

# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정



2018.05.14

53 일차

강사 - Innova Lee(이상훈)

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

학생 - 신민철

[akrn33@naver.com](mailto:akrn33@naver.com)

## UART

위키백과, 우리 모두의 백과사전.

**UART**(범용 비동기화 송수신기: Universal asynchronous receiver/transmitter)는 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송하는 **컴퓨터 하드웨어**의 일종이다. UART는 일반적으로 **EIA RS-232**, **RS-422**, **RS-485**와 같은 통신 표준과 함께 사용한다. UART의 U는 범용을 가리키는데 이는 자료 형태나 전송 속도를 직접 구성할 수 있고 실제 전기 신호 수준과 방식(이를테면 **차분 신호**)이 일반적으로 UART 바깥의 특정한 드라이버 회로를 통해 관리를 받는다는 뜻이다.

통신 데이터는 메모리 또는 레지스터에 들어 있어 이것을 차례대로 읽어 직렬화 하여 통신한다. 최대 8비트가 기본 단위이다.

UART는 일반적으로 컴퓨터나 주변 기기의 일종으로 병렬 데이터를 직렬화 하여 통신하는 개별 **집적 회로**이다. 비동기 통신이므로 동기 신호가 전달되지 않는다. 따라서 수신 쪽에서 동기신호를 찾아내어 데이터의 시작과 끝을 시간적으로 알아 처리할 수 있도록 약속되어 있다. 디지털 회로는 자체의 **클럭 신호**를 사용하여 정해진 속도로 수신 데이터로부터 비트 구간을 구분하고 그 비트의 논리 상태를 결정하여 데이터 통신을 한다.

UART는 보통 마이크로컨트롤러에도 포함되어 있다. 듀얼 UART, 곧 **DUART**는 두 개의 UART를 하나의 칩에 합친 것이다. 수많은 현대의 집적 회로(IC)는 동기화 통신도 지원하는 UART와 함께한다. 이러한 장치들은 **USARTs**(범용 동기화 송수신기: universal synchronous/asynchronous receiver/transmitter)로 부른다.

**목차** [숨기기]

- 1 데이터 송 수신 형태
- 2 UART 모델
- 3 같이 보기
- 4 각주

### 데이터 송 수신 형태 [ 편집 ]

비트 수	1	2	3	4	5	6	7	8	9	10	11
	시작 비트 (Start bit)	5-8 데이터 비트								패리티 비트 (parity bit)	종료 비트 (Stop bit(s))
	Start	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Parity	Stop

가장 일반적으로 각 데이터 비트의 시간에 대해 16/64 배 빠른 **클럭 신호**를 이용하여 시작 비트로부터 세어 각 비트의 경계를 찾아낸다. 이 클럭 신호는 자체적인 내부 클럭 디지털 회로에 의해 발생한다. 보드 설정에 따라 주 클럭으로부터 타이머등을 써서 설정한 속도의 클럭 신호를 만든다. 이것은 프로그래밍에 의한 레지스터 설정에 따라 클럭 신호의 주파수가 바뀐다. 통신 양쪽에서 설정을 미리 약속하고 클럭 신호 발생부의 레지스터를 같은 속도로 설정해야 통신이 원활하게 이루어진다.

- 시작 비트 : 통신의 시작을 의미하며 한 비트 시간 길이 만큼 유지한다. 지금 부터 정해진 약속에 따라 통신을 시작한다.
- 데이터 비트 : 5-8비트의 데이터 전송을 한다. 몇 비트를 사용할 것인지는 해당 레지스터 설정에 따라 결정된다.

## 시리얼 동작 방식

비동기식 시리얼 프로토콜은 몇 가지 규칙에 기반해서 동작합니다.

- Data bits,
- Synchronization bits,
- Parity bits,
- and Baud rate.

프로토콜의 세부 내용들은 유연하게 변경될 수 있도록 짜여져 있습니다. 따라서 두 기기가 시리얼 통신을 하기 위해서는 반드시 같은 설정을 사용해야 합니다.

### Baud Rate (통신 속도)

Baud rate 는 시리얼 라인으로 전송되는 데이터 속도를 말합니다. Bits-per-second (bps) 단위로 표시하는데 이 값을 바탕으로 1 bit가 전송되는데 필요한 시간을 알 수 있습니다. 즉 1bit 데이터를 전송할 때 시리얼 라인을 high/low 상태로 유지하는 시간이며, 데이터를 받기 위해 시리얼 라인을 체크하는 시간입니다.

Baud rates 값은 어떻게든 설정할 수 있지만 통신 속도에 크게 영향을 받지 않는 경우 일반적으로 9600 bps를 사용합니다. 다른 표준 baud are 값으로 1200, 2400, 4800, 19200, 38400, 57600, 115200 을 사용할 수도 있습니다.

값이 높을수록 전송/수신 속도가 빠르지만 115200 를 초과할 수는 없습니다. 많은 마이크로 컨트롤러에서 이 값이 상한선으로 사용됩니다. Baud rate 값이 너무 높거나 양쪽의 설정값이 틀릴 경우 데이터 수신에 문제가 발생합니다.

