To understand the importance of the various aspects of the system's power consumption, the example application, shown in Figure 3, will be used to show various low-power design techniques and trade-offs. A data logging sensor takes a sample every 100 mS and then stores the data to EEPROM once a full page of data is available (32 samples). The system runs for 50 µS to take a sample and for 5 mS to store the data to EEPROM. The sensor is using the PIC16LF1826, a 3V XLP PIC microcontroller.

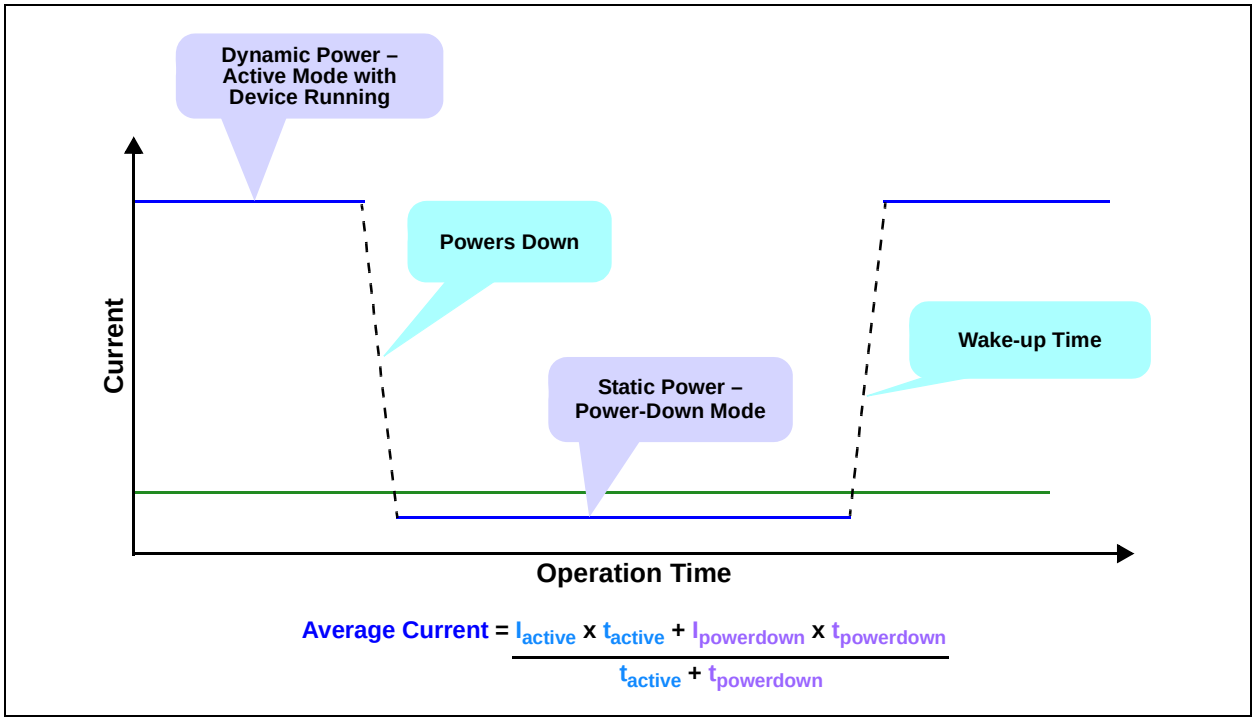
FIGURE 3: LOW-POWER SYSTEM EXAMPLE



Measuring Power Consumption

When measuring the overall power consumption of a system, there are two values which are of primary concern: average power consumption and maximum power consumption. Average power consumption is the sum of the total energy consumed by the system in Dynamic and Static Power modes, divided by the average system loop time, as shown in Figure 4. Average power is important because it provides a single value, which can be used to accurately determine battery life or the total energy use of the system.

FIGURE 4: CALCULATING SYSTEM AVERAGE POWER CONSUMPTION



Maximum power consumption is the worst-case power draw required by the system. It is important to determine maximum power consumption in order to properly design the system power supply. For example, many batteries perform differently depending on the rate of current draw from the battery, and it is important to know what current levels the system's battery is capable of handling.

While measuring power consumption may seem straightforward, accurately capturing average power can be complex for many systems. Most embedded applications do not spend enough time doing a specific task for an ammeter to accurately measure and display the current without modifying the code to wait in a particular state. An oscilloscope can be used to get an idea of the current profile, but may not be able to accurately capture the power consumption for Low-Power modes. As a result, it is often necessary to combine multiple measurement methods in order to accurately model the power consumption of a system in all modes.

MEASURING STATIC POWER

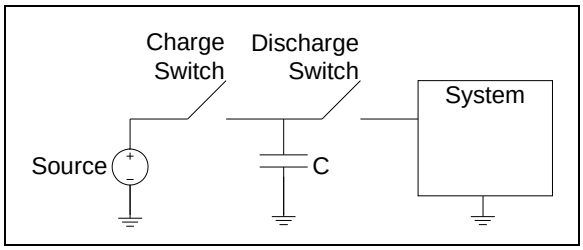
There are a few concerns to consider when measuring static power consumption. First, ensure that the ammeter used for testing has a high enough resolution to accurately measure the MCU's power consumption. For many nanoWatt XLP MCUs, the static power can be lower than 100 nA, which can be outside the maximum resolution for most handheld multimeters and some bench DMMs.

Another concern is the voltage drop out of the ammeter. When set to a low-current range for measuring static power, there will usually be a notable voltage drop across the meter. This can interfere with voltage-sensitive circuits, and often, a Brown-out Reset will occur in an MCU if a short, high-power pulse develops. This voltage drop is frequently not specified in the product documentation and must usually be measured by using a second multimeter or oscilloscope.

Generally, it is not feasible to measure static power with an oscilloscope. For currents under 10 μA, a shunt resistor of at least 5 kOhm must be used in order to create a measurable voltage drop. Using this large of a resistor is likely to interfere with proper operation of the system when not in the low-power state.

In the event that there is no ammeter available that is capable of accurately measuring a very low-power application, an alternate method can be used, utilizing a capacitor. When operating in a Low-Power mode, an MCU will act as a constant-current sink. Using the equation for constant-current discharge from a capacitor, it is relatively easy to measure the power consumption of a low-power system. Connect a capacitor, such as a low leakage 10 μF ceramic capacitor, to the application, as shown in Figure 5, with switches to disconnect the capacitor from the voltage supply and the application.

FIGURE 5: LOW-POWER MEASUREMENT, CAPACITOR METHOD



To measure low power in the capacitor method:

1. Connect both switches to run the system and allow the capacitor to charge to V_{DD} from the voltage source.
2. Disconnect the voltage source (and any voltage meters) and allow the application to run from the capacitor for a set amount of time. This time should be long enough to allow the capacitor to discharge by about 100 mV. Make sure to disconnect the capacitor before the voltage has dropped below the normal operating range of the application.
3. Disconnect the capacitor from the application and connect a voltage meter to measure the remaining voltage on the capacitor.
4. Using Equation 2, calculate the current for a capacitor under constant-current discharge, based on the delta voltage and discharge time.

