

디자인 코너

마이크로프로세서 디자인에서의 A/D 컨버터 선택 요령

마이크로프로세서에 아날로그 값을 입력시키는 일반적인 방법은 ADC(analog-to-digital converter)를 사용하는 것이다. 그 같은 부품을 선택하고 이를 자신의 필요에 맞게 보정하는 방법 몇 가지를 살펴 보자.

그림 1은 간단한 전압 입력 ADC이다. 이 가상의 부품에는 기준 신호(reference)와 측정 대상 신호라는 두 개의 입력이 있다. 출력은 하나로서 입력 값을 나타내는 8 비트 디지털 워드이다.

기준 전압은 ADC가 변환시킬 수 있는 최대 값이다. 예로 든 8 비트 ADC는 0 V에서 기준 전압까지 값들을 변환시킬 수 있다. 이러한 전압 범위는 256 단계 또는 값들로 나누어진다. 따라서 각 단계의 크기는 다음과 같다:

$$V_{ref}/256$$

여기서 V_{ref} 는 기준 전압을 뜻한다. 컨버터의 해상도는 그 단계 크기에 의해 정해진다. 예컨대 기준 전압이 5 V 일 경우, 단계의 크기는 다음과 같다:

$$5V/256 = 0.0195 \text{ V, 즉 } 19.5 \text{ mV}$$

이 8 비트 컨버터는 아날로그 입력을 디지털 워드로 나타낸다. 이 워드의 최상위 비트(MSB)는 입력 전압이 기준 전압의 절반(즉, 기준이 5 V일 경우, 2.5 V)보다 큰지의 여부를 알려준다. 이어지는 각 비트는 이전 비트의 범위 중 절반을 나타낸다.

표에서 볼 수 있는 것은 바로 이런 점이다. 0010 1100에 설정된 각 비트에 대응하는 전압들을 추가 시키면 다음과 같은 결과를 얻게 된다:

$$.625 + .156 + .078 = .859 \text{ V}$$

ADC의 해상도는 기준 입력과 워드 폭에 의해 결정된다. 해상도는 ADC가 측정할 수 있는 최소 전압 변화를 정의한다.

이제까지 사용해온 예에서 해상도는 19.5 mV이다. 이는 입력 전압이 19.5 mV 이하일 경우 출력 값은 0 이 됨을 뜻한다.

기준 입력 값을 줄이면 해상도를 향상시킬 수 있다. 즉, 기준 입력 값을 5 V에서 2.5 V로 바꾸면 해상도는 $2.5/256 = 9.7 \text{ mV}$ 가 된다. 그 대신 측정 가능한 최대 전압은 5 V가 아니라 2.5 V가 된다.

측정 범위를 줄이지 않고도 해상도를 올리는 길은 비트 수가 더 많은 ADC를 사용하는 것 뿐이다.

ADC의 비교

ADC의 속도는 다양하며 사용하는 인터페이스와 정확도 수준도 저마다 다르다. 가장 일반적인 유형의 ADC로는 플래시(flash)형과 축차근사(successive approximation)형, 그리고 시그마-델타(sigma-delta)형이 있다.

형이 있다.

8~16 비트 범위의 축차근사형 ADC를 손에 넣을 수는 있지만 16 비트 버전은 주어진 계열 제품들 가운데 가장 빠르지는 못하다. 가장 빠른 플래시 ADC는 12 비트 부품이 아니라 6 비트나 8 비트 부품이다.

CMOS 공정의 개선은 모든 컨버터 제품군의 속도를 향상시켜 주지만, 갈수록 더 정교한 DSP 기능을 ADC 칩에 집어넣는다 해도 축차 근사형 컨버터의 성능을 개선시키지는 못한다. 그러나 시그마-델타형 컨버터는 개선되는데, 보다 우수하고 빠르게 복잡한 필터들을 추가시킬 수 있게 되기 때문이다.

