

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

RC-CAR let's get it

devie tree

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – hoseong Lee(이호성)

hslee00001@naver.com



# 목차

---

1. 디바이스 트리 이해

2. 기본 작성법

3. Zynq device tree

### 1.1 노트

- 디바이스 트리는 노드들로 구성된 트리 구조를 갖는다. 노드는 또 다른 노드들과 **property** 라고 불리는 속성을 갖는다. 속성은 키 와 값으로 구성된 한쌍으로 이루어져 있다.

“/” 는 디바이스 트리에 표현하고 싶은 전체 장치의 최상위 루트 노드라는 의미이다.

기초적인 형식은 다음과 같다. 노드명 {노드에 대한 내부 내용};

### 1.2 속성

노드는 자식노드 뿐 아니라 속성을 가질 수 있다.

속성은 키와 값으로 구성된 한 쌍이다. 키는 반드시 있어야 하지만 키에 대한 값은 없을 수 도 있다.

속성의 형식은 키 = 값; 이다.

키는 아스키 문자열 형태로 표현, 값은 데이터 형이 따로 정해져 있지 않고, 이진 바이트 배열로 작성된다 .아래는 속성 값을 기술하기 위한 4가지 표현 방법이다.

1) 속성 값에 NULL로 끝나는 문자열을 쓰고 싶다면 쌍 다음표를 사용하여 문자열을 표현한다.

→ String-property = “a string”

2) 2비트 부호 없는 정수형은 < > 로 표현한다.

→ Cell-property = <0xbeef 123 0xabcd1234>

3) 이진 데이터는 [ ] 로 표현한다.

→ Binary-property = [0x01 0x23 0x45 0x67]

4) 위 세가지를 혼합하여 사용하여 표현할때는 , 로 분리하여 사용한다.

→ Mixed-property = “a string”, [0x01 0x23 0x45 0x67], <0x12345678>

## 2.1 최초 구조

최소한의 형태의 구조는 다음과 같다. /{};

이제 이 노드에 보드에 대한 고유한 식별자를 지정해야 한다.

```
/{}
    compatible = "xlnx,zynq-7000"
};
```

`compatible` 이란 속성을 이용하여 우리가 표현 하고자 하는 보드에 대하여 기술하고 있다. `compatible` 속성은 항상 “제조사,모델”순으로 문자열 형태로 표현한다. 네임스페이스 충돌 문제가 발생하지 않게 하는 일종의 약속이다. (xlnx라는 회사이고 모델은 zynq-7000이다.) 운영체제는 이 `compatible` 속성 값을 이용해서 다루고자 하는 디바이스에 대한 정보를 찾아간다.

## 2.2 CPU에 대한 표현

보드 자체에 대한 것을 디바이스 트리에 표현했으므로 해당 보드에 있는 CPU에 대한 내용을 추가 할 것이다. CPU에 대한 추가는 “cpus” 라는 노드 이름을 갖는 노드를 이용해서 표현한다.

```
/{}
    compatible = "xlnx,zynq-7000"

    cpus{
        cpu@0{
            compatible = "arm,cortex-a9";
        };
        cpu@1{
            compatible = "arm,cortex-a9";
        };
    };
};
```

cpus 노드는 다시 cpu@0 와 cpu@1 이라는 형식의 이름을 갖는 자식 노드를 가진다.

이 자식 노드들은 어떤 cpu인가의 정보를 표현하기 위해 `compatible` 속성을 사용한다.

마찬가지로 “제조사,모델” 형식의 값을 가진다.

더 많은 속성들이 각 cpus 노드에 추가될 수 있다.

## 2.3 노드 이름 규칙

위 예에서 각 cpu 의 노드 이름을 보듯이 노드들의 이름을 표현하는 고유 규칙이 있다. 모든 노드는 “이름@장치주소” 와 같은 형태로 표현한다. @장치주소 는 동일한 디바이스를 나타내거나 동일한 내용을 나타내는 노드가 없다면 생략 가능하다. cpus가 바로 그런 예이다. 이 형식에서 “이름” 부분은 최대 31개의 문자 길이를 갖는 간단한 아스키 문자열로 표현한다. 보통 노드의 이름은 장치의 종류를 나타낼 수 있는 이름으로 짓는다. 예를 들면 3com 이더넷 어댑터를 나타내는 노드는 3com509 가 아닌 ethernet 이라고 이름을 짓는다. 노드 이름에서의 “장치 주소”는 단지 노드들이 다른 노드임을 구별하기 위한 용도로 사용된다.