Note: Some capacitor types have significant leakage current which could cause error in this method. To account for the leakage current, repeat the experiment for the same time period with no load on the capacitor to determine the amount of voltage change from leakage.

For example, if a system is run from a 10 μF capacitor for 10 seconds, and experiences a voltage drop of 100 mV, using Equation 2 indicates that the system consumes an average of 100 nA over that 10 seconds.

EQUATION 2: CONSTANT-CURRENT DISCHARGE FROM A CAPACITOR

$$I = C\frac{\Delta V}{t}$$

MEASURING DYNAMIC POWER

Using an ammeter to measure dynamic power under normal system operating conditions is usually not very useful. This is because the sampling speed of most ammeters is not fast enough to accurately measure the real-time power consumption of the system, as it is executing code and changing states. To accurately measure dynamic power with an ammeter, it is necessary to modify the system code to hold it in a particular state in order to measure the power. This provides accurate data for the current consumption of each state, but doesn't provide the execution time information needed to calculate the average power consumption of the system.

An effective way to measure dynamic power consumption is to use an oscilloscope to measure voltage across a shunt resistor. The oscilloscope will allow a designer to determine the changes in power consumption as the system steps through various states during operation, as well as measure the time spent in each state. This facilitates the creation of a complete profile of the application's power consumption. It is important to size the shunt resistor appropriately. It should be large enough to provide measurable resolution on the scope, but small enough not to cause a system brown-out in high-power states. Usually, a 10-100 ohm resistor is appropriate for dynamic power measurements.

CREATING A POWER PROFILE

Once data is collected for both static and dynamic power, the data can be used to create a power profile for the system. The purpose of a power profile is to provide the designer with a clear image of where the primary sources of power use are in the system so that it can be optimized. To create a power profile:

1. Break down the application into states based on varying power consumption.
2. Measure the power and execution time of each state.
3. Determine the total energy consumed in each state by multiplying power and time.

TABLE 2: EXAMPLE POWER PROFILE CALCULATION

Mode	Mode Time for 32 Samples (ms)	Current (μA)		Charge (μA × ms)
		By Device	Mode Total	
Sleep MCU Sleep Sensor Off EEPROM Off	3200	0.8 0 0	0.8	2560.0
Initialize MCU Wake-up Sensor Off EEPROM Off	0.32	0.8 0 0	0.8000	0.26
Sample Sensor MCU Run Sensor On EEPROM Off	1.28	150 16.5 0	166.5	213.12
Scaling MCU Run Sensor Off EEPROM Off	0.32	1300 0 0	1300	416.00
Storing MCU Run Sensor Off EEPROM On	5	150 0 1000	1150	5750.00
TOTAL	3206.92	—	—	8939.38

Average Current (μA), Total Charge/Total Time	2.788
Peak Current (μA)	1300

Once the profile is complete, the task of optimizing the application is much simpler. The profile clearly indicates which states of the system consume the most power so that the designer can focus his efforts on reducing the power of these states. The power profile also simplifies the calculation of the system average power and the maximum power. Consider the profile in Table 2, which is based on the example application from Figure 3. For each Microcontroller mode, the execution time is calculated and current consumption is broken down by device. Note that in this profile, the worst power consumers are the Storing and Sleep modes. A designer, using this system, should focus power reduction efforts on these two modes first, as they have the highest impact on the system.

Performance vs. Power

There are always trade-offs between performance and power consumption in an embedded system. The key to low-power design is creating a system which utilizes the strengths and features of the controller to get the most performance within the power budget. Some critical aspects to consider when designing for performance are:

- Wake-up Time
- Oscillator Speed
- Instruction Set Architecture (ISA)
- Peripheral Features
- Executing from Flash vs. RAM

Wake-up Time

Wake-up time is critical for systems with short active times, as the wake-up process often consumes a similar amount of current as normal operation.

The primary component of wake-up time is the Oscillator Start-up Time (OST). Typical start-up times for various common clock sources on PIC MCUs are shown in Table 3. Note that most high-accuracy and high-speed sources, such as crystals and Phase-Locked Loop (PLL) oscillators, have long start-up times. One method, which allows a system to use a slow start-up source with a fast wake-up time, is the Two-Speed Start-up feature on many PIC MCUs. Two-Speed Start-up initially wakes up from the internal RC oscillator, which typically starts up in a few μs. The system runs from this oscillator until the primary clock source stabilizes, reducing total operating time by running any code that is not timing critical, as shown in Figure 6.

