# 목차

[목차](#h.yqr2ub26pyp)

[General Information](#h.ssuil5ouylvc)

[회로 수정 사항](#h.wfbct0xm71lj)

[Arduino 개발 환경 구축](#h.f1wjlvbt47wv)

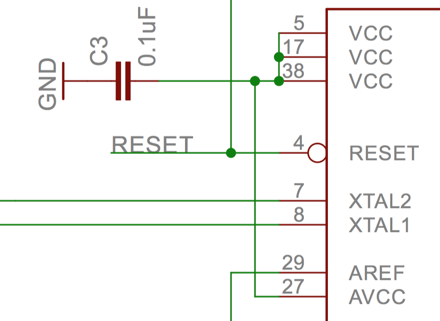
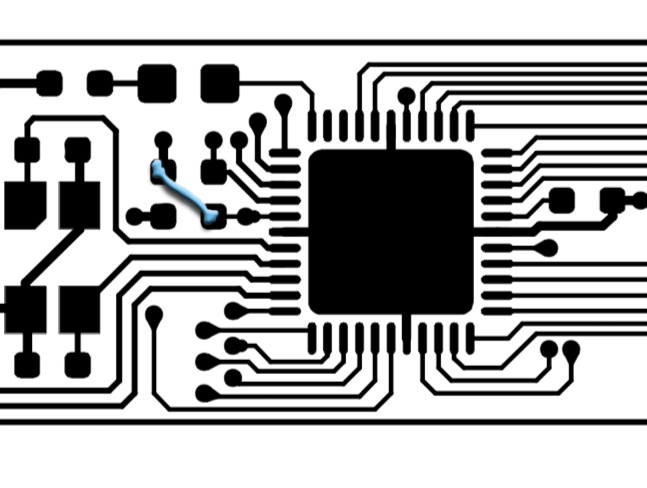
[JTAG disable 시키기](#h.o2sp9co2p1vh)

[터치 감지 회로 동작 원리](#h.6bus8umyktpn)

# General Information

* 회로도
  + schematic: Touch.pdf
  + drawing: gerber.pdf, touch\_circuit.ai
* 구현 파일
  + conductivity\_touch\_sensor\_22lines.zip: AVR쪽 firmware (arduino 환경)
  + conductivity\_touch\_sensor\_visualizer.zip: Serial 통신을 통해 터치 상태 보기 (processing 환경)
  + arduino\_sanguino\_variant: arduino 활용을 위한 arduino 설정 파일과 부트로더 등.

# 회로 수정 사항

* 문제 파트: VCC 연결 누락
* 패치 부분:
* 패치 사진:

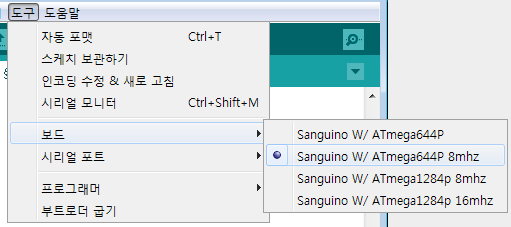
# Arduino 개발 환경 구축

* Sanguino (http://sanguino.cc/)을 터치 회로용으로 사용하기  
  (atmega644p, 8Mhz, 3.3v 으로 사용할 수 있도록..)
  + 참고: arduino\_sanguino\_variant.zip 에 반영되어 있음.
* WinAVR 깔아서 개발 환경 구축
* sanguino 을 arduino 1.0 이상의 ide, hardware/arduino 폴더에 덮어씌우기 (=기존 arduino 폴더 지우고 sanguino가 적용된 폴더로 적용)
* 수정사항
  + \hardware\arduino\bootloaders\atmega\Makefile 에 다음 항목 추가 (bold 표시가 주요 체크부)