## 2.4 디바이스에 대한 표현

노드명 { 노드에대한 내부 내용};

노드명은 디바이스 타입을 반영한 것이지 특정한 모델을 지칭하는 것이 아니다. 3번의 노드 이름 규칙에서 이름@장치주소 와 같은 형태로 표현 할 때, 디바이스에 대한 표현은 이름으로 표현한다.

## 2.5 compatible 속성 사용법

디바이스 트리에 디바이스를 표현하는 모든 노드는 compatible 속성을 가져야 한다. Compatible 속성은 운영체제에서 동작하는 디바이스 드라이버가 다룰 디바이스에 대한 정보를 찾기 위한 키 값이다. Compatible 은 하나의 문자열만 가지는 것이 아니고 여러개의 문자열들이 나열된 리스트이다. 이 문자 열 리스트의 첫번째 문자열은 디바이스를 정확하게 구별하기 위해서 “제조사, 모델” 형식으로 표현해야 한다.

예) compatible = "fsl,mpc8349-uart", "ns16550";

먼저 운영체제는 fsl, mpc8349-uart 문자열을 인식할 수 있는 디바이스 드라이버를 찾는다. 발견 되지 않는다면 ns16550의 문자열을 인식하는 디바이스 드라이버를 찾을 것이다. 운영체제의 디바이스 드라이버 구현 방법에 따라서는 디바이스 드라이버가 자신을 다룰 디바이스 정보를 얻기 위해서 디바이스 트리에 제공된 각 노드를 탐색하면서 compatible 속성에 선언된 문자열 리스트를 가지고, 처리가 진행된다. ‘fsl,mpc8349-uart’ 는 해당 디바이스 자체를 표현한 것이고, “ns16550” 는 내부 레지스터를 다루는 방법이 16550 uart 와 호환 가능하다는 의미이다. 여기서 ns16550 은 널리 알려져 있기 때문에 제조사를 사용하지 않았다.

```

* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*/
#include/ "skeleton.dtsi"

/ {
    compatible = "xlnx,zynq-7000";

    cpus {
        #address-cells = <1>;
        #size-cells = <0>;

        cpu@0 {
            compatible = "arm,cortex-a9";
            device_type = "cpu";
            reg = <0>;
            clocks = <&clk 3>;
            clock-latency = <1000>;
            cpu0-supply = <&regulator_vccpint>;
            operating-points = <
                /* kHz      uV */
                666667  1000000
                333334  1000000
            >;
        };

        cpu@1 {
            compatible = "arm,cortex-a9";
        };

        i2c0: i2c@e0004000 {
            compatible = "cdns,i2c-r1p10";
            status = "disabled";
            clocks = <&clk 38>;
            interrupt-parent = <&intc>;
            interrupts = <0 25 4>;
            reg = <0xe0004000 0x1000>;
            #address-cells = <1>;
            #size-cells = <0>;
        };

        i2c1: i2c@e0005000 {
            compatible = "cdns,i2c-r1p10";
            status = "disabled";
            clocks = <&clk 39>;
            interrupt-parent = <&intc>;
            interrupts = <0 48 4>;
            reg = <0xe0005000 0x1000>;
            #address-cells = <1>;
            #size-cells = <0>;
        };

        ranges;

        adc: adc@f8007100 {

```

왼쪽 그림은 zynq-7000.dtsi 디바이스 트리 파일의 일부이다. 이를 통해 디바이스 트리 작성법을 알아 보도록 한다.

최상위 노드는 “/”이라는 노드명을 가진다. 노드에 보드에 대한 고유한 식별자로 xlnx사의 zynq-7000이라는 모델로 기술하고 있다.

CPU에 대한 표현은 cpus라는 노드 이름을 갖는 노드로 표현해야 한다.

cpus의 자식노드로 cpu@0, cpu@1 형제 노드가 보인다.

노드 이름 규칙으로 “이름@장치주소”와 같은 형태로 표현하여 디바이스에 대한 타입인 cpu를 표현하고 장치 주소”는 단지 노드들이 다른 노드임을 구별하기 위한 용도로서 0, 1로 구분지어 사용하고 있다.

또한 compatible 속성을 사용하여 ARM cortex-A9 듀얼 코어 시스템을 표현했다.

노드명 cpus는 디바이스 타입을 반영한 것이고, 특정한 모델을 지칭하는 것은 이름인 cpu로 표현한다.

하나, 보통 장치주소는 i2c 에서 처럼 하드웨어의 reg값과 일치 시켜서 사용한다.