하드웨어 통신 프로토콜의 이해와 전기신호 분석 실습

mongii@grayhash

대표적인 하드웨어 통신 프로토콜의 종류들

- UART (Universal asynchronous receiver/ transmitter)
- I2C (Inter-Integrated Circuit)
- SPI (Serial Peripheral Interface)

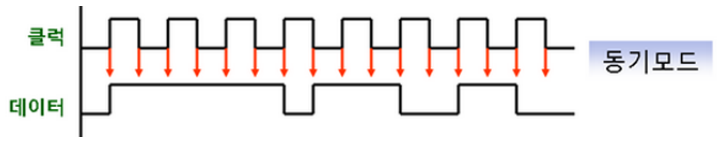
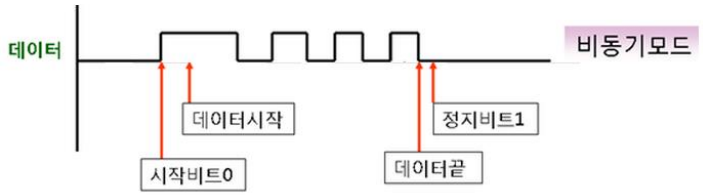
하드웨어 통신이 필요한 경우

- 주변 장치와의 통신
  - CAMERA 모듈
  - Zigbee 무선 통신 모듈
- MCU와 MCU 사이의 통신
- 무엇으로 “Interfacing” 할 것인가?
  - 자신이 선호하는 Interface를 사용하는 칩 구매
  - 자신이 선호하는 Interface로 프로그래밍

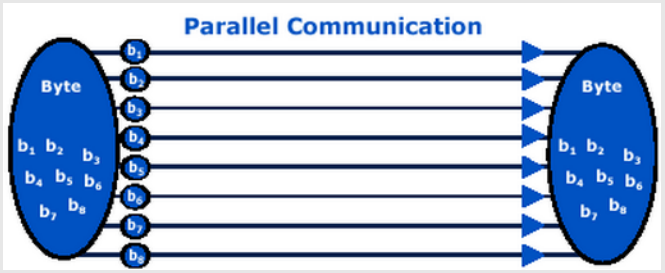
하드웨어 통신이 필요한 경우

- Socket처럼 용도에 따라 사용하기 나름
  - 문자열 통신
  - 데이터, 멀티미디어 통신
  - Zigbee 무선 통신
  - 환경 설정
  - 기기와 PC와의 통신
  - 디버깅 메시지 출력
  - 등등..

동기와 비동기

동기(synchronous)	비동기(asynchronous)
Clock을 기준으로 데이터 참조	시간을 기준으로 데이터 참조
최소 2회선 필요	최소 1회선 필요
	

직렬 연결과 병렬 연결

직렬(Serial)	병렬(Parallel)
데이터 송신 시 1개의 회선 이용	데이터 송신 시 다수의 회선 이용
속도 느림	속도 빠름
비용 절감	비용 많이 듦
	

1. UART

- Universal asynchronous receiver/transmitter
  - 범용 비동기 송/수신기
- 가장 오래 된 하드웨어 통신 프로토콜 중 하나
- 직렬 통신 프로토콜
  - 데이터 송신/수신 시 각각 하나의 LINE만 이용
- 2개의 라인 사용
  - TX : 송신 라인
  - RX : 수신 라인
- Baudrate로 동기화 수행
- 디버깅 용도로 특히 많이 사용됨

2. I2C

- I²C (Inter-Integrated Circuit)
  - 아이스퀘어씨, 아이투씨
- TWI(Two Wire Interface)라고도 불림
- 필립스사에서 개발
- 1:N 통신 가능, 양방향 통신 가능
- 직렬 통신, 동기 방식
- 2개의 라인 사용
  - SDA : 데이터
  - SCL : 클럭

3. SPI

- SPI (Serial Peripheral Interface)
- 모토롤라사에서 개발
- 1:N 통신 가능
- 3개의 라인 사용
  - MISO, MOSI, SCK
- MISO : Master-In, Slave-Out
- MOSI : Mater-Out, Slave-In
- Socket의 server/client와 비슷한 개념

하드웨어 프로토콜 비교

	UART	I2C	SPI
동기/비동기	비동기	동기	동기
통신핀의 구성	송신핀 수신핀	클럭 송수신핀	클럭 송신핀 수신핀
통신대상수	1:1	1:N	1:N

전기 신호 분석 실습

Logic Analyzer 소개

- 전기 신호를 캡쳐하여 0과 1로 보여주는 장비

UART 신호 분석 실습

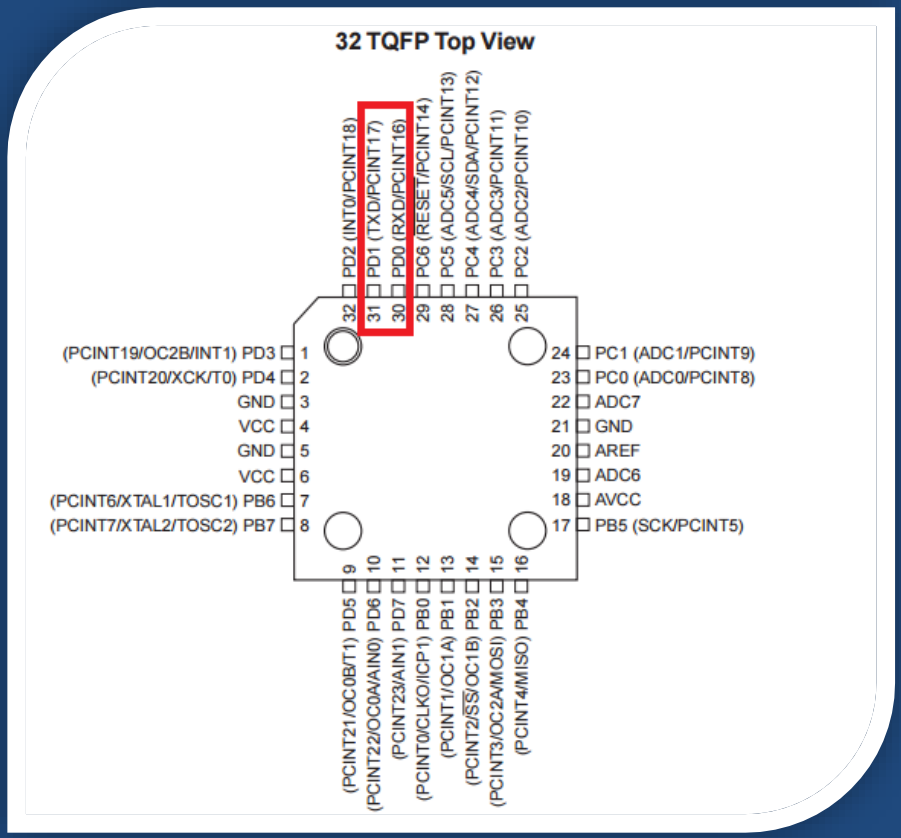
UART 프로그래밍

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
/* Status Register 0A */
UCSR0A = 0x00;

/* Status Register 0B */
/* RX/TX Enable = 10001000 */
UCSR0B = 0x88;

/* Status Register 0C */
/* No parity, 8bit = 0110 */
UCSR0C = 0x06;