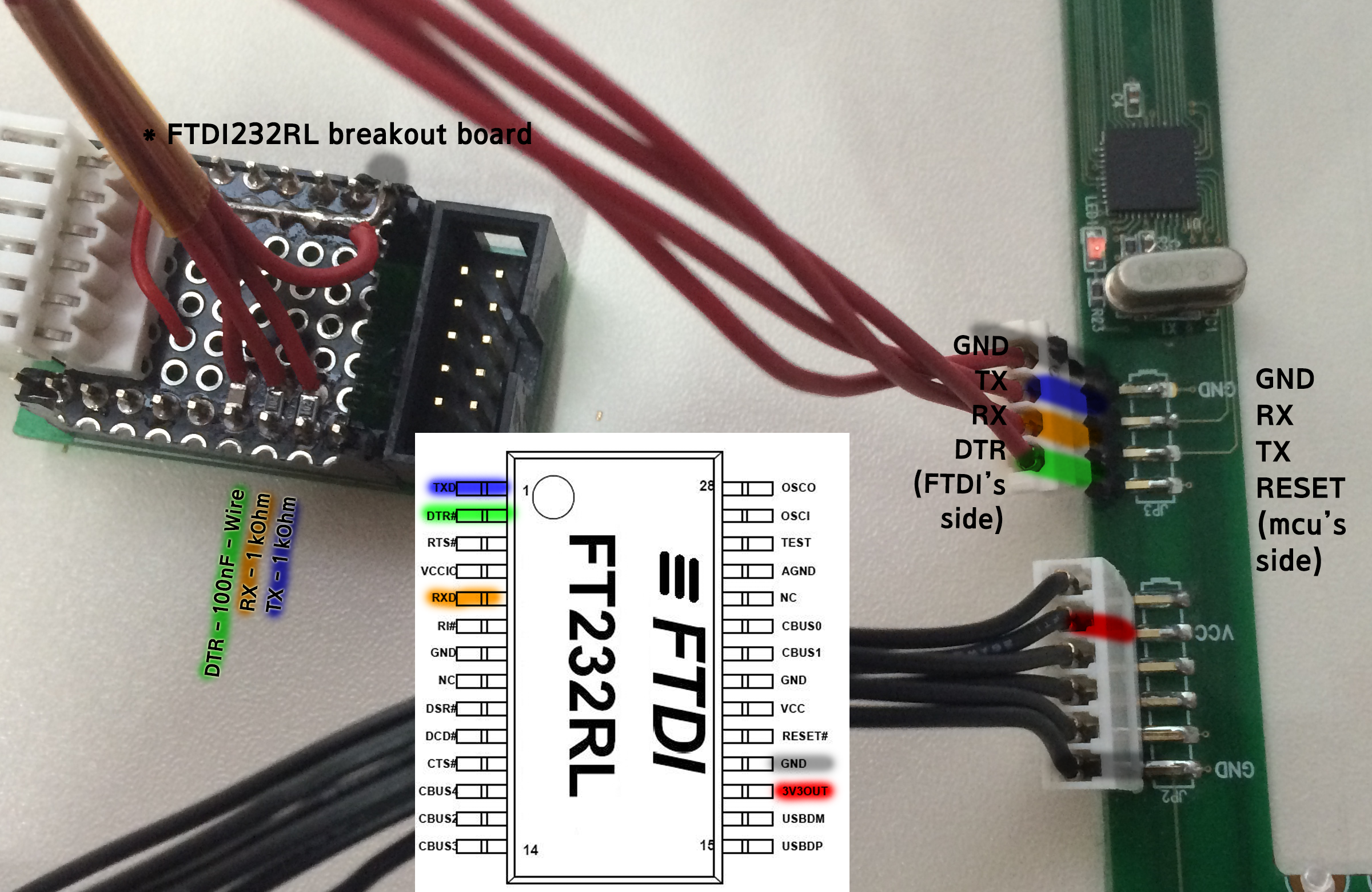
|  |
| --- |
| atmega644p\_8m: TARGET = atmega644p  atmega644p\_8m: MCU\_TARGET = atmega644p  atmega644p\_8m: CFLAGS += '-DMAX\_TIME\_COUNT=F\_CPU>>4' '-DNUM\_LED\_FLASHES=0' '-DBAUD\_RATE=**19200**'  atmega644p\_8m: AVR\_FREQ = **8000000L**  atmega644p\_8m: LDSECTION = --section-start=.text=0xF800  atmega644p\_8m: **$(PROGRAM)\_atmega644p\_8m.hex** |

