Embedded System의 특징

AESOP/고도리

임베디드 시스템의 특징(1/9)

- 왜 임베디드 시스템은 다른가?(PC와의 차이점)
  - ✓ 일반적인 계산 목적이 아니라 특별한 임무를 위한 시스템
  - ✓ 목적에 알맞은 Processor의 특징에 의존한다
    - ➢ 목적에 알맞는다면 어떠한 Processor도 선정가능
    - ➢ Ex> 로봇완구 – Pollux, S3C6410, x86도 가능
  - ✓ 가격에 민감하다.
  - ✓ 실시간으로 동작해야 하는 경우가 있다
  - ✓ 일반적으로 시스템용으로 정해진 Software가 없다.
  - ✓ Software적인 fail문제는 PC보다 훨씬 심각한 문제를 야기할 수 있다.
  - ✓ 전원의 제약을 받는다.
  - ✓ 열악한 환경에서의 동작이 요구되는 경우도 있다(ex> 기지국)
  - ✓ PC보다 훨씬 적은 resource (요즘에는 PC에 근접)
  - ✓ Storage에 대한 제약이 있다.
  - ✓ 효과적인 개발을 위해서 특정한 툴이 필요하다(ex> JTAG)

임베디드 시스템의 특징(2/9)

- 왜 임베디드 시스템은 다른가?(계속)
  - ✓ Stand-alone: 독립적이냐 아니면 종속적이냐(ex> 단말/기지국)
  - ✓ HMI(Human-Machine Interface) 설계가 중요
  - ✓ 테스팅 및 검증의 어려움 → 제품이 출시된 후 실제 상황에서 문제가 생기는 경우가 많음
    - ➢ 기지국의 특정 site에서의 fail
    - ➢ 세탁기를 이용한 감자씻기의 예(아프리카에서 세탁기를 이용 감자를 씻은 적이 있음. 고장시 누구 문제인가???)
  - ✓ Hardware의 문제인지 software의 문제인지 명확한 판단이 어려움

임베디드 시스템의 특징(3/9)

- 가격적인 측면
  - ✓ ***보통*** 가격에 민감한 경우가 많다.
  - ✓ 대량 생산 제품의 경우는 가격에 민감하지만, 인공위성-화성탐사선과 같은 기능위주의 제품은 가격에 덜 민감한 편
  - ✓ 부품단가 위주로 계산
  - ✓ 여러가지 factor를 위주로 가격적인 측면을 계산
    - ➢ NAND 와 After Service
    - ➢ 단말의 경우 A/S 비용증가 때문에 응급복구 시스템의 구성
    - ➢ 위성이나 시스템 동작시 사람의 접근이나 변경이 힘든 경우는 비용을 늘이더라도 안정한 시스템을 만들어야 함
    - ➢ 즉, 비용대비 성능의 미묘한 선에서 결정을 해야한다 ➔ 시스템 ***설계자의 역량***에 따라 제품의 성공/실패가 결정될 비율이 높음

임베디드 시스템의 특징(4/9)

- 실시간 동작의 필요성
  - ✓ 일반 PC의 경우와는 다르게 즉각적인 응답이 필요한 경우가 있음
  - ✓ 단말보다는 주로 제어 시스템의 경우가 많음
  - ✓ 시간에 민감하게 제약 받는 경우(time sensitive constraint)
    - ➢ Ex> print출력 시간
  - ✓ 시간에 결정적으로 제약 받는 경우(time critical constraint)
    - ➢ Ex> 비행기 제어시스템
    - ➢ Ex> 엘리베이터 제어 시스템
  - ✓ 비용대비 실시간의 필요성과 software사이에서의 고려
    - ➢ 상용 RTOS, 공개 RTOS, 리눅스를 이용한 실시간 구성(timer interrupt를 이용한)
    - ➢ 개발환경과 비용과의 관계도 고려. 즉, 목적하는 시스템의 성능에 따라 고려를 해야 함.