샘플 앤 홀드

ADC의 동작은 DC 신호의 변환시에는 간단하다. 그러나 입력 신호가 변환시에 하나의 최하위 비트(LSB) 이상 변경될 경우, ADC는 잘못된(또는 최소한 부정확한) 결과를 생성하게 된다. 이러한 오류를 줄이는 방법 중 하나는 ADC에 앞서 저역 필터를 배치하는 것이다. ADC 입력이 변환 주기 내에 하나의 최하위 비트 이상 변경되지 않도록 필터 패러미터들이 선택된다.

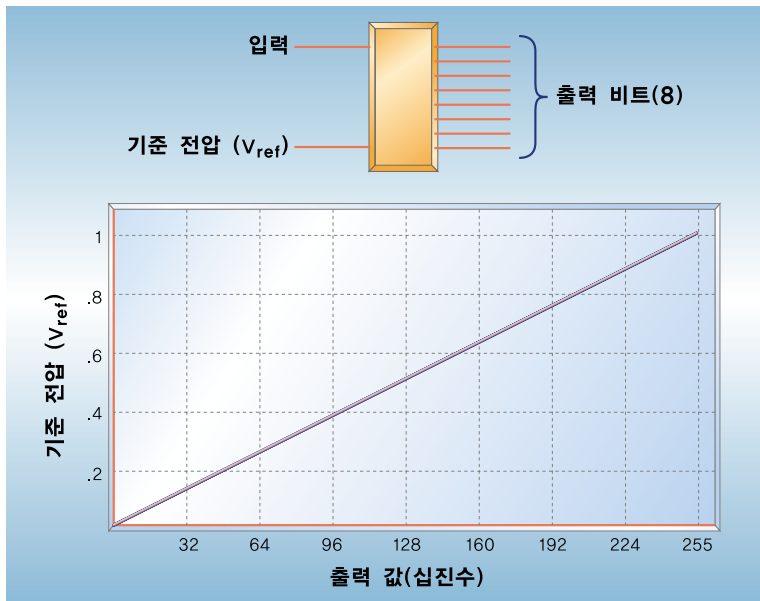


그림 1: 가상 부품의 입력과 출력을 보여주는 간단한 ADC

입력의 변화 문제를 해결하기 위한 또 다른 방법은 ADC 앞에 S/H(sample-and-hold) 회로를 추가하는 것이다. S/H 회로에는 제어 입력 기능을 갖춘 아날로그(반도체) 스위치가 있다. 이 스위치가 닫히면 입력 신호가 홀드 콘덴서에 연결되며, 버퍼 출력은 입력을 따르게 된다. 스위치가 열리면 입력은 콘덴서와의 연결이 끊어지게 된다.

이상적인 경우라면 홀드 콘덴서에 누설 현상이 없고 버퍼 증폭기의 입력 임피던스는 무한대가 되어 출력이 영원히 안정 상태를 유지해야 할 것이다. 하지만 실제로는 홀드 콘덴서에 누설 현상이 발생하고 버퍼 증폭기의 입력 임피던스는 한정적이므로 출력 수준은 콘덴서가 방전됨에 따라 그라운드 수준을 향해 천천히 내려가게 된다.

S/H 회로가 출력을 홀드 모드로 유지하는 성능은 홀드 콘덴서의 품질과 버퍼 증폭기의 특성(주로 입력 임피던스), 그리고 샘플/홀드 스위치의 품질(실제의 전자 스위치들은 개방시 약간의 누설을 일으킨다)에 의해 좌우된다.

홀드 모드에서 출력이 나타내는 드리프트 양을 수하율(droop rate)이라고 하며, 이는 초당 밀리볼트, 마이크로초당 밀리볼트, 또는 마이크로초당 마이크로볼트로 표시한다.

실제의 S/H 회로는 입력 임피던스도 한정되어 있는데, 이는 전자 스위치가 완벽하지 않기 때문이다. 이는 샘플 모드에서 홀드 콘덴서가 모종의 저항을 통해 충전되고 있음을 의미한다. 이 때문에 S/H 회로가 입력을 받아 들일 수 있는 속도가 제한된다. S/H가 전체 입력을 받아 들이기 위해 샘플 모드로 남아 있어야만 하는 시간을 획득 시간(acquisition time)이라고 하며, 나노초나 마이크로초 단위로 지정된다.