/* 중요 : Baud Rate 설정 */
/* BPS = 9600 */
UBRR0H = 0;
UBRR0L = 103;


while(1)
{
    UDR0 = 'A';
    _delay_ms(1000);
}
}
```

UART 프로그래밍

- UART에 사용되는 레지스터 목록
  - UCSR0A
  - UCSR0B
  - UCSR0C
  - UBRR0H
  - UBRR0L
  - UDR0

- UART에 사용되는 핀들
  - PE0 (RXD0)
  - PE1 (TXD0)



코드 설명

- UCSR 레지스터
  - Control and Status Register의 약자
  - 통신 환경 설정

- UBRR 레지스터
  - Baud Rate Register의 약자
  - 통신 속도 정의

  * UART CALC :
    http://wormfood.net/avrbaudcalc.php

코드 설명

- UCSR0A 레지스터
  - 송수신되는 데이터의 현황이 기록됨 (00000000)

7	6	5	4	3	2	1	0
데이터 수신완료 시 1	데이터 송신완료 시 1	송신 버퍼가 비어있으면 1					

- UCSR0B 레지스터
  - 기능 설정 (10001000)

7	6	5	4	3	2	1	0
데이터 수신 시 인터럽트 발생			수신(RX)부 활성화	송신(TX)부 활성화			

코드 설명

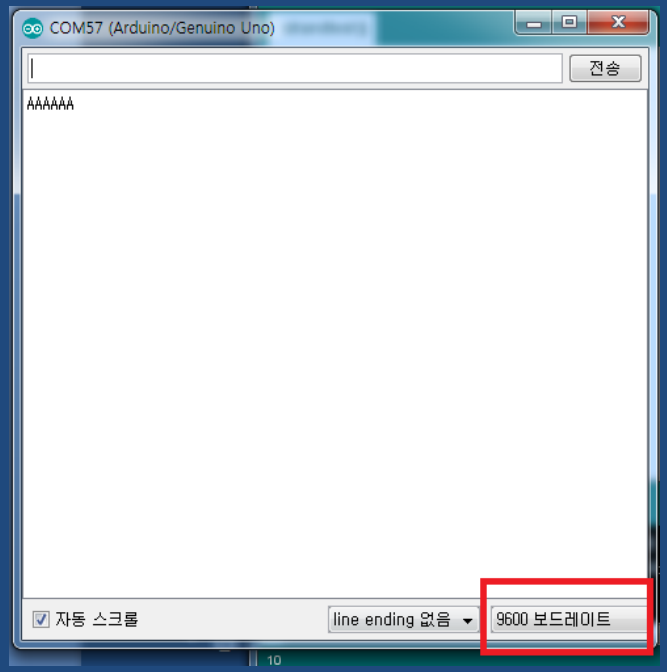
- UCSR0C 레지스터
  - 통신 설정 (00000110) => 8비트 데이터

7	6	5	4	3	2	1	0
X	동기or비동 기모드 선택	패리티 모드(1)	패리티 모드(2)	스톱 비트	데이터 비트(1)	데이터 비트(2)	

- UBRR0H, UBRR0L 레지스터
  - 통신 속도 설정
  - 초당 전송되는 bit의 수
  - /* BPS = 9600 */
  - UBRR0H = 0;
  - UBRR0L = 47;

아두이노 UART 프로그래밍

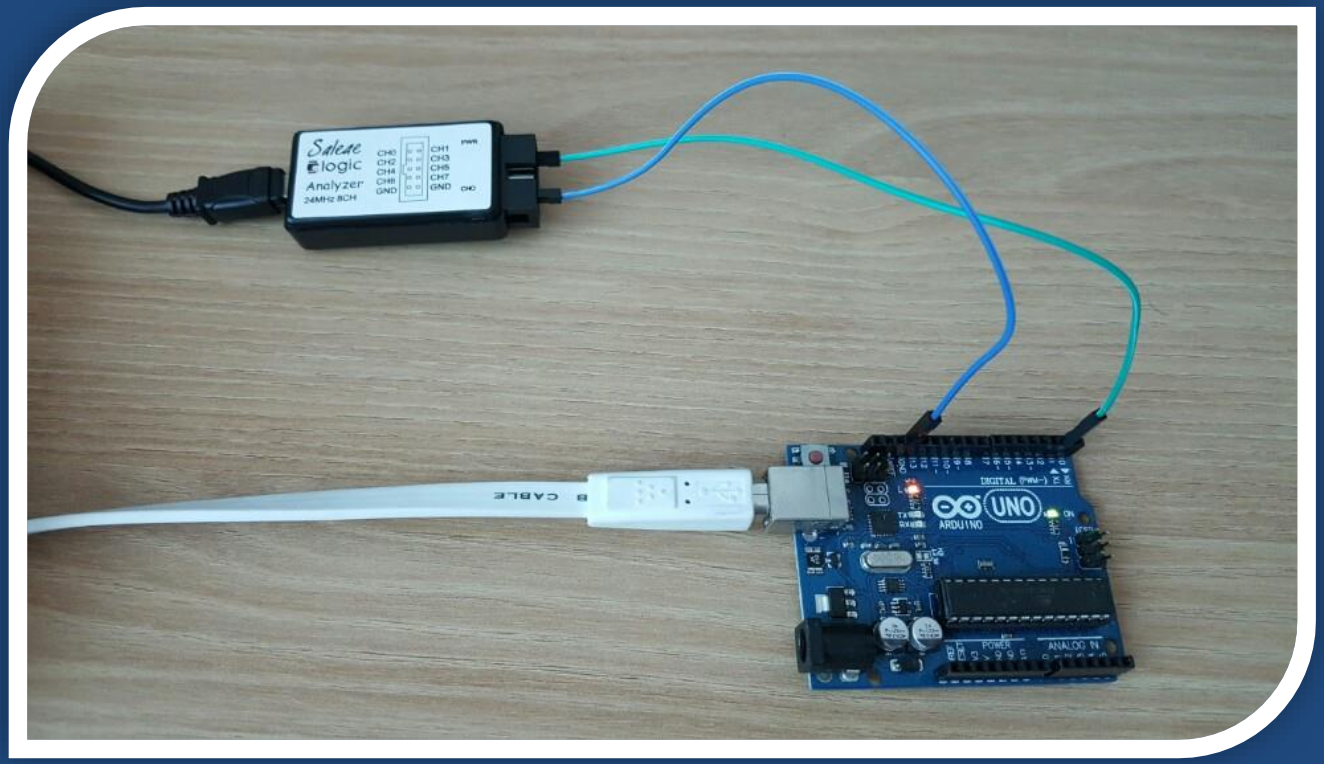
```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("A");
  delay(1000);
}
```



- USCR0A/B/C, UBRR0H/L, UDR0 레지스터를 몰라도 됨
- 장점이자 단점(특히 해커들에게는!)이 될 수 있음