FIGURE 6: TWO-SPEED SCOPE

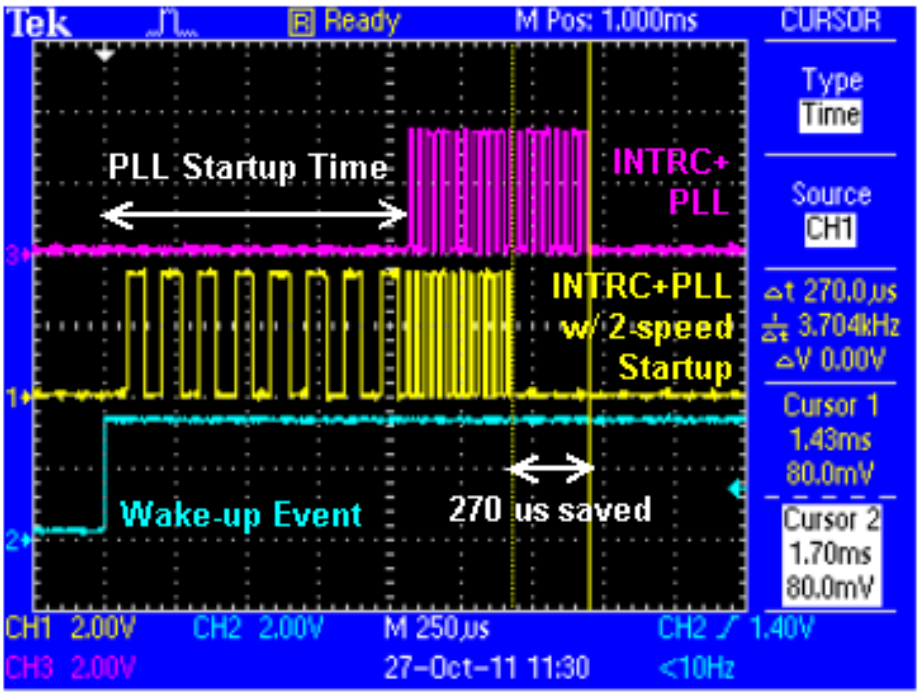


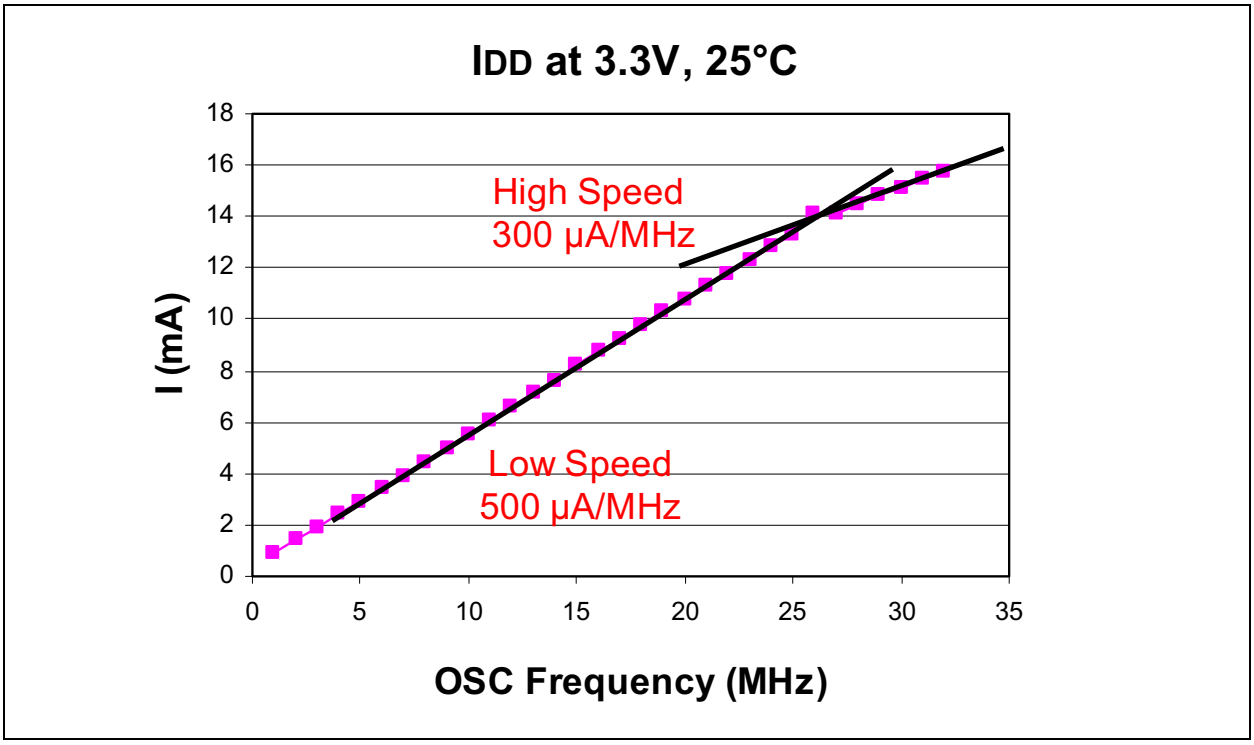
TABLE 3: TYPICAL WAKE-UP TIMES FOR VARIOUS OSCILLATORS

Oscillator	Frequency	Start-up Time (ms)
Internal Fast RC Oscillator	8 MHz	.001-.010
Low-Power RC Oscillator	31 kHz	0.3
Primary Crystal	8 MHz	0.5-1.0
Internal RC + PLL	32 MHz (8 MHz x 4)	1.0
Primary Crystal + PLL	32 MHz (8 MHz x 4)	1.5-2.0
Secondary Crystal	32.768 kHz	100-1000

Clock Speeds and Power Efficiency

One of the most common questions about low-power design asks: What is the best frequency to run at in order to minimize power consumption? Is it better to run at the lowest speed possible or to run at high speed and then Sleep afterward? While this will vary by MCU, usually higher speeds are more efficient than low speeds. This occurs primarily because of the fixed current of initially powering the oscillator and the MCU. At low frequencies, this fixed bias current represents a significant portion of the total power consumption. However, at higher frequencies, it becomes negligible. The result is that the higher frequencies typically have lower μA/MHz, allowing for more efficient operation. Refer to Figure 7 to see how higher speeds can be more efficient.

FIGURE 7: DYNAMIC CURRENT vs. OSCILLATOR SPEED, DEMONSTRATING EFFICIENCY OF HIGH SPEEDS



There are a couple of caveats to this rule to consider. If the power source is low impedance and capable of handling high currents without an issue, it remains true. However, for higher impedance sources, such as coin cell batteries, it no longer holds. With lower current power sources, the internal impedance of the power supply will reduce the effective power output to the MCU when high current is consumed. This can result in the reduction of the battery life below the rated capacity. It can also cause unreliable system operation due to unexpected Brown-out Resets when the battery's output voltage drops, due to the high-current load.

Second, at higher speeds, the system voltage often becomes important. Many MCUs are not capable of operating at full speed through the entire voltage range of the device. Therefore, when running at high speed from a battery-powered application, it is necessary to monitor VDD to determine if the battery voltage is dropping close to the minimum level required for full-speed operation. If VDD drops below this voltage, it can cause code mis-execution. Most MCUs provide a Low-Voltage Detect (LVD) feature, which will interrupt the device if VDD drops close to this range. This will allow the firmware to reduce the operating frequency at low voltage, allowing the system to extend its lifetime.

Instruction Set Architecture (ISA)

The primary dynamic power specification, advertised by MCU manufacturers, is µA/MHz or µA/MIPS. Unfortunately, as many designers have discovered, neither of these specifications are very accurate as they don't take into account the MCU's CPU architecture. To truly compare power consumption, it is necessary to consider how much actual work is performed per unit of energy consumed. Different architectures will perform varying amounts of work in a single clock cycle. This makes the Instruction Set Architecture of an MCU an important component of power consumption. Clocking scheme, cycles per instruction and available instructions all have a major impact on the performance of the device, which directly affects the amount of work done in one cycle, and therefore, the power consumption.

The first major considerations are the MCU clock ratio and cycles per instruction. The clock ratio is the ratio of the input system clock frequency (FOSC) to the internal instruction clock frequency (FCY or SYSCLK). Eight-bit PIC MCUs have a divide-by-4 architecture, while 16-bit PIC MCUs have a divide-by-2 architecture. PIC32 devices have a 1-to-1 clock ratio. It is important to consider the clock ratio when checking the specifications for a device. In some cases, devices specify power consumption based on FOSC and others based on FCY. Devices which specify FOSC will refer to the speed in MHz (e.g., 8 MHz), while a device specifying FCY will use MIPS (e.g., 8 MIPS).

However, clock ratio cannot be considered alone when comparing power consumption. The other important component to the equation is cycles per instruction. All PIC MCUs are RISC architectures, with most of the instructions taking one cycle to execute. Even so, many other MCUs (even some which advertise as RISC) require multiple cycles per instruction. This difference can make it very difficult to directly compare power consumption between different architectures. In some cases, if the majority of the instructions on an MCU require a given cycle count, an average cycle/instruction value can be calculated and used for power comparisons; others are complex enough that this is not possible. If this is the case, it is generally only possible to compare power consumption by using a benchmarking test.

Benchmarking is the only complete method to compare devices which take Instruction Set Architecture into account. It is also the most time-intensive method to compare devices. For a simple benchmark, there are some open source benchmark standards, which can be used to do get an initial comparison. By combining benchmark completion times with data sheet power consumption specifications, it is often possible to get a reasonable comparison of device power consumption. Even so, these benchmarks usually utilize only the CPU of a device and do not exercise device peripherals or advanced features. An ideal comparison is to use a simplified application, which performs the major time or power consuming portions of a system, and performs a power budget analysis using this simplified system as a more accurate benchmark.

