Efficient Metal Detection and Inductive Sensor Monitoring in sub-µA Deep Sleep Mode

Introduction

With the Low Energy Sensor Interface (LESENSE), the EFM32 microcontrollers are able to monitor a wide variety of sensors while continuously staying in energy saving deep sleep mode. Applications in need of wake on capacitive touch, sensing metallic objects or monitoring resistive sensors can be implemented with total consumption below 1.2 µA.

The sensors that LESENSE can interface range from simple thermistors, inductive or capacitive devices, to more complex analog sensors. In many applications such sensors are used to give input from the outside world to a microcontroller. As energy consumption and battery life become increasingly important, the microcontroller needs to be able to monitor these sensors by using as little energy as possible. The best way of doing this is staying in deep sleep as much as possible. This document will present three different measurement principles, capacitive, inductive and resistive, that can be implemented with the low energy sensor interface. References to the different application notes are also included.

The Challenge

Staying asleep for as high percentage of the time as possible and do the necessary measurements quickly and effectively, without energizing unnecessary circuitry, is the main challenge for a low power sensor application. Until now applications typically need to wake up the CPU to do sensor measurements (Figure 1). For a small microcontroller based system the CPU, when active normally consumes a lot of energy compared to the rest of the system. This means that reducing the amount of time the CPU is active, is one of the best ways to decrease energy consumption. Still, sensor measurements often require special excitation and sampling patterns. Wake up conditions may be complex and the required sampling rate might be high.

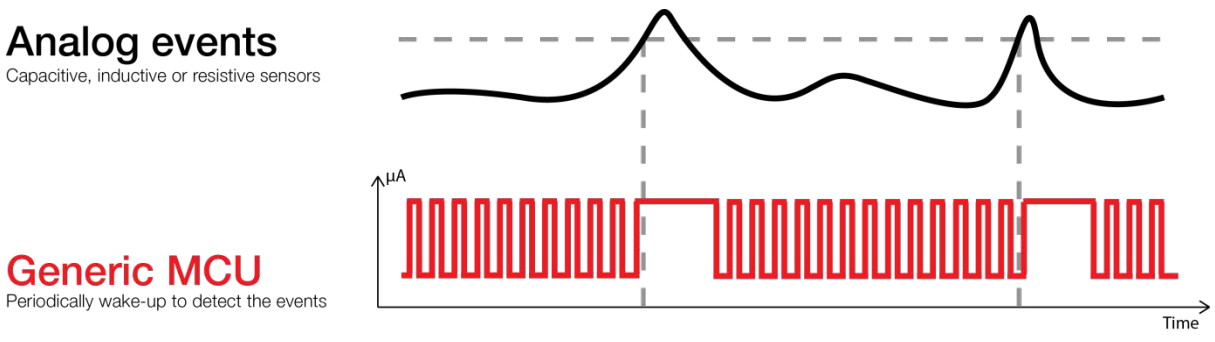

Figure 1: High energy consumption with CPU active during every measurement.

Solution

A sensor measurement system that can operate autonomously even in deep sleep modes is the ideal solution. The autonomous system must be configurable for different external components and sensors. Different measurement principles should be easy to accommodate, and the system should incorporate energy friendly calibration features. How can all of this be done autonomously, saving precious energy for more complex tasks that really needs the energy hungry CPU? The answer: LESENSE.

LESENSE achieves this with a highly configurable sequencer and decoder that can also incorporate several of the ultra energy efficient peripherals in the EFM32. Efficient duty cycling of these resources means that a minimum of energy is used to monitor a sensor, (Figure 2). Further, the devices can be set to wake up only if certain conditions are met. This can be as simple as a finger touching a capacitive button, or it can be more complex transitions in the decoder, filtering sensor values before a wake up signal is issued. Either way, the Low Energy Sensor Interface will give the microcontroller a sense of the outside world without leaving its deepest beauty sleep.