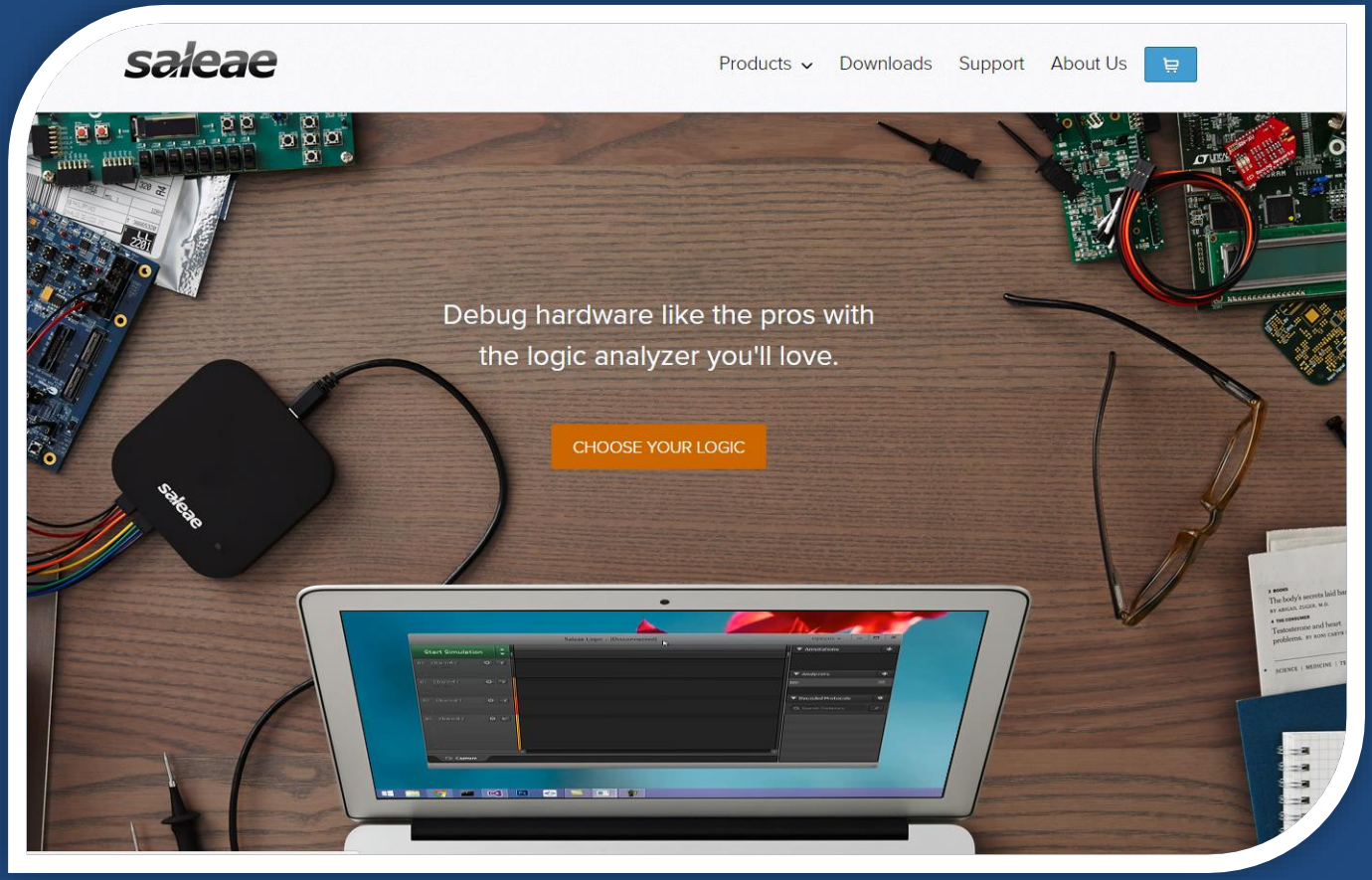
Logic Analyzer 연결

- GND = GND
- CH0 = TX



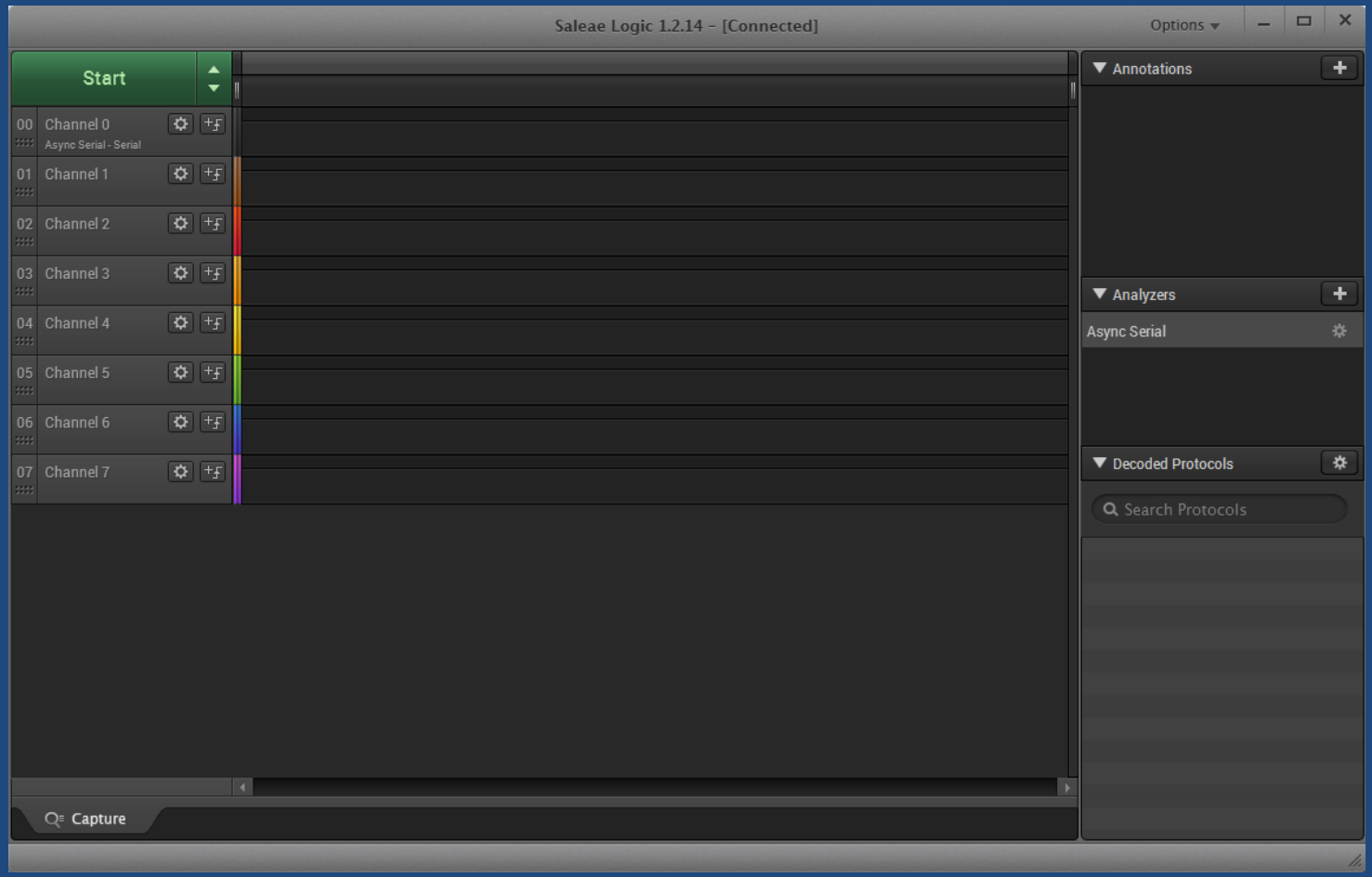
Saleae Logic Analyzer 설치

- https://www.saleae.com/



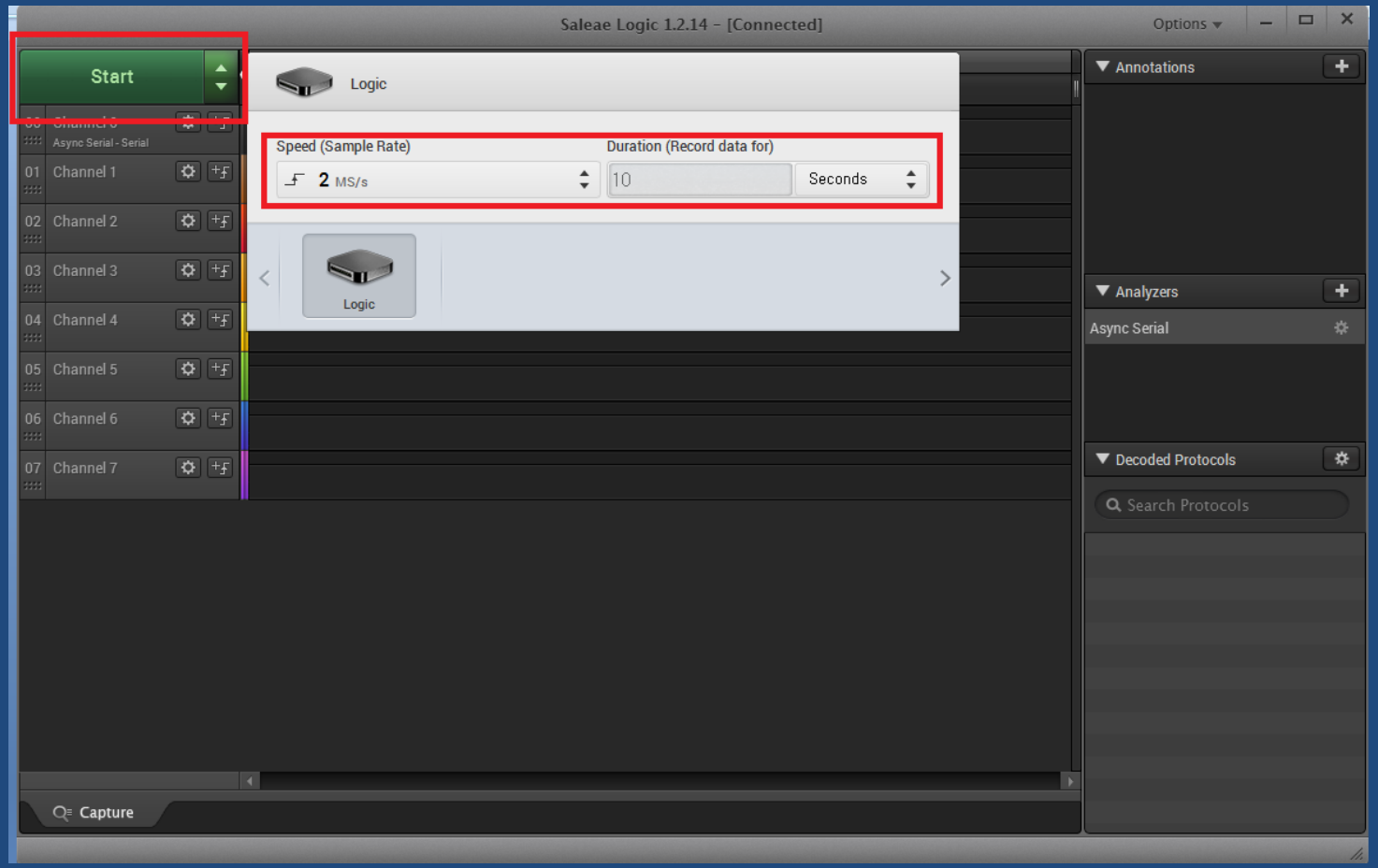
Logic Analyzer 실행

- “Connected” 확인

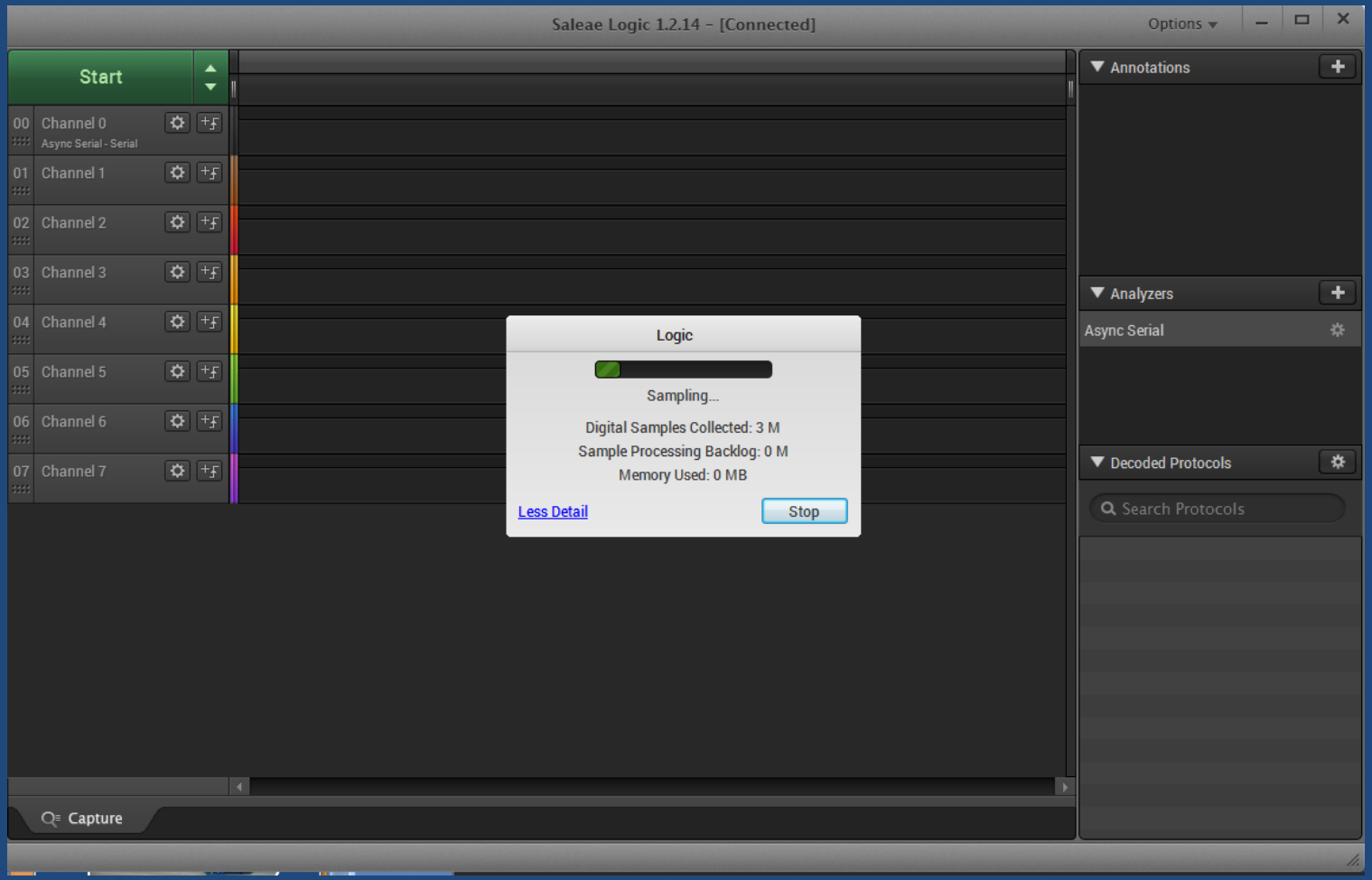


신호 분석 시작

- Speed, duration 설정 후 Start 버튼 클릭

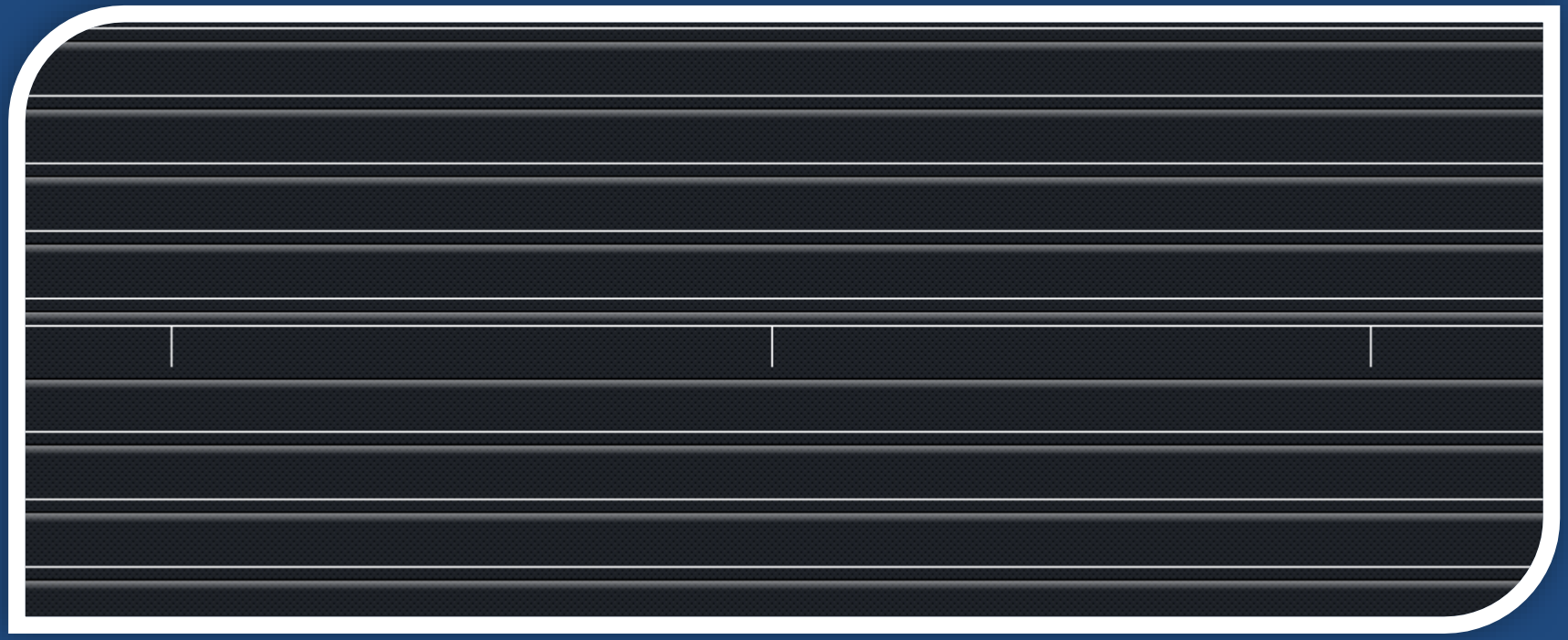


신호 분석 시작


Saleae Logic 1.2.14 - [Connected]
Options
Start
00 Channel 0
Async Serial - Serial
01 Channel 1
02 Channel 2
03 Channel 3
04 Channel 4
05 Channel 5
06 Channel 6
07 Channel 7
Annotations
Analyzers
Async Serial