Properly Utilizing Peripherals

Peripheral features on a microcontroller can help substantially reduce power consumption. However, often it can be deceiving which peripherals consume high power and which are low power. For example, when comparing a UART to an SPI, many designers would choose the UART as the lower power module. Because it requires fewer toggling I/Os and runs at a lower speed, this limited speed actually makes the UART worse for power consumption, as an SPI is usually able to complete a transaction much more quickly than a UART. While the SPI may consume more power during the transaction, afterward the device can go to Sleep and eliminate the highest source of power consumption, the CPU. Table 4 provides a list of some of the common features on MCUs and gives a rough baseline for their estimated power consumption. Of course, this will vary by vendor, peripheral settings and by application, but it should provide a basis for determining which module to use when multiple choices are available.

The following sections provide some further tips on how to effectively use various peripherals to reduce the power consumption of a system.

TABLE 4: POWER CONSUMPTION OF COMMON PERIPHERALS

Serial Communications	**Current (μA)**	**Time to Send 10 Bytes (ms)**	**Total Charge (μA × ms)**
UART (57.6k)	200	1.74	347.22
I^2C™ (400 kHz)	1000	0.25	250.00
SPI (4 MHz)	700	0.02	14.00
Analog Modules	**Current (μA)**	**Time to Convert 10 Samples (ms)**	**Total Charge (μA × ms)**
Low-Speed A/D (100 ksps maximum)	250	0.1	25
High-Speed A/D (500 ksps maximum)	1000	0.02	20
Low-Speed Comparator	10		
High-Speed Comparator	100		
DAC/CV_{REF}	5		
Low-Power Modules	**Current (μA)**		
Brown-out Reset (BOR)	5		
Watchdog Timer (WDT)	0.5-1.0		
Real-Time Clock and Calendar (RTCC)	0.5-1.0		
Timer (31 kHz)	0.5-1.0		

ANALOG-TO-DIGITAL CONVERTERS (A/D)

The Analog-to-Digital Converter (A/D) is primarily an analog module and many of the circuits used to perform conversions do not vary significantly with speed, such as reference voltage circuits, signal amplifiers, signal buffers, etc. Because these analog circuits make up the bulk of power consumption for the A/D, it is usually beneficial to run an A/D at a higher speed than necessary for the application, disabling it in-between samples. The A/D should be set to use the fastest conversion clock possible and reduce the sampling time to the minimum necessary to maintain measurement accuracy.

DMA AND FIFO BUFFERS

A Direct Memory Access (DMA) controller is a powerful tool to reduce power consumption. The DMA improves performance by off-loading data transfer tasks from the CPU. Any features which can reduce CPU run time can help to drastically reduce power consumption, as clocking the CPU is the most power-intensive task in an MCU.

Many peripherals, capable of operating in Sleep, have built-in FIFO buffers which reduce power by enabling longer Sleep times. The FIFOs will store received or sampled data and will interrupt once the buffer is full. This allows the device to only wake up once to process the data. This consumes much less power than fully waking up a device to allow the CPU to handle each individual transfer.

BROWN-OUT RESET (BOR)

Brown-out Reset (BOR) protection circuits are a double-edged sword for low-power applications. On one hand, they protect the application from mis-execution as batteries die, or when high transient currents cause dips on the voltage supply. On the other hand, they tend to consume high power while a device is in Low-Power modes. An average BOR, in order to maintain accuracy, typically has a bias current of about 5 μA.

It is important to utilize flexible features on a BOR in order to have the best power performance. Some useful power reducing BOR features are:

- **Automatically disabled BOR in Sleep mode.** Many PIC MCUs have the ability to automatically disable the BOR when the device enters Low-Power mode. Because the primary purpose of a BOR is to protect the device from mis-execution of code at low voltage, it is often no longer needed when the CPU is not running. Therefore, it is valuable to have a module that is automatically disabled in Sleep and automatically re-enabled when waking up the CPU.
- **Accuracy and power setting controls.** Some PIC MCUs have a BOR with programmable current consumption. This allows a designer to choose the current range, which makes the most sense for the system, based on the accuracy and voltage levels required.
- **Low-Power BOR or “Deep Sleep” BOR mode.** Sometimes an application doesn’t require protection from mis-execution in the form of a constantly active BOR. However, completely disabling the BOR circuit can leave an MCU vulnerable to lock-up if the supply voltage drops close to the transistor threshold voltage, without dropping all the way to ground. As a result, many newer PIC MCUs have a Low-Power BOR (LPBOR), which provides downside protection to ensure the MCU will always have a Power-on Reset (POR). This mode typically consumes about 50 nA to enable, considerably less than a full-fledged BOR.

WAKE-UP SOURCES

The proper selection of wake-up sources is often an overlooked MCU feature, which can help reduce power consumption. Modern MCUs have a wide variety of wake-up sources and can wake up from almost any peripheral module on the device. While many applications will traditionally wake up, based on a timer to check for events, power can be reduced by enabling a peripheral, which is specifically capable of capturing a particular event. For example, I^2C™ and SPI modules can receive data while asleep, then wake up the device afterward. This can save power over having to wake the device every few milliseconds to check for incoming transmissions.

Another example is the value of a Real-Time Clock (RTC) versus using a timer module for wake-up. Keeping time with a timer module usually requires waking a device, once every second, to update the clock and check for an alarm condition. However, a Real-Time Clock and Calendar (RTCC) will allow the device to keep time and remain asleep until the alarm occurs, reducing power.

Executing from Flash vs. RAM

Powering and reading the Flash memory of an MCU is one of the most significant contributions to power consumption. Reading from RAM consumes less power than Flash. To take advantage of this difference, many high-end MCUs, such as the PIC32, provide an option to execute code from RAM to help reduce dynamic power consumption. However, there are trade-offs and overhead to consider in order to determine if it is applicable to a particular application.

In order to run from RAM, the code must be copied from Flash to RAM. This task consumes significant time and power. In order to be worthwhile, the functions run from RAM, which generally should be limited in size or should be computationally-intensive functions.

A potential trade-off to consider is that for some device architectures, running from RAM can reduce performance due to the limitations of using a single memory bus for execution and operation. For example, the Harvard architecture used in PIC32 devices, normally uses one memory bus to read from Flash and another to read from RAM. This allows the device to run at high speed without stalling the processor. However, when operating from RAM, only a single memory bus is used, which can force the processor to stall during some instructions which write to RAM. Make note of these potential differences on a device when calculating the value of executing from RAM.