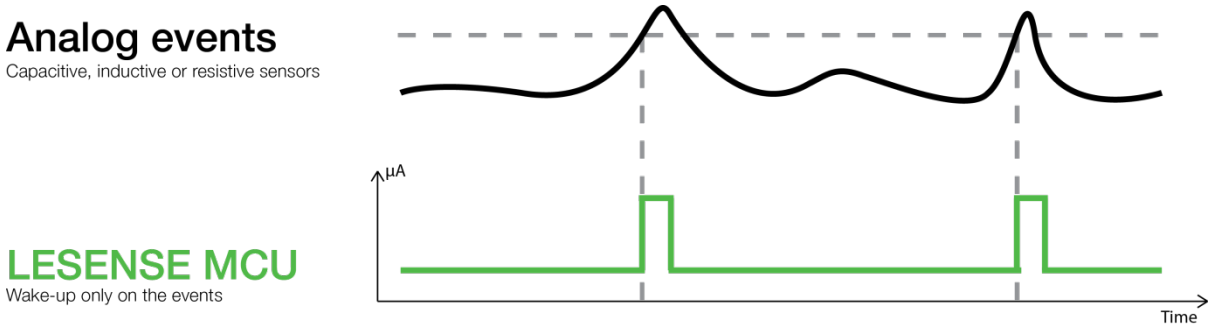

Figure 2: Reduced energy consumption with LESENSE, waking up the CPU only when necessary.

More specifically the Low Energy Sensor Interface consists of analog comparators and a DAC (Digital to Analog Converter) which can be controlled by a very low power sequencer-module. The sequencer is able to operate in sub-µA modes running from a 32 kHz clock source. In addition to the sequencer, the comparator outputs can be counted, compared or passed on directly as interrupts. For accurate measurements, the digital-to-analog converter (DAC) can be selected as comparator reference and excitation source.

The sequencer controls which pins are connected to the comparator, how long the comparator is active and when the comparator output should be passed on to be counted or compared. Excitation with a DAC voltage or GPIO-pins can also be executed before or while the comparator is active. After a measurement, the counter or comparator output is buffered and stored for later processing. Interrupts or signals can also be passed on directly (through the peripheral reflex system) when the counter or output crosses pre-configured thresholds.

After a scan is finished, the results can be passed on to the low power decoder which is part of LESENSE. Each state in the decoder has configurable next states and trigger conditions. This makes it

possible to interpret several sensor-readings and only wake up the CPU if they match a pattern over time (Figure 3).

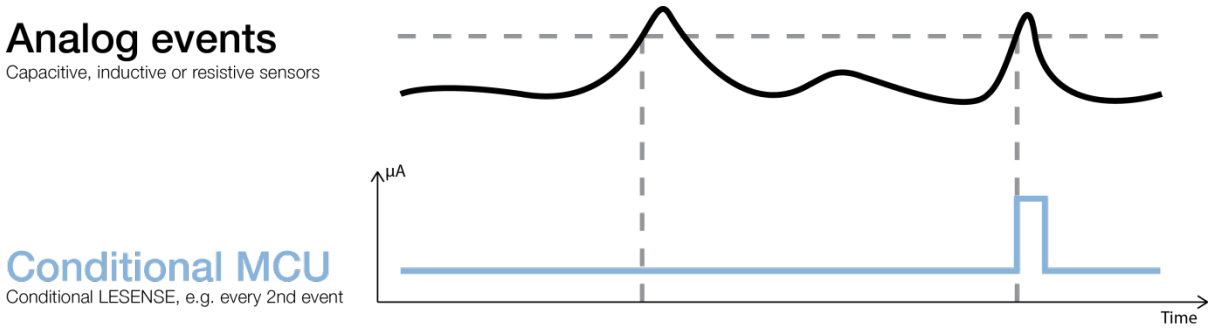

Figure 3: Conditional wake up on analog events.

Since the sensor results are available through the PRS (Peripheral Reflex System), they can be passed on to other low energy peripherals like serial ports or counters. The pulse counter, for instance, is able to take in PRS-signals and count them in quadrature counter mode. This enables the system to only wake up after a certain number of rotations. Think of a water-meter with a rotating vane, the rotations can be measured with LESENSE and counted with the quadrature counter. After, for example 10 rotations the CPU can be woken up to update the display and usage statistics. With a traditional MCU, the CPU would be needed for all the sequencing and control of the comparator. With the EFM32 all of this can be handled by LESENSE, allowing the chip to stay in deep sleep.

Since many applications must handle varying environmental conditions, continuous or periodic calibration of sensors-thresholds and measurement-timing is often critical for the performance. Since LESENSE stores several results in memory, the CPU does not need to wake up all the time to capture measurement results. These results can be used for periodic calibration and updates of trigger thresholds and measurement-timing.