### Framing the data (데이터 구조)

전송에 사용되는 데이터 패킷은 아래와 같이 구성됩니다. 각각의 구성요소는 Start bit를 제외하고 가변적인 크기를 가질 수 있습니다.

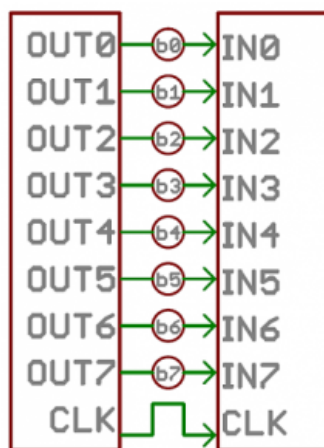
Frame:	Start	Data	Parity	Stop
Size (bits):	1	5-9	0-1	1-2

## 시리얼(SERIAL) 통신 소개

임베디드 시스템은 각종 프로세서와 회로들 간에 서로 통신하며 동작하는 것이 핵심이기 때문에 데이터를 주고 받는 표준 프로토콜들이 필요한 것은 당연합니다. 수많은 프로토콜들이 있지만 일반적으로, 크게 두 개의 카테고리로 나눌 수 있습니다. 바로 패러럴(병렬, parallel)과 시리얼(직렬, serial)입니다.

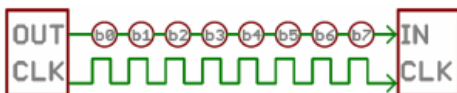
### 패러럴(병렬)과 시리얼(직렬)

패러럴(Parallel) 인터페이스는 다수의 비트(bit)를 한번에 전송하는 방법입니다. 이들은 보통 8 또는 16 또는 그 이상의 라인을 통해 동시에 데이터를 보내줍니다.



그리고 타이밍(clock)에 맞춰 모든 라인이 같이 동작할 수 있도록 clock(CLK) 라인이 필요하므로 8-bit data bus 의 경우 9라인이 사용됩니다.

반대로 시리얼(Serial) 인터페이스는 데이터를 스트림으로 바꿔서(직렬화, serialization) 한 번에 한 비트씩 전송합니다. 아래와 같이 시리얼 인터페이스는 clock 라인을 포함 2라인으로 데이터를 전송할 수 있습니다.



패러럴이 고속도로라면 시리얼은 국도입니다. 대신 고속도로가 국도보다 소모되는 자원이나 비용이 많은 셈입니다.

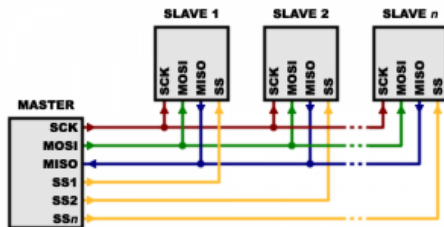
패러럴은 빠르고 상대적으로 구현이 쉽지만 input/output(I/O) 라인을 많이 소모합니다. 아두이노 UNO 보드에 사용한다고 생각해 보시면 패러럴 방식이 소모하는 I/O 라인이 너무 많다고 느껴지실겁니다. 따라서 속도를 좀 희생하더라도 I/O 라인을 아낄 필요가 있습니다.

## I2C 통신 개요

I2C (Inter-Integrated Circuit, 또는 TWI - Two Wire Interface) 는 복수개의 슬레이브 장치가 복수개의 마스터 장치와 통신하기 위한 프로토콜입니다. 앞서 살펴본 SPI 와 마찬가지로 하나의 완성품을 구성하는 요소들 간의 근거리 통신을 위해 고안되었습니다.

시리얼 통신에서 살펴보았다시피 **비동기식 시리얼 통신(이하 UART)**은 클럭을 맞춰줘야 하고 데이터 라인으로 들어오는 신호를 항상 주시해야 하기 때문에 오버헤드가 있으며 하드웨어가 복잡해지는 단점이 있습니다. 결정적으로 UART 통신은 1:1 통신만 가능합니다.

반면 동기식 시리얼 통신인 SPI 는 클럭(CLK) 라인을 이용해 데이터를 동기화 하므로 하드웨어 구조도 간단하고 1:N 통신이 가능합니다. 송신용 핀과 수신용 핀이 분리되어 있고, full-duplex(동시 송수신) 연결을 이용해 10만 비트의 전송 속도를 지원하기도 합니다. 하지만 SPI 통신은 통신에 필요한 핀이 많아지는 단점이 있습니다. 하나의 장치를 연결하는데 4개의 라인 (CLK, MOSI, MISO, SS)이 필요하고, 추가로 장치를 더할 때마다 라인이 하나씩 추가됩니다. 그리고 N:N 통신은 할 수 없습니다.



I2C 통신은 SPI 통신의 이런 단점들을 보완할 수 있는 동기식(synchronous) 시리얼 통신 방법입니다. UART 통신처럼 단 두 라인만 사용하고 1008 개의 슬레이브 장치를 지원합니다. 또한 N:N 통신도 지원이 가능합니다. (단 마스터 장치끼리 통신은 불가) 하드웨어 요구사항이 SPI 보다 복잡하긴 하지만 UART 통신보다는 간단합니다. 통신 속도면에서도 SPI 와 UART 통신의 중간쯤 됩니다.(10kHz/40kHz)