SOFTWARE DESIGN

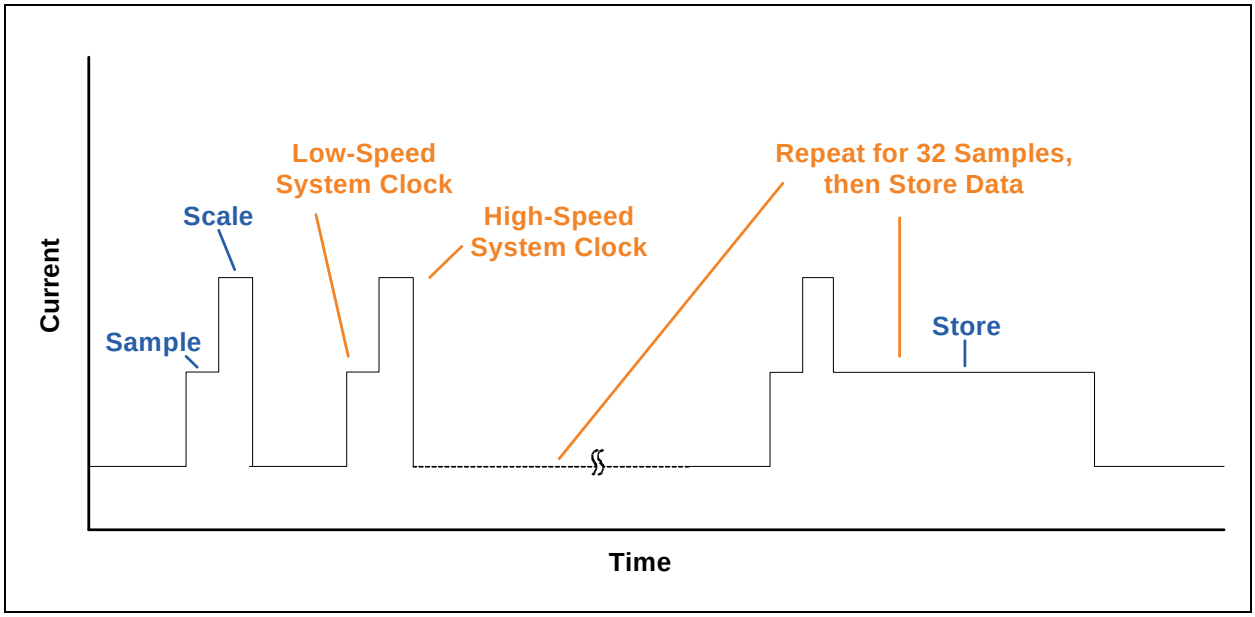
Software design has a critical impact on low-power system design. Selecting the best low-power devices, and optimizing hardware for low power, can easily be wasted if the proper techniques are not utilized when writing system software. The following sections provide some general tips for reducing power consumption in software.

Conditional Code Execution

One software technique, which can be effectively used in many applications, is to utilize a conditional wake-up and code execution structure. For example, Figure 8 shows a graph of the power consumption of the sensor presented in Figure 3. The sensor system takes 32 samples before writing to the EEPROM. This allows the system to minimize power by running for a very short time on most wake-up events and ensuring that the high-current operation of writing to Flash is done as infrequently as possible. If each sample was written individually, the system would consume significantly more current.

Additionally, this system uses a varying system clock to further improve power. This system uses a lower speed clock source, during the sampling and storing processes, and only uses the high-speed clock for the scaling process. This is because the sampling and storing times are fixed by the A/D and EEPROM which are used in the system. Reducing the frequency during this time, reduces current consumption without increasing run time. The scaling time depends only on the performance of the MCU and varies with operating frequency. Therefore, during this time, it is optimal to run quickly allowing the system to return to Sleep afterward.

FIGURE 8: SENSOR POWER CONSUMPTION



Idle and Doze

Generally, it is easy to tell when Low-Power modes, such as Sleep, should be used in an application, as there is an obvious down time when the system can enter Sleep mode. Dynamic power-saving modes, Idle and Doze, tend to be less obvious. For example, some cases when a system should use Idle or Doze mode are:

- **Replacing Wait loops.** Low-power applications should avoid loops where the CPU is performing time-wasting actions, such as polling a bit. System designers should replace these loops with a call to Idle mode, such as follows:

  Replace:
  ```
  while(!ADCInterruptFlag);
  ```
  With:
  ```
  while(!ADCInterruptFlag)
  {
    Idle(); //wake on ADC interrupt
  }
  ```
  This will have the same effect in the code, but at a much lower power cost, as the CPU will not be clocked until the A/D interrupt wakes up the device. Note that this will not impact other functionality of the device, as other interrupts can wake up the device and be handled properly, with the device returning to Idle afterwards. In cases where the CPU needs to perform some processing while waiting on the flag, consider using Doze mode rather than Idle to run the CPU at a slower speed while waiting.
- **Running high-speed peripherals.** Consider the scenario where a system's primary job is to maintain a high-resolution PWM output. The high-resolution requirement will force the system to maintain a high-speed clock. In this case, Sleep mode cannot be used to reduce power consumption due to the need to maintain the PWM. Similarly, Idle mode may not be feasible because the CPU will need to monitor the system state to update the PWM, as necessary. Even so, it is unlikely that the CPU will need to run at the same speed as the PWM in order to perform this job. In this case, Doze mode is an ideal option, allowing the CPU to perform the necessary tasks at low speed.

Interrupts vs. Polling

In the effort to wipe out power draining Wait loops from the system, a low-power system designer should consider switching from a polling-based system to an interrupt-based system. Utilizing interrupts allows the system to use power-saving modes, such as Sleep and Idle, much more frequently in place of waiting for an event with polling. Consider the sensor presented in Figure 3 and detailed in Table 2. If this application required a significant amount of processing time for each sample, it would not make sense to wait in Idle mode, polling the A/D flag while taking the sample. Instead, an interrupt-based method could be used so that the system could process the previous sample while taking the current one. If the system is simple and doesn't require much processing, or have background tasks which need to run frequently, polling can be the better option as it allows simpler, more compact code without any additional current consumption.

Power Optimization and 'C'

'C' is a great tool for writing embedded programs, but compared to Assembly, C takes away much of the designer's control over the code. As a result, some engineers are hesitant to use C in applications which require minimal power consumption, preferring to use Assembly to manually optimize the code.

While it is true that code written by highly experienced engineers using Assembly will almost always be more efficient than C, current compilers have optimizations that minimize the differences. As a result, it is often worthwhile to use C for the code portability and maintenance improvements it offers over Assembly. In fact, as the complexity of a project increases, the likelihood that the C compiler will be more efficient than handwritten Assembly increases as well. Assembly has the most value when used for very small programs or small sections of larger programs, which are processing-intensive and can be more easily hand optimized.

There are optimizations which can be done by the programmer, in C or Assembly, which cannot be performed by a compiler optimizer. One example of an optimization, which can drastically improve code, is to simplify program flow for common cases. Consider the case where an A/D sample has been completed and it is identical to a previous sample. Optimized code would compare the current sample to the previous sample, as well as other common values prior to performing time-intensive processing steps. If the new sample matches one of these values, the code can use a precalculated result and avoid significant processing time. In some cases, this modification to the program architecture can drastically reduce total execution time. Most applications include some cases that appear much more frequently than others in many of the application's functions, which can take advantage of this method.

HARDWARE DESIGN

Board Level Considerations

Designing low-power hardware follows a few, very straightforward requirements:

- Maximize impedance in current paths
- Minimize impedance in high-speed switching paths
- Minimize leakage currents
- Minimize operating duty cycles

The following recommendations utilize these rules, in various circuits of a system, to minimize the power impact of those components.