앞서도 언급했듯이, 전자 스위치는 불완전하며 홀드 모드에서조차도 입력 신호의 일부가 출력부에 나타난다. 이를 피드스루(feedthrough)라고 하며, 대개 데시벨 단위로 지정된다. 출력 오프셋은 입력과 출력 간의 전압차이다.

S/H 회로의 데이터시트에는 홀드 모드 오프셋과 샘플 모드 오프셋이 대개 밀리볼트 단위로 표시되어 있다.

소프트웨어

S/H를 사용하는 ADC 시스템은 하드웨어의 급격한 변화를 수용해야만 할 것이다. 어떤 시스템에서는 소프트웨어가

포트나 레지스터 출력 비트를 이용해 S/H 제어 입력부를 직접 제어한다.

통상적으로 S/H는 샘플 모드에 놓이게 되며, 소프트웨어는 획득 시간 요건이 반드시 충족되도록 해줘야만 한다. 어떤 시스템에서는 S/H를 변환이 필요할 때까지 샘플 모드에 그대로 놔둬으로써 이 같은 일

비트	비트 7	비트 6	비트 5	비트 4	비트 3	비트 2	비트 1	비트 0
볼트	2.5	1.25	.625	.3125	.156	.078	.039	.0195
출력값	0	0	1	0	1	1	0	0

표 : 아날로그 입력을 디지털 워드로 나타내 주는 8 비트 ADC 상의 변환 예

을 간단하게 해낼 수도 있다.

S/H 회로가 홀드 모드로 놓인 후에는 또 다른 비트(나 임의의 주소에 대한 기록 작업 또는 다른 어떤 동작)에 의해 ADC가 가동된다. 변환이 완료된 후에는 소프트웨어가 결과를 판독한다.

그러나 임의의 한 인터럽트(또는 인터럽트가 쌓이는 최악의 경우)로 인해 S/H 회로의 출력이 최하위 비트 하나 이상 수하될 경우에는 문제가 발생할 수도 있다. 이러한 일이 발생할 수 있을 경우 소프트웨어는 S/H를 홀드 모드로 전환하기 전에 인터럽트를 디스에이블시킨 뒤 변환에 들어간 후에 이들을 다시 인에이블시켜야 할 수도 있다. 이렇게 하면 ADC는 S/H 수하 현상이 발생하기 전에 변환 작업을 완료할 수 있다.

소프트웨어는 S/H의 충전 시간도 수용해야만 한다. 전자 스위치가 닫혀 입력 신호를 S/H 콘덴서에 연결시킬 때, 콘덴서가 충전되기까지는 한정된 시간이 소요된다. 스위치를 비롯하여 입력을 구동시키는 모든 소스의 임피던스 값은 영(0)이 아니기 때문이다. 이러한 임피던스 값들의 총합이 충분히 클 경우, 소프트웨어는 변환 작업에 들어가기 전에 홀드 콘덴서가 최종 값의 1 LSB 이내로 충전할 시간을 가질 수 있도록 지연을 추가해야 할 수도 있다.

마이크로컨트롤러 내장 ADC

많은 마이크로컨트롤러들이 칩 내장 ADC를 포함하고 있다. 그 대표적인 소자들이 Microchip PIC167C7xx 계열 제품과 Atmel AT90S4434이다. 대부분의 마이크로컨트롤러 ADC들은 축차 근사형인데, 이는 이 방식이 마이크로컨트롤러 다이 상에서 속도와 면적 비용 간에 최상의 절충을 제공하기 때문이다.

Microchip 계열 제품 같은 일부 마이크로컨트롤러들은 입력 핀 하나를 기준 전압으로 사용할 수 있도록 해준다. 이는 대개 일종의 정밀 기준과 관련되어 있다. 변환이 이루어진 뒤에 A/D 컨버터에서 읽을 수 있는 값은 다음과 같다:

$$(V_{in}/V_{ref}) \times 256$$

어떤 마이크로컨트롤러들은 공급 전압을 기준으로 사용한다. 5V 시스템에서 이는 V_{ref} 가 항상 5 V임을 뜻한다. 따라서 3.2 V의 신호를 8 비트 ADC로 측정하면 다음과 같은 결과를 얻게 된다:

$$\begin{aligned} & (V_{in} \times 256)/V_{ref} \\ &= (3.2v \times 256)/5V \\ &= 16310 \\ &= A316 \end{aligned}$$