Inductive Sensor Example

In many applications it is more convenient to use inductive sensing instead of capacitive sensing for proximity detection. Inductive sensing typically relies on the sensed object being made of metal. One measurement principle that can be used is the shorter decay-period of oscillations in a coil when metal is nearby the coil. Oscillations in an inductor of coil will generate a varying magnetic field around the inductor. Eddy currents will be induced in metal if it is placed in this magnetic field. This will in turn dissipate the energy in the oscillating system quicker. To make this oscillating system, all that is needed is a coil and a capacitor in parallel.

LESENSE can be used to excite and then measure the decay period or damping-factor in this type of tank-circuit or LC-circuit. To measure the damping factor, the analog comparator is configured to generate a high output each time the sensor voltage exceeds a certain level. These pulses are counted using the asynchronous counter within LESENSE and then the counter value is compared with a threshold value. If the number of pulses does not exceed the threshold level, the sensor is

said to be active, otherwise it is inactive. Figure 4 illustrates how the output pulses from the comparator correspond to the damping of the oscillations.

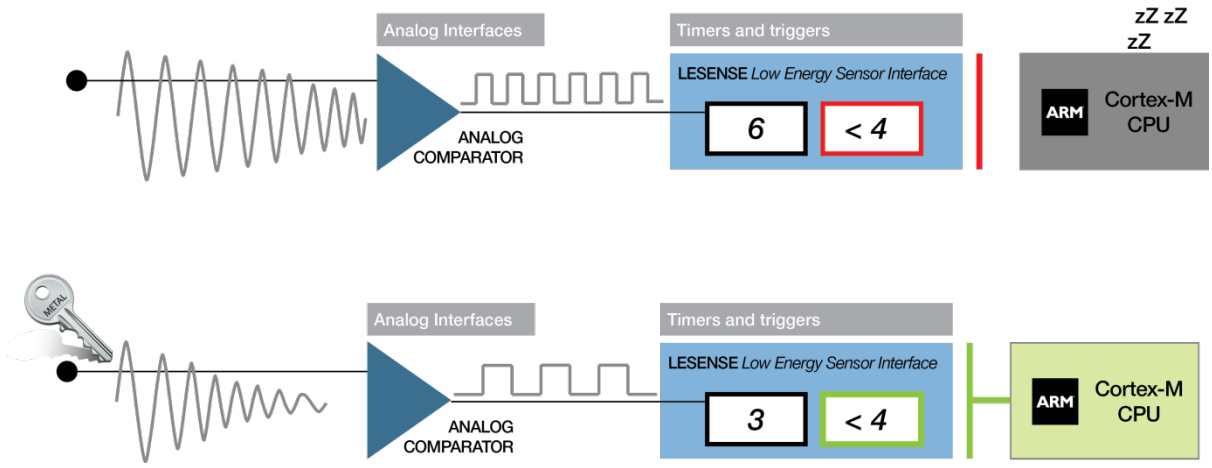

Figure 4: Inductive sensor

Current consumption

The current consumption of the inductive sensing application is mainly influenced by sampling frequency and the inductance of the coil. The larger inductance the coil has, the more energy is needed to excite it. All this energy needs to dissipate before a new sampling can begin.

Empirical results have shown that a 390uH coil which fit in a 2 by 2 mm footprint can easily detect metal 5-6 millimeters away. This setup, sampled at 20 Hz, only adds a 200 nA of current consumption in the deep sleep EM2 mode, resulting in a total consumption of 1.2 µA. This number includes the total EM2 current consumption of the EFM32, LESENSE operation and RTC running from the 32 kHz oscillator.

Conditional Wakeup

Since LESENSE also includes a fully functional and configurable decoder, many different conditional wake-up triggers can be implemented. The decoder can be utilized to only wake up from deep sleep when several sensor conditions are met at once. It can also track consecutive measurements to issue a wake up only after a special sequence of threshold crossings has occurred. This can for instance be that a temperature and humidity sensor both has reached their thresholds, or a pressure sensor must trigger 10 times in a row before a wake up event occurs.

Since the inputs to the decoder can be connected to the peripheral reflex system, almost any peripheral can interact with it. This makes it possible to logically combine several GPIO-pins to trigger a wake up, or use the decoder to decode serially transmitted data. The possibilities with a fully configurable decoder are truly end-less.