PUSH BUTTONS

The primary problem with push button power consumption is that buttons typically interact in the very slow human time domain, rather than the high-speed embedded system domain. A very quick button press will still take hundreds of milliseconds. With a 10 kOhm pull-up on a button, this means hundreds of milliseconds of 300 μA of power, which is considerable in a low-power system. Increasing the value of the pull-up can help to reduce power, but at high resistances, may reduce noise immunity. Additionally, increasing the size of the pull-up increases the time constant of the signal, which increases the rise time before high voltage is detected.

A better method is to use the MCU's internal pull-up resistors as the voltage source for a push button. Because the internal pull-up can be enabled and disabled, dynamically in code, as soon as the button press is detected, the MCU can disable the pull-up, removing the current path to ground. This improves the push button circuit in other ways as well, as it provides a simple debouncing and reduces the external component count. For a button which needs to be able to detect a push-and-hold state, the pull-up can be periodically re-enabled to determine if the button is being held.

LEDs

LEDs are a power consumption problem for similar reasons as buttons. They consume high power to run (2 mA-50 mA) and must be run for long periods of time in order to be useful to a user. The following are two of the main techniques for reducing LED power consumption:

- Drive the LED at a lower duty cycle using a PWM. This gives the code some degree of control over the LED power consumption by being able to dynamically reduce the brightness, if necessary.
- Drive the LED at lower current levels by increasing the size of the current-limiting resistor. This is an especially useful technique with newer, high brightness and high-efficiency LEDs. These LEDs can be blindingly bright when driven at the rated current levels. However, for a system which doesn't require a bright LED, they can be driven at much lower currents and still remain highly visible. While a 2 mA LED might not be very visible below this rating, a high brightness LED, rated for 20 mA, can often still be highly visible when driven under 1 mA.

CONNECTING AND CONTROLLING I/O PINS

In many applications, incorrect use of the I/O pins is a common source of unexpected, high-power consumption. There are a few rules for using a bidirectional I/O pin on a PIC microcontroller to minimize power consumption.

Unused Port Pins

By default, PIC microcontroller I/O pins power up as inputs. A digital input pin consumes the least amount of power when the input voltage is near V_{DD} or V_{SS}. If the input is at some voltage between V_{DD} and V_{SS}, the transistors inside the digital input buffer are biased in the linear region and they will consume a significant amount of current. On an unused I/O pin, which is left floating, the voltage on the pin can drift to $V_{DD}/2$ or oscillate, causing the unused I/O pin to consume significant power. The I/O pin can use as much as 100 μA if a switching signal is coupled onto the pin.

An unused I/O pin should be left unconnected, but configured as an output pin, driving to either state (high or low), or configured as an input with an external resistor (about 10 kΩ) pulling it to V_{DD} or V_{SS}. If such a pin can be configured as an analog input, the digital input buffer is turned off, preventing the excess current consumption caused by a floating signal. Any of these methods will prevent the floating node case, minimizing power.

Analog Inputs

Analog inputs have a very high impedance so they consume very little current. They will consume less current than a digital input if the applied voltage would normally be centered between VDD and VSS. Sometimes, it is appropriate and possible to configure digital inputs as analog inputs when the digital input must go to a low-power state.

Digital Inputs and Outputs

As long as a digital input is pulled to VDD or VSS, only the pin input leakage current will be consumed.

There is no additional current consumed by a digital output pin, other than the current going through the pin to power the external circuit. Close attention should be paid to the external circuits to ensure that the output is being driven to the state which causes the lowest power consumption. If an external circuit is powered down, make sure that any I/O pin connected to it is driven low to prevent sinking current through the disabled circuit.

For digital inputs and outputs with high switching frequencies, make sure that there is no stray capacitance on the bus by minimizing trace length and eliminating unnecessary components.

SIZING PULL-UP RESISTORS

I/O lines with pull-up or pull-down resistors can be a high source of power drain in an application and are often targeted as a source of power optimization. As noted before, push buttons and LEDs have power consumption set by the size of the resistor used. For a low-power application, the general rule is to use pull-ups that are as large as possible. However, a designer should take note of the effect this will have on the circuit. For example, on an I^2C module, the pull-up size determines the maximum speed of the bus. Using pull-ups that are too large will increase the time constant of the bus and slow down communications. This could result in a net increase in current consumption.

CAPACITOR LEAKAGE CURRENT

All capacitors have a small amount of charge loss through the dielectric, even after fully charging. This loss is typically referred to as the ‘capacitor’s leakage current’. The amount of leakage current varies by capacitor size and type. Typically, tantalum and electrolytic caps are high leakage, and ceramic and film capacitors are low leakage.

TABLE 5: LEAKAGE CURRENTS FOR COMMON CAPACITORS

Capacitor Leakage (10 μF)	
Electrolytic	5 μA
Tantalum	1 μA
Ceramic	20 nA
Film	5 nA

Leakage specifications vary for different capacitors. In some cases, leakage current is specified directly. In other cases, it is determined by the capacitor’s Insulation Resistance (IR). The Insulation Resistance is typically specified in either Megaohms or Megaohms × Megafarads. To determine the leakage current in the latter case, use the following formula:

EQUATION 3: CAPACITOR LEAKAGE CURRENT CALCULATION

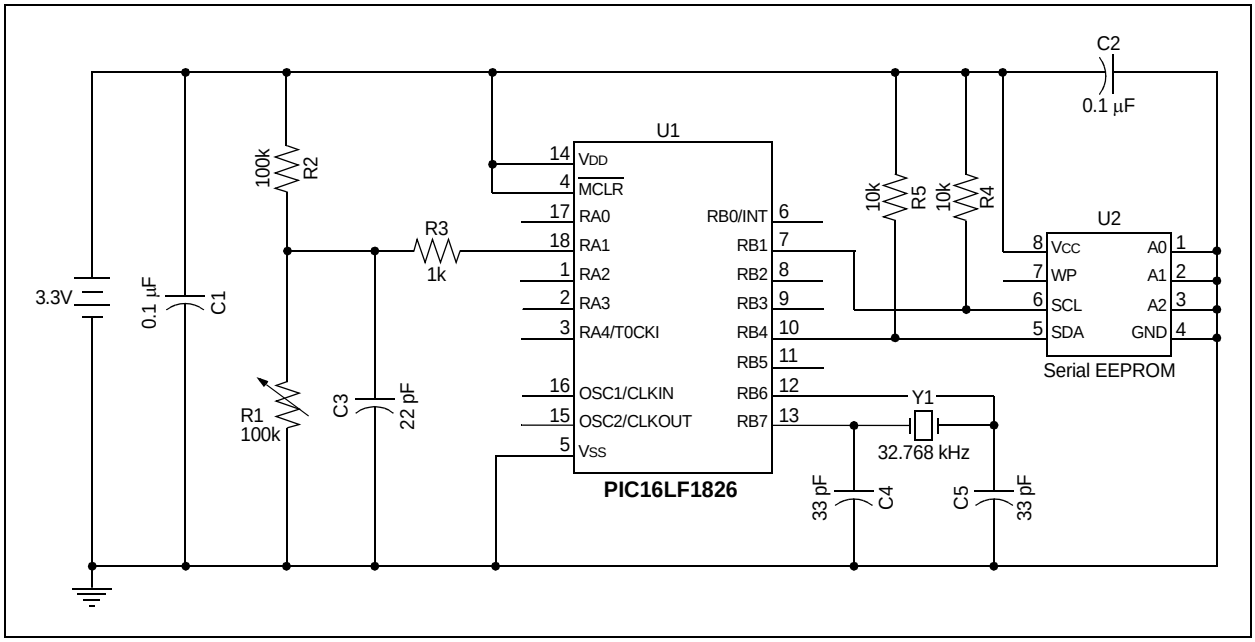
$$I\,(nA) = \frac{V \times C}{IR\,(M\Omega \times MF)}$$