* + \hardware\arduino\bootloaders\atmega 폴더에서   
    **make atmega644p\_8m**  
    명령을 수행하여, ATmegaBOOT\_168\_atmega644p\_8m.hex 파일 생성 확인
  + \hardware\arduino\boards.txt 에 다음 항목 추가 (bold 표시가 주요 체크부)

|  |
| --- |
| #########################################################  atmega6448m.name=Sanguino W/ ATmega644P 8mhz  atmega6448m.upload.protocol=stk500  atmega6448m.upload.maximum\_size=63488  atmega6448m.upload.speed=**19200**  atmega6448m.bootloader.low\_fuses=**0xFD**  atmega6448m.bootloader.high\_fuses=**0x9A**  atmega6448m.bootloader.extended\_fuses=**0xFF**  atmega6448m.bootloader.path=atmega  atmega6448m.bootloader.file=ATmegaBOOT\_168\_atmega644p\_8m.hex  atmega6448m.bootloader.unlock\_bits=0x3F  atmega6448m.bootloader.lock\_bits=0x0F  atmega6448m.build.mcu=atmega644p  atmega6448m.build.f\_cpu=**8000000L**  atmega6448m.build.core=arduino  atmega6448m.build.variant=standard |

* 수정된 arduino를 기동하여 다음 사항 확인
  + 
  + AVR ISP를 연결 후, 도구 => Bootloader 굽기를 통해 부트로더 로딩하면, arduino 환경에서 개발할 수 있음.
  + 참고로, NEWTC의 AD-USBISP V6.0 을 사용할 경우 기본 avrisp를 선택하면 에러가 나는데,   
    \hardware\arduino\programmers.txt  
    에 다음 라인을 추가하면 사용할 수 있음.

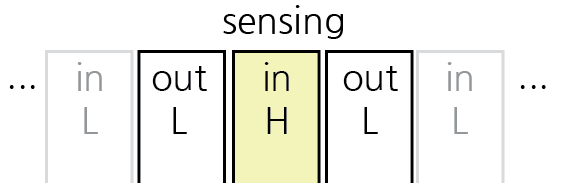
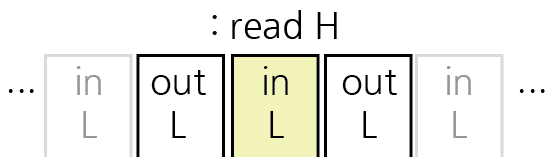
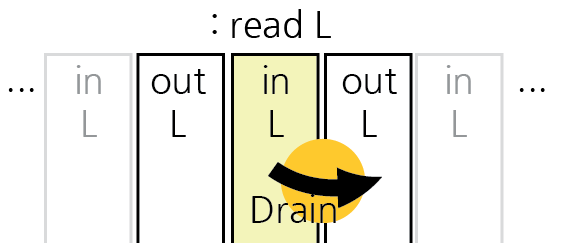
|  |
| --- |
| avrisp\_newtec.name=AD-USBISP V6  avrisp\_newtec.communication=serial  avrisp\_newtec.protocol=stk500v2 |

* Arduino interface용 FT232RL 회로
  + 

# JTAG disable 시키기

|  |
| --- |
| // Disable JTAG. \* need to command twice.  MCUCR|= (1<<JTD); MCUCR|= (1<<JTD); |

# 터치 감지 회로 동작 원리

* 관련 파일
  + conductivity\_touch\_sensor\_22lines.zip
  + conductivity\_touch\_sensor\_visualizer.zip
* 동작 설명
  + 초기 세팅  
    
    - 센싱 라인을 input으로 설정 & pull up (AVR’s internal pull-up through 20k~50k resister)
      * +- VCC  
         |  
         pull-up R (20~50k)  
         |  
        PIN ----+--- strip (5~10k) --- resister (5M) ---- GND
    - 주위의 두 라인을 output LOW 으로 설정
    - 위의 세 라인을 제외한 나머지 라인은 input & low 상태로 설정 (internal pull-up off = high impedance)
  + 손가락 센싱 start  
    
    - 센서 핀의 internal pull-up을 해제 (= digitalWrite(LOW) )
    - 수 microsecond 기다림 (현재 30us)
  + 센서 값 읽음
    - HIGH: 손가락이 터치되지 않았을 경우, 터치 strip의 자체 capacitance 때문에, HIGH 값이 출력 (= low로 drain 되지 않음)  
      
      * PIN(H) -------- strip (5~10k) --- resister (5M) ---- GND  
         ======= Drain (slow) ======>
    - LOW: 손가락이 터치되었을 경우, 손가락이 센서 strip과 주변 strip 을 동시에 터치하여 circuit이 형성될 경우, High 상태에서의 voltage가 주변 out low 상태로 설정된 strip들로 drain되면서, low 값이 출력  
      
      * +- GND   
         |   
         Finger (≈5M) ⇑ => Drain(fast)  
         |   
        PIN(H) ----+--- strip (5~10k) --- resister (5M) ---- GND