하지만 이같은 결과는 5 V 서플라이의 값에 따라 달라진다. 예컨대 서플라이 전압이 1 퍼센트 더 높아지면 결과 값은 5.05 V가 된다. 이때 A/D 변환 값은 다음과 같이 된다:

$$(3.2V \times 256) / 5.05V = 1621 = A216$$

따라서 서플라이 전압이 1 퍼센트가 변하면 변환 결과는 1 카운트(count) 변화된다. 통상적인 파워 서플라이들의 값은 2 퍼센트나 3 퍼센트 변화할 수 있으므로 파워서플라이 값의 변화는 결과에 상당한 영향을 미칠 수 있다. 파워 서플라이의 출력은 장하, 온도, AC 입력 값의 변화, 그리고 서플라이별로 빈번히 달라질 수 있다.

이는 모든 ADC 디자인에 영향을 미치는 기준의 정확성 문제를 대두시킨다. 전형적인 ADC 기준의 공칭 값은 2.5 V이지만, 2.47~2.53 V(이 값들은 실제 부품의 데이터 시트에서 가져온 것이다) 범위에서 변화할 수 있다. 이것이 10 비트 ADC라면 2 V의 입력을 기준 범위의 양 극단에서 변환시킬 경우 다음과 같은 결과를 얻게 된다:

$V_{ref} = 2.47$ V일 때,

$$\text{결과} = (2V \times 1,024) / 2.47 = 82910$$

$V_{ref} = 2.53$ V일 때,

$$\text{결과} = (2V \times 1,024) / 2.53 = 80910$$

부품에 따른 기준 전압의 변화는 20 카운트의 출력 변화를 가져올 수 있다. 오류의 비율은 전체 범위에 걸쳐 동일 수준을 유지하지만, 수치적 오류는 물론 ADC 값이 큰 쪽이 더 크다.