Note that the initial leakage current for some capacitors can be higher as the chemicals in the dielectric are polarized. The current will start high and drop to the rated leakage value as the capacitor is held at full charge. When selecting capacitors for a low-power system, wherever possible, it is best to select capacitors with low leakage current.

Powering External Circuits

The MCU is only a single component in a complete system and all devices in the system will contribute to the system’s total power consumption. While many modern ICs are designed with Low-Power modes, other components have different standards for what low power means. While an MCU might have a Sleep mode with power consumption near 100 nA, often devices, such as external transceivers, consume a few microamps in Standby mode. As a result, it is often desirable to allow the MCU to control power to external circuits in a system in order to minimize total power consumption. Some external circuits, which an MCU can easily control that consume high power, are analog voltage dividers, LCD backlights, sensors and transceivers (e.g., RF and RS-232). Consider the system shown in Figure 9. By modifying the system so that the voltage divider (R1) and serial EEPROM (U2) are controlled by the MCU’s I/O (RA0 and RB0, respectively), the system leakage current can be substantially reduced. For high-power devices, which cannot be driven by the MCU’s I/O directly, use the PIC device to drive a MOSFET and power the circuit. Making these modifications causes the design to resemble the optimized low-power system from Figure 3.

FIGURE 9: NON-OPTIMIZED LOW-POWER SENSOR SYSTEM



There are a few concerns to be aware of when utilizing this technique. First, ensure that all I/O pins, connected to the disabled external circuits, are in the proper states before powering components on or off. Ordinarily, the best procedure for handling the I/O is the following:

- Powering on a device:
  1. Tri-state all connected I/O.
  2. Power on the external device and wait for the device to start up.
  3. Configure the I/O to the operational state.
- Powering off a device:
  1. Tri-state all connected I/O.
  2. Power off the external device.
  3. Configure the I/O to drive low as outputs.

The I/O pins are tri-stated, prior to changing the state of the device, to ensure that there are no bus conflicts while the system is powering up or down. When an external circuit is disabled, it is good practice to drive the connected I/O pins low as outputs to ensure that those lines are not allowed to float.

When using this technique, consider the cost of repowering a device versus the static power consumption. For an external device with high inrush current, large capacitors to charge, or long power-on times, there is a high-current consumption associated with power-on. The charge lost having to repower these components can cause more current consumption than is saved by powering it down. Make sure to perform a power budget on the full system to determine which state is lower power.

Power Supply Design

The selection and design of a power supply for a low-power system can have major impacts on the power consumption of the system. Linear regulators, switching regulators and other PMICs all require some amount of power to operate, and many are tuned for systems with high-power requirements. For battery-powered systems, it may not be feasible or desirable to use a regulator at all in the power supply.

Switching regulators usually provide greater efficiency for applications, operating at higher power levels. This makes them ideal for line powered, low-power systems, which focus on increasing efficiency and reducing dynamic power consumption. Even so, they tend to have very low efficiency at low supply currents. A boost regulator, that can be 95% efficient at 50 mA, is likely to have efficiency around 20% or lower at a load current of 10 µA. Keep this in mind when designing a low-power system, which may spend most of its time asleep, as it can result in a linear regulator being a more efficient choice overall.

In some cases, it can be ideal to run a battery-powered system without a regulator. In particular: LiFeS2 (Lithium AA Batteries), LiMnO2 (Coin Cells) and Alkaline batteries all operate within the 1.8-3.6V operating range of most low-power microcontrollers. Refer to the battery section of this document for more details on these battery chemistries. By designing a system capable of operating directly from the battery, significant gains in system lifetime are possible.

Consider a low-power system designed to run from two alkaline AA batteries. These batteries have a linear discharge curve and will discharge from 3.6V to 2.7V over the first 50% of the battery life. Using a low dropout 2.5V regulator, only about half of the battery capacity will be used before the regulator is below the minimum operating voltage. The charge remaining in the batteries from 2.7V down to 1.8V cannot be used, wasting half of the battery capacity. Additionally, there will be some amount of quiescent current consumption by the regulator during operation, further reducing the effective battery life. There are two major trade-offs of operating without a regulator. One is the loss of a consistent voltage reference for the application. However, most modern MCUs have an internal reference voltage that can be used in place of the regulator to determine the current V_{DD} voltage and perform analog conversions. The other trade-off is that all of the components of the application will need to support a wider voltage range, which can increase costs.

Low-Power Crystal Design

One of the major impacts of the attempts to reduce MCU power consumption has been in the drivers for low-power 32.768 kHz crystals. In order to minimize current consumption, the drive strengths for a typical crystal driver have become very low, so much so, that now a crystal can effectively be run at current levels below 1 μA. The result of this, is that the crystal circuit design must be done very carefully, and be well tested to ensure that it will start up correctly and run accurately.

The basics of low-power crystal design are to place the crystal as close as possible to the MCU and if possible, surround the crystal driver circuit with a ground ring to prevent coupled noise.

Additionally, it is very important to ensure that the tank capacitors are properly sized for both the crystal and the driver. Note that this is a major change from many older crystal designs. The average load capacitance for a 32 kHz crystal is typically 12.5 pF, which results in tank capacitors of about 22 pF. However, as the crystal driver currents have dropped, they may no longer effectively start a crystal with this high of a load capacitance. Low-power crystals with ideal load capacitances, from 3.7 pF to 6 pF, should be used in these systems to get more reliable operation at low power.

For a more complete analysis of how to design low-power crystal driver circuits, refer to *AN1288, "Design Practices for Low-Power External Oscillators", AN849, "Basic PICmicro® Oscillator Design"* and *AN949, "Making Your Oscillator Work"*.

BATTERY CONSIDERATIONS

A large number of low-power systems are mobile or remote devices running from battery power. The following sections will provide an introduction to designing a battery-based application, focusing on some of the key design decisions when using a battery.

Battery Chemistries

Battery chemistry has a major impact on the specifications and performance of a battery. Each chemistry has a specific voltage discharge profile, internal impedance and self-discharge current. Some of the most common primary battery (non-rechargeable) chemistries are Alkaline (standard AA/AAA), lithium-manganese dioxide (coin cells) and lithium-iron disulfide (lithium AA/AAA). The most common rechargeable chemistries are nickel-metal hydride (rechargeable AA/AAA) and Lithium Ion. Table 6 shows a summary of the characteristics of the various battery chemistries.