Decoded Protocols
Search Protocols
Logic
Sampling...
Digital Samples Collected: 3 M
Sample Processing Backlog: 0 M
Memory Used: 0 MB
Less Detail
Stop
Capture

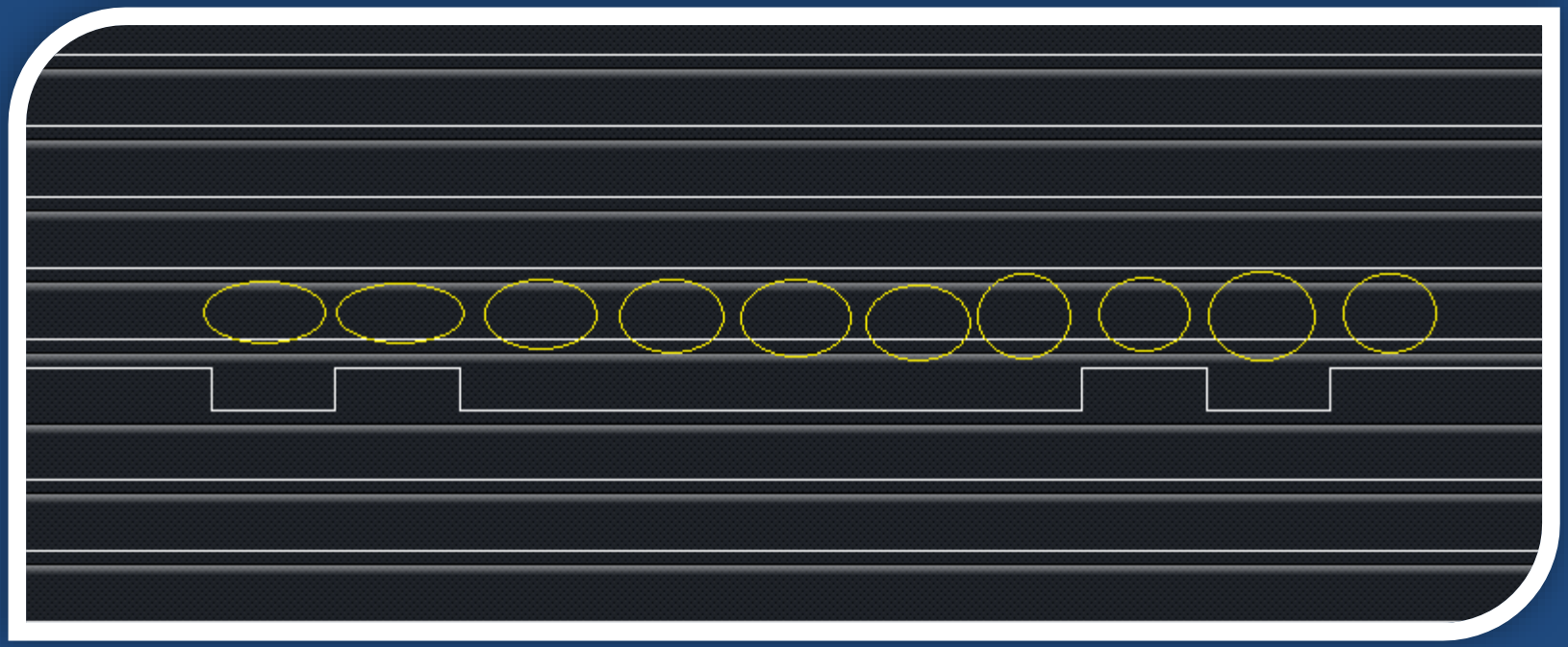
UART의 TX 핀 스니핑 결과

- 1초 간격의 신호 반응을 볼 수 있음
- 보이지 않으면 마우스휠(혹은 키보드 +/-)로 조정

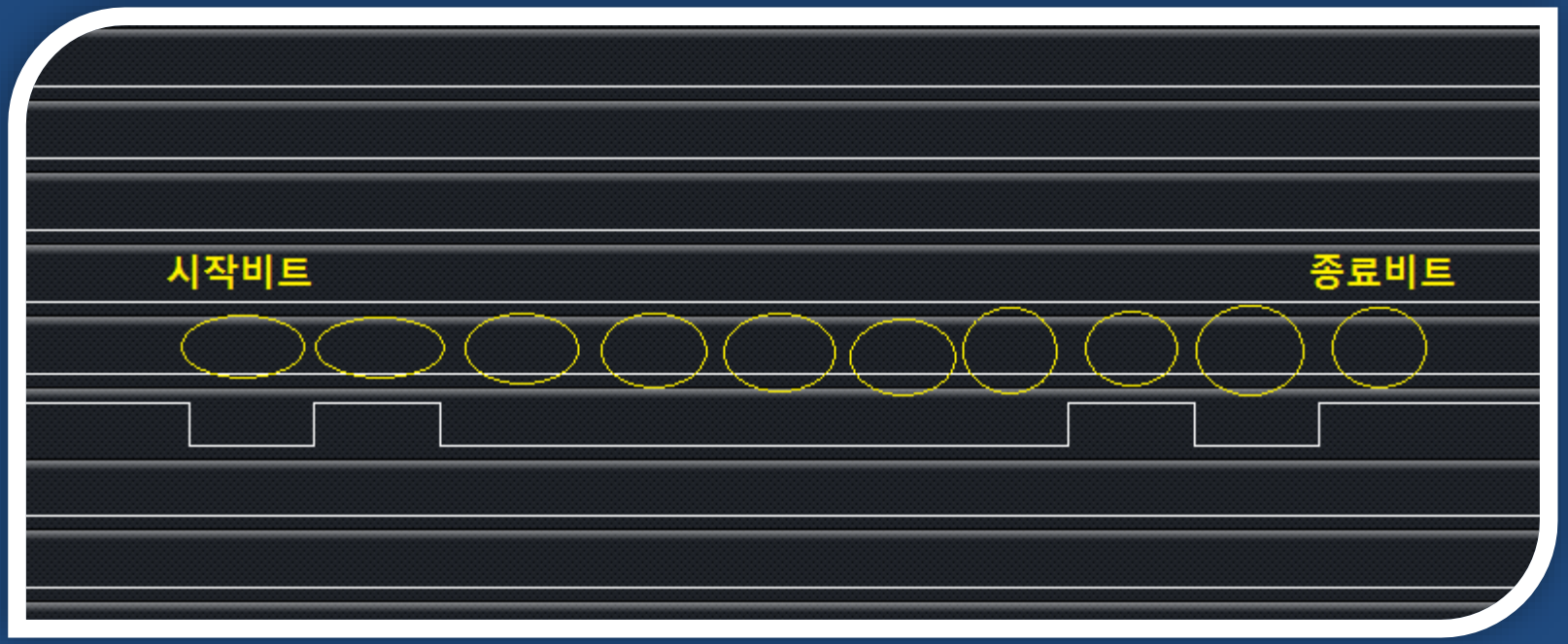


마우스휠을 이용하여 확대

신호를 비트 단위로 구분

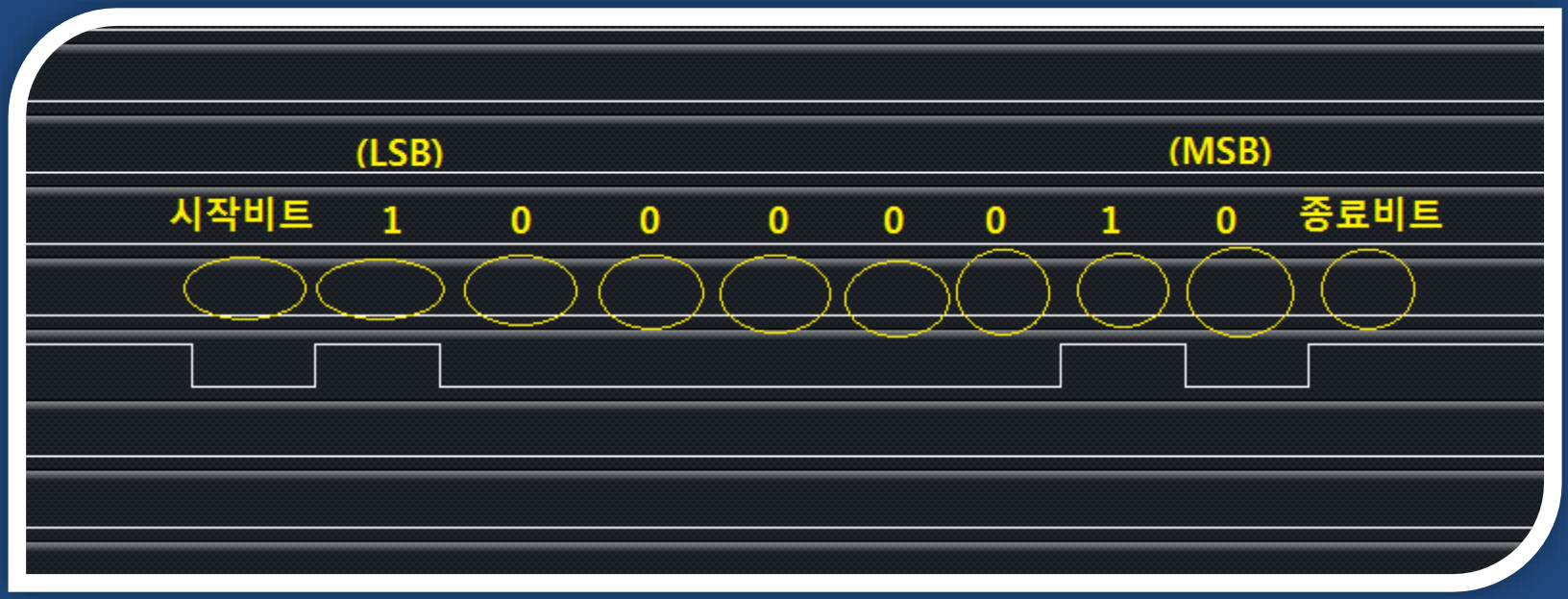


시작 비트와 종료 비트

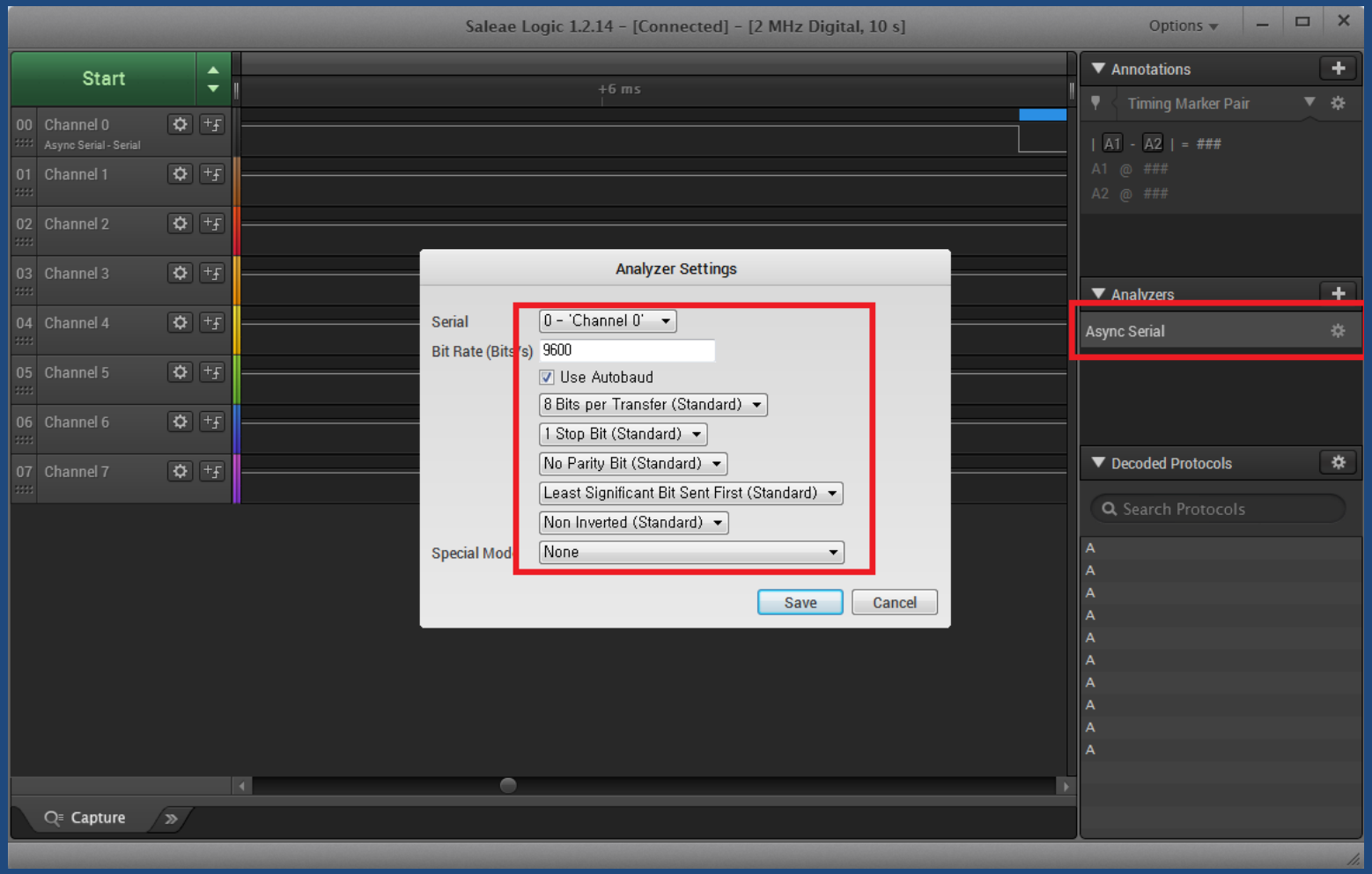


데이터 비트 분석

- 0b01000001 = 0x41 = ‘A’

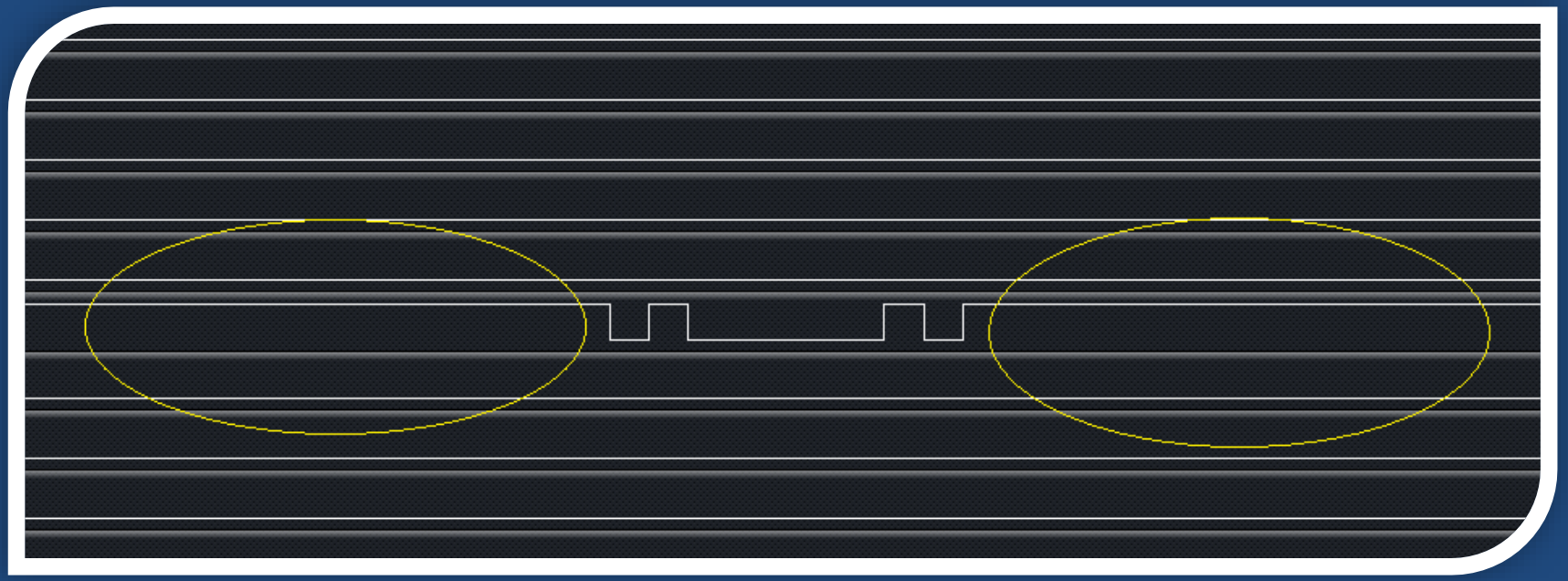


설정 변경이 필요한 경우


Saleae Logic 1.2.14 - [Connected] - [2 MHz Digital, 10 s]