소프트웨어 보정

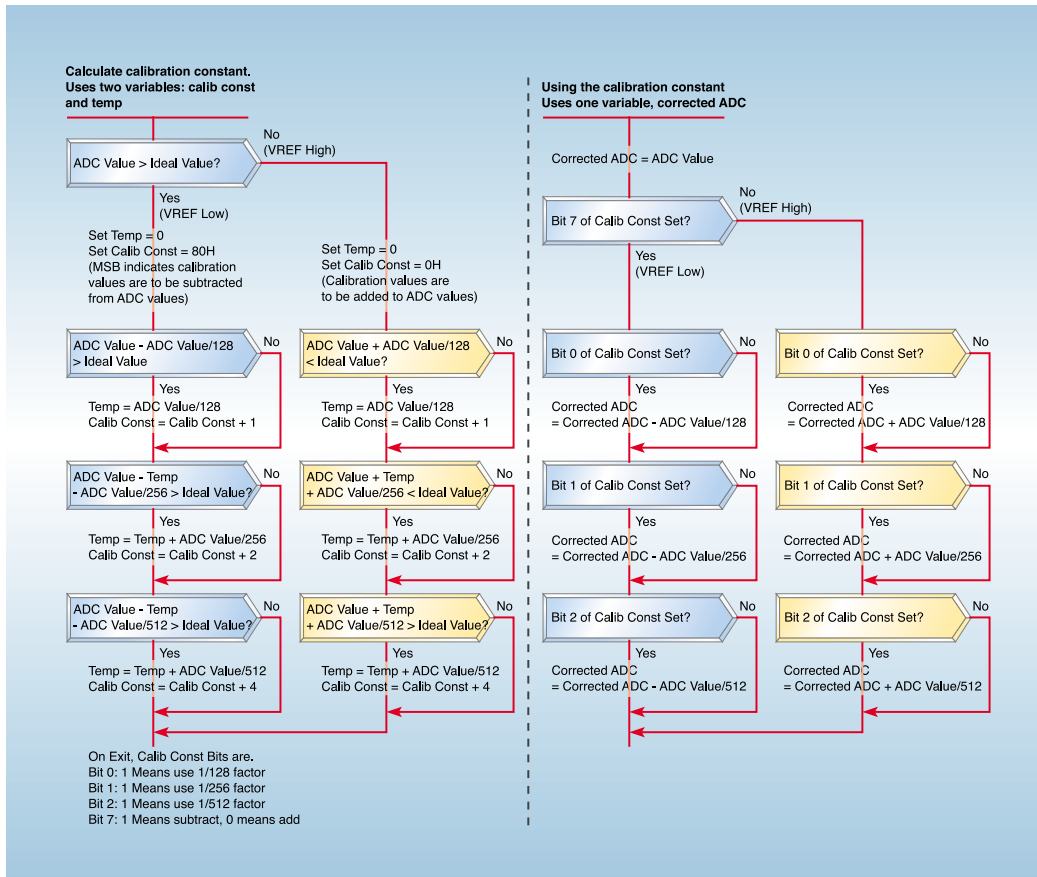


그림 2: 보정 상수를 계산 및 사용하기 위해 사용한 알고리즘을 보여주는 흐름도

때로는 정확한 기준 값이 필요한데, 이 값이 제품 비용으로 가능한 것보다 더 정확해야 할 때가 있다. 그러나 수작업으로는 조정할 수 없을 경우, 소프트웨어로 기준 전압의 변화를 보정할 수 있다. 이는 대개 이미 알고 있는 정확한 입력 값을 제공함으로써 이루어 지는데, 이 값이 ADC의 보정에 사용된다. 이 기준 값은 매우 정확하다(따라서 매우 비용이 많이 든다). 생산 라인에는 필요한 것은 불과 몇 개 뿐이기 때문이다.

우리가 살펴본 2.47 V의 예에서는 정확히 2 V의 전압이 ADC에 대한 입력 값이 될 수 있다. 소프트웨어는 ADC를 판독할 때 정확한 값이 819여야 함을 알고 있다. 따라서 보정 상수는 829/819, 즉 1.012가 된다.

이와 유사하게, 기준 전압 2.53 V에 대한 보정 상수는 809/819, 즉 0.988이다.

이는 ADC 값을 보정하기 위한 부동소수점 연산이 필요함을 의미하는 것 같다. 물론 부동소수점 연산 기능을 갖춘 프로세서를 사용하고 있다면 이는 받아들일

수 있는 접근 방법일 것이다. 그러나 보다 간단한 프로세서의 경우에는 부동소수점 연산을 구현할 만한 실행 시간이나 코드 공간이 없을 수도 있다. ADC 값의 수정 문제를 다루는 방법중 한 가지는 룩업 테이블을 이용하는 것이다. 그러나 이 방법은 가능한 모든 ADC 값에 대한 참조 값—10 비트 ADC의 경우 1,024 워드 크기의 테이블—을 유지하기에 충분한 비휘발성 저장 장치를 필요로 한다는 단점이 있다.

전압 기준 값은 공칭 값에 꽤 가깝다. 그렇지 않다면 기준 값으로서는 유용하지 못할 것이다. 기준 값이 동작 온도 상에서 충분히 안정되어 있다고 가정한다면, ADC 오류는 ADC로부터 읽어 들이는 값의 일정 비율이 된다. ADC의 해상도는 한정되어 있으므로 ADC 오류를 1 LSB 이상의 정밀도로 보정하는 것은 아무런 의미가 없다.

이러한 사실을 염두에 둔다면 ADC의 보정 과정을 단순화할 수 있다. 룩업 테이블 대신에 소프트웨어에게 오류를 보정하려면 ADC 판독 값에서 몇 (이진수) 퍼센트를 가감해야 할 지를 알려주는 값을 저장하는 것이다. 즉, 1 LSB의 정확도에 이르기까지 1/8, 1/16 또는 1/24을 가감할 수 있다. 이를 위해서는 단지 보정 상수 하나만 저장하면 되며, 계산 과정은 일련의 시프트 및 가산(shift-and-add)이나 시프트 및 감산(shift-and-subtract) 연산으로 이루어진

다.