Application Example: Counting Rotations

By connecting LESENSE to other low energy peripherals it can keep track of a rotating wheel with very low power consumption. The results from the sensor evaluation can automatically be fed into the decoder or to a quadrature counter through the PRS in order to do conditional wake up or keep track of rotations. This is illustrated in Figure 5. Two coils are placed close to a rotating wheel where half of it is covered with metal. LESENSE is sampling each coil fast enough to catch the passing metal-part of the wheel. The output from each sampling is fed to the quadrature counter through the PRS system. If the counter reaches for example 5 rotations in the same direction it issues an interrupt.

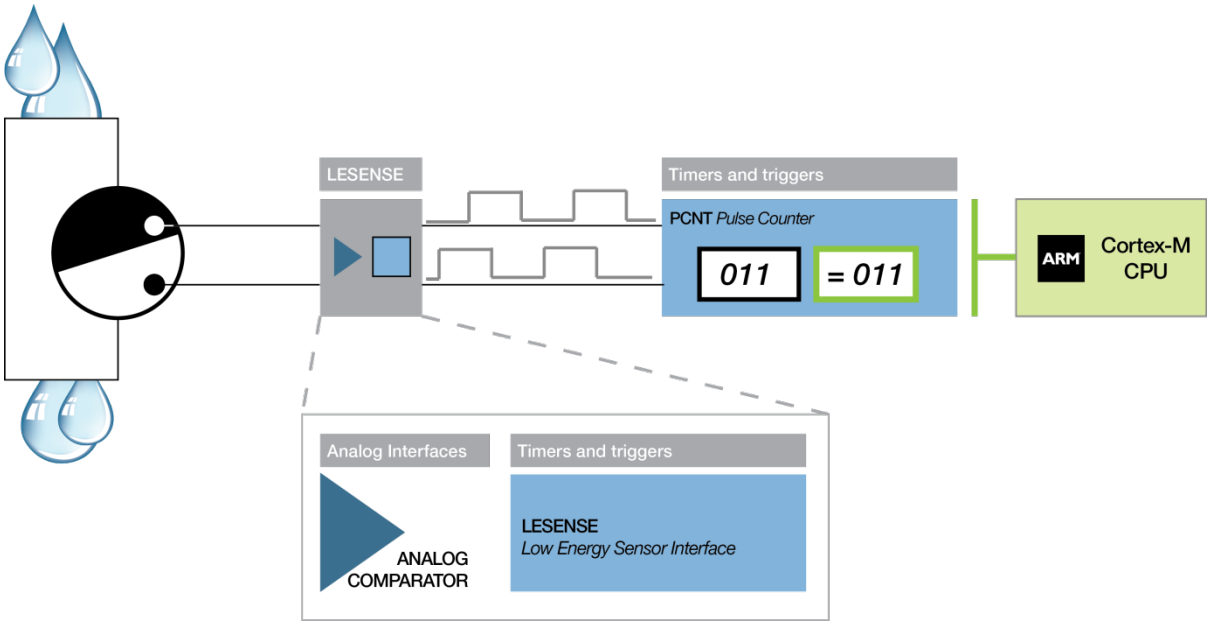

Figure 5: Counting rotations, utilizing both LESENSE and the Pulse Counter module.

Summary

The Low Energy Sensor Interface (LESENSE) enables the EFM32 microcontroller to monitor many different kinds of analog sensors while staying in deep sleep mode. Running from the low frequency clock source, LESENSE can monitor up to 16 sensors in sub-µA sleep modes. Typical average current consumption is around 1.2 µA. Applications include any kind of capacitive, inductive or resistive sensing, rotation counting, GPIO-state decoding or similar. LESENSE also has a fully configurable decoder to decode sensor outputs. The decoder can evaluate sensor states and wake up the CPU when a special combination of sensor outputs or patterns over time occurs. By combining the sensor interface with the built in decoder, the possibilities are only limited by the imagination of the designer.

For further information about LESENSE, please refer to the following application notes:

- an0028 Low Energy Sensor Interface - Capacitive Sense
- an0029 Low Energy Sensor Interface - Inductive Sense
- an0036 Low Energy Sensor Interface - Resistive Sense
- an0040 Hardware Design for Capacitive Touch
- an0053 IR Sensor Monitoring Using LESENSE