Options
Start
+6 ms
00 Channel 0
Async Serial - Serial
01 Channel 1
02 Channel 2
03 Channel 3
04 Channel 4
05 Channel 5
06 Channel 6
07 Channel 7
Annotations
Timing Marker Pair
Analyzers
Async Serial
Decoded Protocols
Search Protocols
Analyzer Settings
Serial
0 - 'Channel 0'
Bit Rate (Bits/s)
9600
Use Autobaud
8 Bits per Transfer (Standard)
1 Stop Bit (Standard)
No Parity Bit (Standard)
Least Significant Bit Sent First (Standard)
Non Inverted (Standard)
Special Mod
None
Save
Cancel
Capture

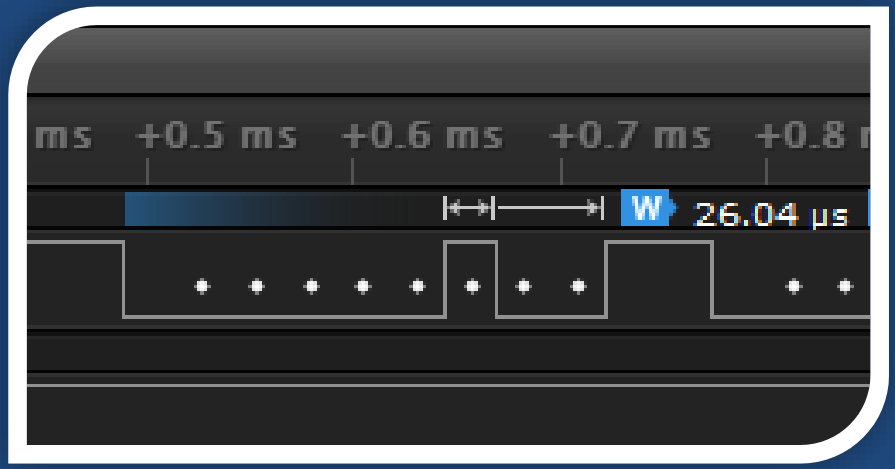
TX 핀의 기본 상태는 HIGH다.

- TX 핀과 GND에 LED를 연결했을 때 빛이 나는 이유