The voltage profile describes how the output voltage of the battery changes as it is discharged. Most batteries will provide graphs in the data sheet, showing the voltage profile at specific discharge currents. The voltage profile affects the regulation and operating voltage requirements of a system. For example, alkaline batteries have a sloping discharge curve which will drop linearly as the battery is used. Applications which require a steady voltage will need a regulator to ensure a consistent operating voltage. By contrast, most lithium chemistries have very flat discharge curves. They will output a constant voltage until the end of the battery life when the voltage output will rapidly drop off. The flat discharge curve lithium chemistries are harder to fuel gauge since it is no longer feasible to measure battery voltage to determine remaining charge. However, a system will be able to use a higher percentage of the energy in the battery, before the voltage drops too low, compared with alkaline batteries.

The internal impedance of a battery is a result of the chemical nature of batteries. Because they require a chemical reaction to release energy, batteries cannot react instantaneously to the electrical needs of a system. As a result, a short pulse of high current on a battery will often result in a sharp drop in the output voltage, which a designer must consider when creating a battery-based system. This is particularly true of lithium coin cell batteries, which often have current limits around 10 mA, due to the high internal resistance. Operating above this limit will waste a large amount of the capacity of the battery as voltage drops across the internal resistance.

Self-discharge current is caused by the ongoing chemical reaction inside the battery. Even when no current is being drawn from a battery, the internal chemical reaction is taking place. This means that some of the battery capacity is lost over time. Alkaline batteries and most secondary batteries have high self-discharge currents, which can make them unsuitable for very long lifetime applications. Lithium primary batteries have very low self-discharge, typically less than 1% of battery life per year. Battery manufacturers usually specify self-discharge as a component of shelf life, indicating the number of years a battery can remain unused and retain most of its capacity (typically 80%).

TABLE 6: ELECTRICAL CHARACTERISTICS OF COMMON BATTERY CHEMISTRIES

Chemistry	Type	Typical Form Factor	Nominal Voltage	Voltage Profile	Self-Discharge (%/mo)	Nominal Internal Resistance
Alkaline	Primary	AA/AAA	1.5V	Sloped	0.08%	150-300 mΩ
Li/MnO_2	Primary	Coin Cell	3.0V	Flat	0.05%	10k-40k mΩ
Li/FeS_2	Primary	AA/AAA	1.5V	Flat	0.30%	90-150 mΩ
Lithium-ion	Secondary	Varies	3.6V	Flat	20%	30-40 mΩ
Ni/MH	Secondary	AA/AAA	1.2V	Sloped	30%	30-40 mΩ

Estimating Battery Life

Using the average current from the calculated power budget, it is possible to determine how long a battery will be able to power the application. Table 7 shows life-times for typical battery types using the average power consumption of the system, described in the power budgeting example from Table 4. The equation for calculating battery life is given in Equation 4:

EQUATION 4: BATTERY OPERATING LIFETIME CALCULATIONS

$$Life\ (hours) = \frac{Capacity\ (mAh)}{System\ Current + Battery\ Self\text{-}Discharge\ Current\ (mA)}$$

TABLE 7: BATTERY LIFE ESTIMATIONS FOR LOW-POWER SENSOR

Battery	Capacity (mAh)	Self-Discharge (%/yr)	Battery Life (years)
CR2032	220	1%	8.3
Lithium AAA	1150	1%	36.7(1)
Alkaline AAA	1150	4%	17.5(1)
Li-ion(2)	850	20%	4.4

Note 1: This time exceeds the typical shelf life of the battery, indicating the battery may physically fail before fully discharging.

2: Capacity varies by size; value used is typical.

Note that Equation 4 is a simple tool meant for generating an initial estimation. Reviewing the results of the equation in Table 4 for the AAA sized batteries shows that it predicts a lifetime that exceeds the typical shelf lives for these battery types. It is possible that the battery will physically fail via other means, such as the packaging leaking, before it is fully discharged. A designer should be aware of all of the limitations of a battery when calculating its design lifetime.

In order to simplify the process of power budgeting and battery life estimation, Microchip provides a tool, the XLP Battery Life Estimator, which can be used to get a basic calculation of battery life based on the battery being used and the system power profile. This tool supports most nanoWatt XLP PIC MCU devices and can be configured for a wide range of battery types. The tool can be downloaded from www.microchip.com/ble.

CONCLUSION

Low-power design is a complex topic, far beyond what can be covered in a single document. This application note provides an introduction to many of the topics which a low-power system designer would encounter during the design process. A designer must be aware of the sources of power consumption in a system, key power related specifications of components, hardware and software power optimization techniques, and the relationship between power and performance. With knowledge of the low-power design aspects, which are critical for their applications, designers can focus on diving deeper into these topics in order to create a truly power optimized system.

REFERENCES

Microchip Low Power Design Center, www.microchip.com/xlp

Energizer Product Data Sheets, Energizer Inc., http://data.energizer.com/.

B. Ivey, *AN1267*, *"nanoWatt and nanoWatt XLP Technologies: An Introduction to Microchip's Low-Power Devices"* (DS01267), Microchip Technology Inc. 2009.

Microchip Applications Staff, *"Combined Tips 'n Tricks Guide"* (DS01146), Microchip Technology Inc., 2009

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-61341-832-1

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/02/11

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Power Management IC Development Tools ***category:***

Click to view products by Microchip ***manufacturer:***

Other Similar products are found below :

EVAL6482H-DISC EVAL-AD5522EBUZ EVAL-ADM1060EBZ EVAL-ADM1073MEBZ EVAL-ADM1166TQEBZ EVAL-ADM1168LQEBZ EVAL-ADM1171EBZ EVAL-ADM1276EBZ EVB-EN5319QI EVB-EN5365QI EVB-EN6347QI EVB-EP5348UI MIC23158YML EV MIC23451-AAAYFL EV MIC5281YMME EV 124352-HMC860LP3E ADM00513 ADM8611-EVALZ ADM8612-EVALZ ADM8613-EVALZ ADM8615-EVALZ ADP1046ADC1-EVALZ ADP1055-EVALZ ADP122-3.3-EVALZ ADP130-0.8-EVALZ ADP130-1.2-EVALZ ADP130-1.5-EVALZ ADP130-1.8-EVALZ ADP160UJZ-REDYKIT ADP166UJ-EVALZ ADP1712-3.3-EVALZ ADP1714-3.3-EVALZ ADP1715-3.3-EVALZ ADP1716-2.5-EVALZ ADP1740-1.5-EVALZ ADP1752-1.5-EVALZ ADP1754-1.5-EVALZ ADP1828LC-EVALZ ADP1870-0.3-EVALZ ADP1871-0.6-EVALZ ADP1873-0.6-EVALZ ADP1874-0.3-EVALZ ADP1876-EVALZ ADP1879-1.0-EVALZ ADP1882-1.0-EVALZ ADP1883-0.6-EVALZ ADP197CB-EVALZ ADP199CB-EVALZ ADP2102-1.25-EVALZ ADP2102-1.2-EVALZ