예로 든 2.47 V는 ADC 값에 .988을 곱해줌으로써 보정할 수 있었다. 초기 값에서 1/128, 1/256, 그리고 1/512을 차례로 빼줘도 동일한 결과를 얻을 수 있다. 원래의 2V 입력 예제 값에 정수 연산을 수행하여 얻은 결과는 다음과 같다:

$$829-829/128-829/156-829/512=829-6-3-1=819$$

이 결과는 ADC 판독 값을 819로 보정해 주는데, 이는 기준 공칭 값이 2.5 V일 경우 이상적인 값이다. 이와 유사하게 기준 값이 2.53 V일 때 읽은 값들은 1/128과 1/256을 더해줌으로써 보정할 수 있다.

현재 사용 중인 입력 값에 정밀한 보정 전압을 인가할 필요는 없다는 사실에 주의하자.

ADC가 보정하고자 하는 기준 값을 사용하는 한 여분의 ADC 입력을 어느 것이든 사용할 수 있다.

기준 값이 예상 동작 온도 범위에서 충분히 안정적이 되도록 해야만 한다. 그렇지 않을 경우, 결과 값들은 보정시의 온도 근처에서만 신뢰성을 갖게 된다.

이 접근 방법은 결과 값 이동시 발생하는 절단(truncation)으로 인한 반올림 오류를 야기한다.

2.47 V의 예를 이용하여 스프레드시트를 작성해 보니, 모든 경우에 있어서 보정된 값은 이상적인 값의 2 카운트 내에 들었다. 이 정도 수준의 보정은 2 V 입력에 대한 본래의 변화(10 카운트)보다 훨씬 우수한 것으로서, 많은 애플리케이션들에 있어서 이 정도면 충분하다. 이 정도의 오류 조차도 허용할 수 없는 애플리케이션이 있다면 그야말로 더 나은 기준 값이 필요하거나, 혹은 수작업으로 조정할 수밖에 없을 것이다.

이 교정 기법은 저항기의 허용오차 스택업과 같은 다른 시스템 부정확성을 보상하는 데도 사용할 수 있다. 측정 대상이 전압 입력 값으로 이루어져 있다면, 그것에 정밀 전압을 인가하고 한 번의 보정 작업을 수행함으로써 ADC의 기준 값 변화와 입력 조절에서의 저항기 허용오차 효과를 상쇄할 수 있다.

보정 값의 계산 및 이용

보정 값은 기지의 기준 값을 읽은 뒤 사용할 수정 약수(이진 약수)를 찾아냄으로써 계산할 수 있다.

주어진 예의 경우, 이상적인 ADC 값과 최악의 경우의 ADC 값 간의 차이는 1.2 퍼센트를 넘을 수 없으므로 원래 값의 절반이나 4분의 1로 시작해 봤자 아무런 의미도 없다. 유일하게 테스트하여 사용한 값들은 1/128, 1/256 및 1/512이다.

계산기를 이용하면 손쉽게 보정 약수를 알아낼 수 있지만, 자신의 애플리케이션에 사용하려는 고정소수점 프로세서 상에서 이를 계산해야만 한다면 정수 기반의 접근 방법이 필요하다. 그림 2는 이 예에서 보정 상수를 계산 및 이용하기 위해 사용한 알고리즘을 흐름도 형태로 보여주고 있다.

이 방법에서는 1 바이트(또는 워드)를 이용함으로써 보정 상수를 저장한다.

비트 7은 기준 전압이 로우 상태(보정 값들을 감소해야 함)인지 혹은 하이 상태(보정 값들을 가산해야 함)인지를 알려 준다. 비트 0, 1, 그리고 2는 1/128, 1/256 및 1/512중 어떤 약수를 사용하는 지를 나타낸다.

보정 값의 기록

모든 보정 방식에 있어서 비휘발성 저장 장치의 사용 가능 여부는 중요한 요소이다. 많은 마이크로컨트롤러들이 EEPROM을 내장하고 있다.

보정 작업은 대개 회로 기판의 테스트시에 수행된다. 대량 생산 환경에서 이는 아마도 일종의 베드오브네일즈(bed-of-nails) 자동 테스트 장비에 의해 이루어질 것이다.

