

마이크로컨트롤러 고객들이 요구하는 것은 무엇인가? 고객들의 요구는 성능과 주변장치이며, 품질과 신뢰성은 당연히 요구되는 것이다. 경쟁력 있는 가격대 또한 필수적인 사항이다. 하지만 무엇보다도 임베디드 소프트웨어의 갈수록 높아지는 복잡성에 대처할 수 있는 효과적인 솔루션을 제공하는 것이 진정한 경쟁 우위일 것이다. 여러 조사에 따르면 임베디드 소프트웨어 개발이 갈수록 더 많은 인원의 팀에 의해서 이루어지고 있다. 그런데 예정된 일정으로 완수되는 프로젝트 분량은 50%에도 미치지 못한다. 소프트웨어 복잡성과 검증이 임베디드 개발 커뮤니티의 두 가지 중대한 과제이다.

그러므로 지능적이고 유연한 솔루션이 필요하다. 시장의 모든 당사자들이 동일한 코어를 이용한다면 컴파일러 업체들의 작업이 훨씬 수월해질 것이다. 하지만 그렇다고 해서 임베디드 소프트웨어 디자이너의 복잡성과 개발 시간이 크게 줄어들 수 있을 것인가? 아마 그

8비트 MCU 용의 대다수 소프트웨어조차 C나 C++로 프로그램하고 있다. 한편 디자이너들은 생산성 측면의 필요와 까다로운 출시시간을 충족하기 위해서 전문적이며 강력한 컴파일러 및 디버거를 기꺼이 구입할 것이다. 그런데 또 한편에서는 인피니언의 DAVE™ Bench 같은 무료 툴 체인이 평가 및 개발 작업에 있어서 중요한 역할을 하고 있다. 첫째, 이러한 툴은 고객의 평가 작업을 수월하게 하고 비용을 절감할 수 있도록 하며, 둘째, 상용 툴 체인이 반드시 달성해야 하는 최소한의 성능 기준을 제시함으로써 혁신을 촉진하는 촉매제 역할을 한다.

소프트웨어 라이브러리는 임베디드 소프트웨어 개발을 좀더 관리가 용이하도록 하는 또 다른 유용한 요소이다. 이들 라이브러리는 개발자가 표준적 알고리즘을 최적화하는 부담을 덜 수 있기 때문이다. 그러므로 TriCore DSP 라이브러리 같은 DSP 라이브러리나 로우 레벨 드라이버들이 널리 이용되고 있다. 하지만 효

소프트웨어 체제에서의 마이크로컨트롤러 자동 코드 생성기와 레퍼런스 디자인을 이용해 갈수록 높아

글/Dr.Stephan Zizala, Senior Director, Infineon Technologies AG

령지 않을 것이다. 가장 큰 문제는 주변장치에 의한 것이기 때문이다. 이러한 점에서 툴 체인, 라이브러리, 자동 코드 생성기, 레퍼런스 디자인이 갈수록 중요해지고 있다. 소프트웨어 개발을 위해 최적화된 마이크로컨트롤러 하드웨어에 미래가 달려 있는 것이지 그 반대는 아니다.

복잡한 소프트웨어를 관리가 용이하도록 하기 위해서는 컴파일러와 디버거가 중요한 역할을 한다. 타이밍적으로 매우 중대한 작업을 제외하고는 마이크로컨트롤러를 더 이상 어셈블리 언어로 프로그램하지 않는다.

과적인 임베디드 소프트웨어 개발을 위해서 무엇보다 혁신적이었던 것은 정적 라이브러리에서 DAVE 같은 GUI 기반 마이크로컨트롤러 구성 툴로 전환한 것이다. 이 툴은 거의 10년 전에 개발된 것인데 여전히 널리 이용되고 있으며 특히 강력한 주변장치를 위한 로우 레벨 드라이버를 개발할 때 높이 평가되고 있다.

위에서 열거한 요소들과 함께 소프트웨어 개발자들은 또한 마이크로컨트롤러 하드웨어에 대해 깊이 있는 이해를 필요로 한다. 이 문제를 해결하기 위해서는 DAVE Drive 같은 자동 코드 생성기가 앞으로 중요한

역할을 할 것이다. DAVE Drive는 에너지 효율 전기 드라이브 용의 코드 생성을 위해 많은 이점을 가능하게 한다. 임베디드 소프트웨어 개발자들은 자신이 선택한 모터 제어 토폴로지에 적합한 제어 알고리즘을 선택하고 파라미터를 구성하기만 하면 된다. 그러면 이 자동 코드 생성기가 특정한 마이크로컨트롤러 하드웨어에 대해서 완벽하게 맵핑을 실시한다.

갈수록 높아지는 디자인 복잡성에 대처하기 위해서는 강력한 컴파일러 및 디버거, IP 라이브러리, 자동 코드 생성기와 함께 인피니언의 모터 제어를 위한 애플리케이션 키트나 연결성을 위한 IO-Link 키트 같은 레퍼런스 디자인을 이용하는 것이 유용할 것이다. 이들 레퍼런스 디자인은 소프트웨어 개발자가 전력 스테이지, 센서, 트랜시버, 심지어 소형 모터를 포함하는 실제적인 애플리케이션 셋업으로 자신의 소프트웨어를 테스트하기에 적합하도록 되어 있다. 그러므로 목표 하드웨어가 개발되기를 기다리느라 소요되는 시간을 절약할

지는 디자인 복잡성에 대처



Dr. Stephan Zizala

수 있다. 여기서 한 걸음 더 나아가서 이와 같은 레퍼런스 디자인들을 이용해서 소프트웨어를 최적화하고 테스트할 수 있다. 그런 다음, 일련의 제품들 중에서 성능과 가격 요구를 충족하는 마이크로컨트롤러를 선택하는 등과 같이, 소프트웨어에 따라서 최종적인 하드웨어를 최적화할 수 있다.

이러한 추세는 갈수록 더 빨라질 것이다. 아날로그-디지털 변환, 통신, 연결성 등을 위해서 갈수록 더 많은 수의 강력한 주변장치들이 마이크로컨트롤러로 통합될 것이다. 이러한 경향의 가장 큰 동력은 비용적인 이유

이며, 한편 마이크로컨트롤러 업체들은 계속해서 자신들의 주변장치를 차별화할 것이다. 하지만 무엇보다도 앞으로의 관건은 마이크로컨트롤러 하드웨어를 좀더 유연하고 구성 가능하도록 만드는 것이다. 그럼으로써 지금보다 훨씬 더 다양한 애플리케이션 유형을 포괄하며 대다수 마이크로컨트롤러 업체들이 지원하는 플랫폼 솔루션이 가능해질 것이다. 한편 이렇게 되면 마이크로컨트롤러가 훨씬 더 복잡해질 것이다. 바로 이런 점에서 GUI 기반 구성 툴, 자동 코드 생성기, 레퍼런스 디자인이 앞으로 더욱 더 중요해질 것이다. **SN**