신호 분석을 통해 baudrate 알아내기

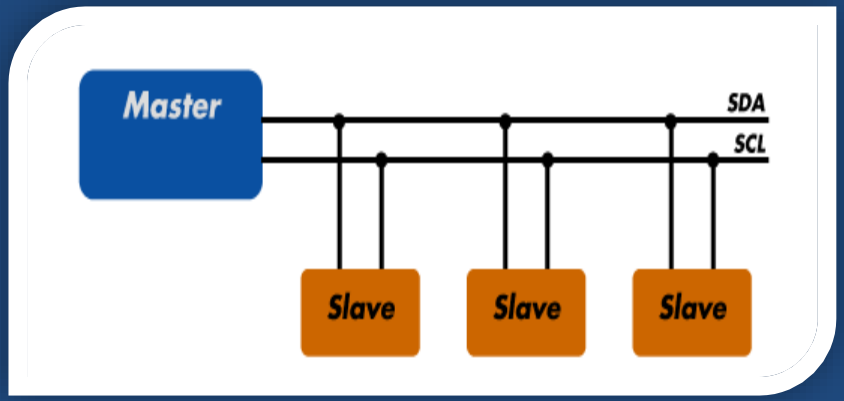
- 9600 : 100us
- 14400 : 69us
- 38400 : 26us
- 57600 : 17us
- 115200 : 8.6us



I2C 신호 분석 실습

I2C란?

- 하드웨어 통신 프로토콜의 한 종류
- TWI라고도 불리움
  - Two Wire Interface
- 2개의 Line으로 통신함
  - SDA(TWD) : DATA
  - SCK(TWCK) : CLOCK
- MASTER/SLAVE 구조로 작동
  - 하나의 MASTER에 여러 개의 SLAVE 연결 가능
  - ADDRESS를 통해 각 SLAVE를 구별
- 통신 속도
  - High speed 모드 3.4Mbps
  - Fast 모드 400Kbps
  - 표준 모드 100Kbps (arduino default)
  - 저속 모드 10Kbps

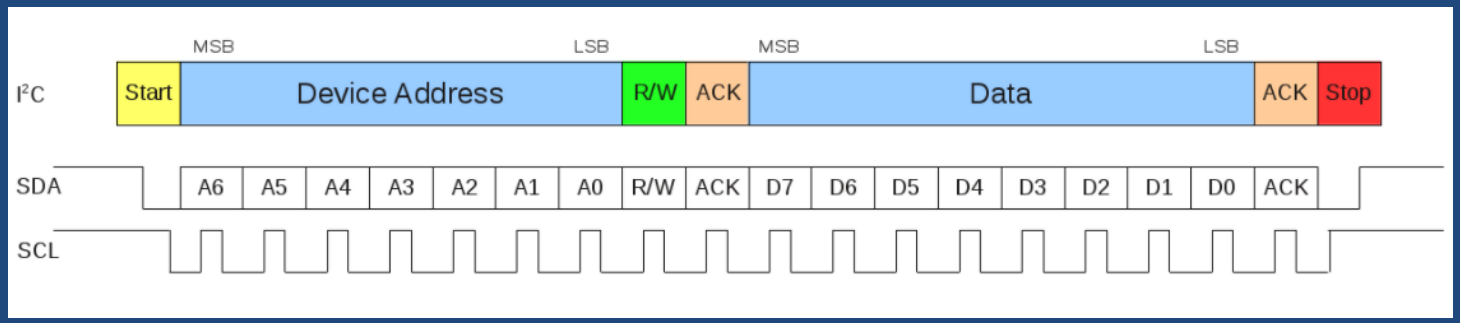


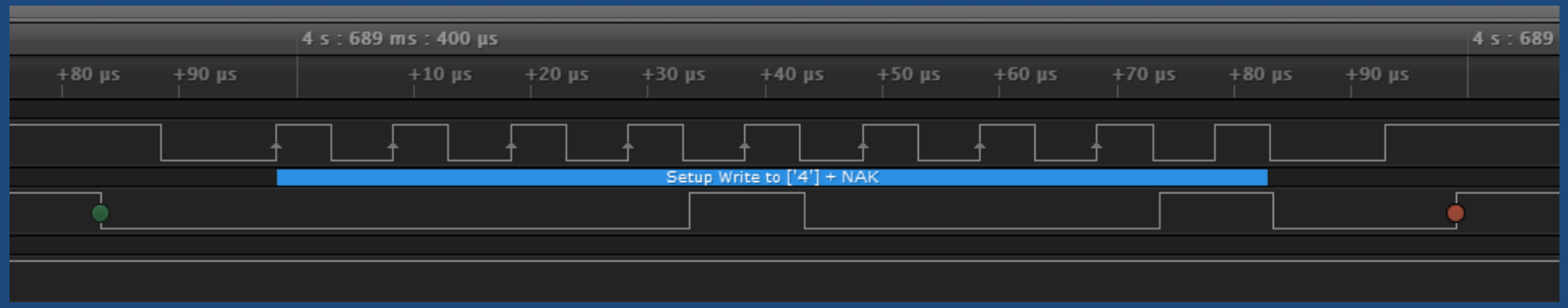
I2C 프로그래밍 (master)

```
#include <Wire.h>

#define SLAVE_ADDR 4 // 슬레이브의  주소

void setup() {
  Wire.begin(); // Wire 라이브러리 초기화
}

void loop() {
  Wire.beginTransmission(SLAVE_ADDR);
  Wire.write('A');
  Wire.endTransmission();
  delay(1000);
}
```

I2C 신호 분석 결과

- Address = 7bit (0000100)
- Mode = 1bit (READ = 1 , WRITE = 0)
- Response = 1bit (ACK = 0, NAK =1)





Address = 0000100(4), Write(1), NAK(1, 응답 없음)

결론 : 신호 분석의 장점

- 신호의 패턴을 보고 어떤 용도의 핀인지 파악할 수 있음
  - Ex> CLOCK핀, UART의 TX 핀
- 문제 발견 시 원인을 분석해낼 수 있음
  - EX> 신호 반전 현상
- Unknown 프로토콜을 분석해 낼 수 있음
- Sniffing을 통해 정보를 얻어낼 수 있음
  - EX> 암호화 KEY

감사합니다.