임베디드 시스템의 특징(5/9)

- **Software의 오동작 문제**
  - ✓ Software가 오동작 했을 경우 일반 PC보다 훨씬 심각한 문제가 발생할 가능성이 높음
  - ✓ 단말보다는 제어 시스템쪽에 더 심각한 증상
  - ✓ 오동작의 심각성
    - ➢ 무인시스템의 경우의 문제 ➔ 기지국, 위성시스템, 군사관련시스템
    - ➢ 인간과 관련된 시스템 ➔ 엘리베이터 제어시스템의 오동작, 교통제어 시스템의 오동작
    - ➢ 단말의 특정 부분 제어 시스템의 문제 ➔ GPIO핀의 잘못된 제어로 인한 발열 및 파손
  - ✓ 오동작에 대한 대비책을 세워야 함
    - ➢ Watchdog timer등과 같은 시스템을 감시하는 다른 시스템의 도입

임베디드 시스템의 특징(6/9)

- 전원문제
  - ✓ PC에서는 발생하지 않는 전원에 대한 문제가 발생할 경우가 있음
  - ✓ 저전력문제
    - ➢ 단말기 – 이동전화, Tablet PC 등
    - ➢ Ex> 지리산 반달곰 위치 표시 시스템(배터리가 떨어졌을 경우 어떻게 할 것인가????)
  - ✓ Hardware의 문제? → 저전력 지원 SoC
  - ✓ Software의 문제? → 저전력 지원 SoC를 제어하는 software의 유무 (Power Management System)
  - ✓ 시스템의 목적에 알맞은 전원 디자인
  - ✓ 전원에 대한 문제는 시스템 디자인 결정에 골고루 영향을 미침
  - ✓ 전력의 제약은 SoC의 선택, 속도의 선택에 영향을 미침
- 발열문제
  - ✓ 열이 많이 나는 chip을 사용할 경우
  - ✓ 밀폐된 공간에서의 발열문제(ex> 벽에 장착된 web-pad)
  - ✓ 다른 부품에 의한 발열 문제(ex> LCD에 의한 발열)

임베디드 시스템의 특징(7/9)

- 열악한 동작환경 조건
  - ✓ 임베디드 시스템은 어디서나 사용됨.
  - ✓ 비행기, 북극, 사막, 우주에서도 동작해야 함
  - ✓ 이에 따른 Hardware에 대한 제약사항 발생(ex> Industrial version의 부품들을 사용해야 함)
    - ➢ 문제가 개발의 종료시점에 나타나는 경우가 많음
    - ➢ 왜냐하면 이런 동작조건테스트는 대부분 제품출시 이전에 이루어짐
    - ➢ Specification과는 다른 동작(Chamber test시)
    - ➢ 온도 문제 때문에 RTC(Real Time Clock)의 동작에 문제가 생길 경우
    - ➢ Display장치의 경우 온도에 취약함(ex>LCD)
    - ➢ 전자파 문제
    - ➢ 제품의 출시 시점에 심각한 영향을 끼침

임베디드 시스템의 특징(8/9)

- 일반 PC보다 훨씬 적은 resource를 가지고 있음
  - ✓ RAM크기의 제한
  - ✓ SoC의 기능의 제한(속도/3D지원 등과 같은)
  - ✓ 사용할 수 있는 부품의 제약: PCI등과 같은 PC기반부품이 많은 경우는 chip을 구하기 힘들고, 제품가격에도 문제가 생김
  - ✓ 이에 따른 software의 수정사항이 생김 ➔ Chip을 구하기는 하였으나 Device Driver를 구하기 힘든 경우와 많이 사용하지 않는 경우가 많으므로 software적인 Bug가 검증이 안된 경우가 많을 수 있음.
  - ✓ Storage 매체의 제한
    - ➢ 저장매체로 NOR/NAND등의 Flash를 사용한 경우가 많음
    - ➢ 용량에의 제한으로(가격대비 용량문제) Software Packaging에의 문제점 – Linux의 Rootfs이 클 경우
    - ➢ Android system의 경우 이런 문제를 극복하기 위해서 근래는 eMMC와 SD card등과 같은 저장매체를 사용

임베디드 시스템의 특징(9/9)

- 효과적인 개발을 위한 특정 툴과 개발방법이 필요
  - ✓ 전용 디버깅 장비를 사용하는 경우가 많음(ex> Trace-32)
  - ✓ Cross-toolchain이 필요하다
  - ✓ 디버깅에 제한을 많이 받는다.