대개는 핀 하나를 접지시킨다든가 해서 프로세서를 일종의 “보정 모드”로 들어가도록 하고 싶을 것이다. 생산 테스트 장비는 이날로그 입력에 매우 정밀한 전압을 인가하고 보정 핀을 접지시키도록 프로그램할 수 있다. 그러면 마이크로컨트롤러는 보정 모드로 들어가 기준 값을 읽어 들인 뒤 보정 값을 계산하거나 록업 테이블을 생성할 수 있다.

어떤 경우에는 마이크로컨트롤러에 보정 코드를 추가할 만한 메모리가 없을 때가 있다. 이런 경우에는 마이크로컨트롤러로 하여금 ADC 값을 하나의 출력 핀(직렬 방식)이나 일단의 핀들(병렬 방식)로 돌려 보내게 하여 생산 테스트 장비가 판독하도록 할 수 있다.

그러면 외부 컴퓨터가 보정 값이나 테이블 값들을 계산한 뒤 동일한 인터페이스를 통해 마이크로컨트롤러로 되돌려 보낼 수 있다.

생산 장비가 마이크로컨트롤러 내부 회로도 프로그램할 경우, 보정 데이터를 플래시 메모리에 프로그램 되는 데이터에 내장시킬 수 있다. 보정되는 기준 값이 마이크로컨트롤러 내에 있다면, 테스트 장비는 먼저 보정 프로그램을 마이크로컨트롤러에 로드시키고 보정을 수행한 뒤에 실제적인 애플리케이션 코드를 로드시켜야만 할 수도 있다.

결정적으로, 일부 초소형 마이크로컨트롤러들은 애당초 보정용으로 내놓을 만큼 충분한 핀이 없다. 이 경우에는 대개 출력 핀이 보정 핀 역할도 수행하도록 할 수 있다. 즉, 외부 저항기를 이용해 핀을 정지시키는 것이다. 생산 장비는 전력을 공급하기 전에 핀을 접지시키으로써 보정 모드를 선택한다.

그 동작 방법은 마이크로컨트롤러가 모든 핀들을 입력 상태로 놓고 전력을 넣는 것이다. 그리고는 보정 핀을 출력으로 구성하기 전에 읽어 들인다. 이 때 핀이 하이 상태라면 정상적인 동작에 들어간다.

하지만 핀이 로우 상태라면 마이크로컨트롤러가 보정 모드로 들어갈 수 있도록 외부적으로 접지시켜야만 한다. 물론 출력부는 핀이 외부 접지될 때 아무 것도 손상시키지 않는 것이어야만 한다.

끝으로, 여분의 ADC 입력에 정밀 전압을 인가하는 기준 값을 보정하고 있다면, 그 입력 자체를 이용하여 시스템을 보정 모드로 바꿀 수 있다. 즉, 저항기 하나를 이용해 여분의 입력을 제로 스케일의 ADC 전압(우리가 사용한 예에서는 접지)으로 만든다. 그런 다음 핀 에서 사전 결정된 임계값(예컨대, 전체 전압의 2/3)을 넘는 전압이 감지되면 소프트웨어가 보정 모드로 들어가도록 하는 것이다.

보정 전압을 선택할 때는, 기준 전압이 가능한 최하 값일 경우에 ADC를 포화시키지 않는 범위에서 최대 값을 선택하고 싶을 것이다. 그러면 비트 반올림 오류로 인해 보정 상수(또는 테이블)의 계산시 정확성을 잃는 일이 없게 된다.

이는 대개 보정 전압이 전체 값의 90 퍼센트 이상이 되도록 해주지만, 그보다는 설계가 수월해지도록 그 값에 가장 가까운 표준 기준 전압을 선택하고 싶을 것이다.

일부 애플리케이션에서는 ADC 입력의 변화를 찾아냄으로써 기준 값 문제를 피해갈 수 있다. 그 값을 고정 값에 비교하는 대신에 광 센서가 10 퍼센트 변화하는지 혹은 온도가 25 퍼센트 내려가는지를 지켜 볼 수도 있다. □

[Embedded Systems Programming]

By Stuart Ball

E-mail: stuart@stuartball.com