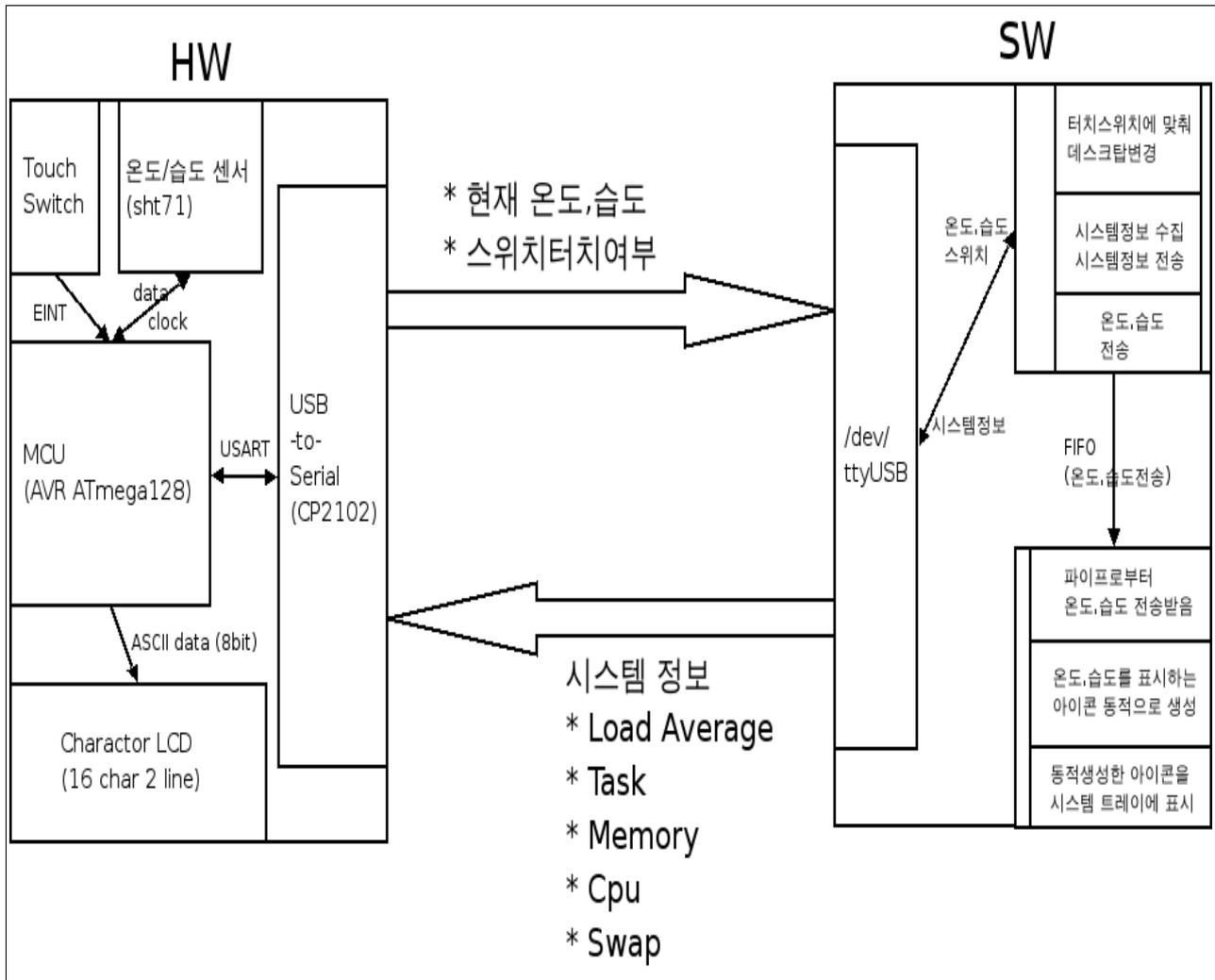


# System Overview



## \* HW

하드웨어는 크게 5 부분으로 나누어 집니다. 그중 사용자와 인터페이스 하는 부분은 터치스위치, 온습도센서, Charactor LCD 세 부분이고, usb-to-serial 은 PC 와의 통신을 위한 부분입니다. MCU 로는 AVR ATmega128 을 사용하여 avr-gcc 를 이용하여 펌웨어를 컴파일 하였습니다.

터치스위치 회로는 AVR 의 외부인터럽트에 연결하여, 터치스위치에서 신호가 발생하면 AVR 에서는 외부인터럽트가 발생합니다. 인터럽트 핸들러에서는 몇번인터럽트가 발생했는지를 USART 를 통해 통신부(cp2102) 로 전송하고, cp2102 는 usb 를 통해 PC 로 정보를 전송합니다.

온도습도센서(sht71) 은 두개의 AVR 핀에 연결되고 하나는 데이터선, 하나는 SCK선으로 이용합니다. 온습도 센서를 통해 받은 데이터는 마찬가지로 cp2102 를 통해 PC 로 전송합니다.

Charactor LCD 는 16 character 2 line LCD 를 사용하였습니다. PC 측 프로그램은 시스템 정보를 제가 정한 프로토콜에 맞추어 USB 를 통해 전송합니다. AVR 은 USB 를 통해 받은 데이터를 CLCD 로 보내고, CLCD 에서는 이를 화면에 표시합니다.

로드애버리지는 CLCD 의 윗줄에 항상 표시되며, 실시간으로 갱신됩니다. 그외 정보들은 CLCD 의 아랫줄에 옆으로 스크롤 되어 표시됩니다.

\* SW

PC 쪽 프로그램은 모두 리눅스에서 구현 하였습니다. 전체적으로 두개의 프로그램으로 나누어져 있습니다. 하나는 USB 로 부터 데이터를 받아, 윈도우매니저의 데스크탑을 전환하고, 시스템 정보를 USB 를 통해 AVR 로 전송하는 프로그램이고, 다른 하나는 IPC(FIFO) 로 연동 되어 시스템 트레이에 아이콘을 만들어주는 프로그램입니다.

cp2102 는 usb-to-serial 칩으로 PC 에서는 일반적인 시리얼포트 처럼 사용하는 식으로 동작합니다. windows 에서는 COM 포트 로 인식되고, 리눅스에서는 ttyUSB 로 인식됩니다. 이를 사용하는 프로그램에서도 마찬가지로 terminal 프로그램을 구현하고, 장치파일로 ttyUSB 를 선택하면 됩니다.

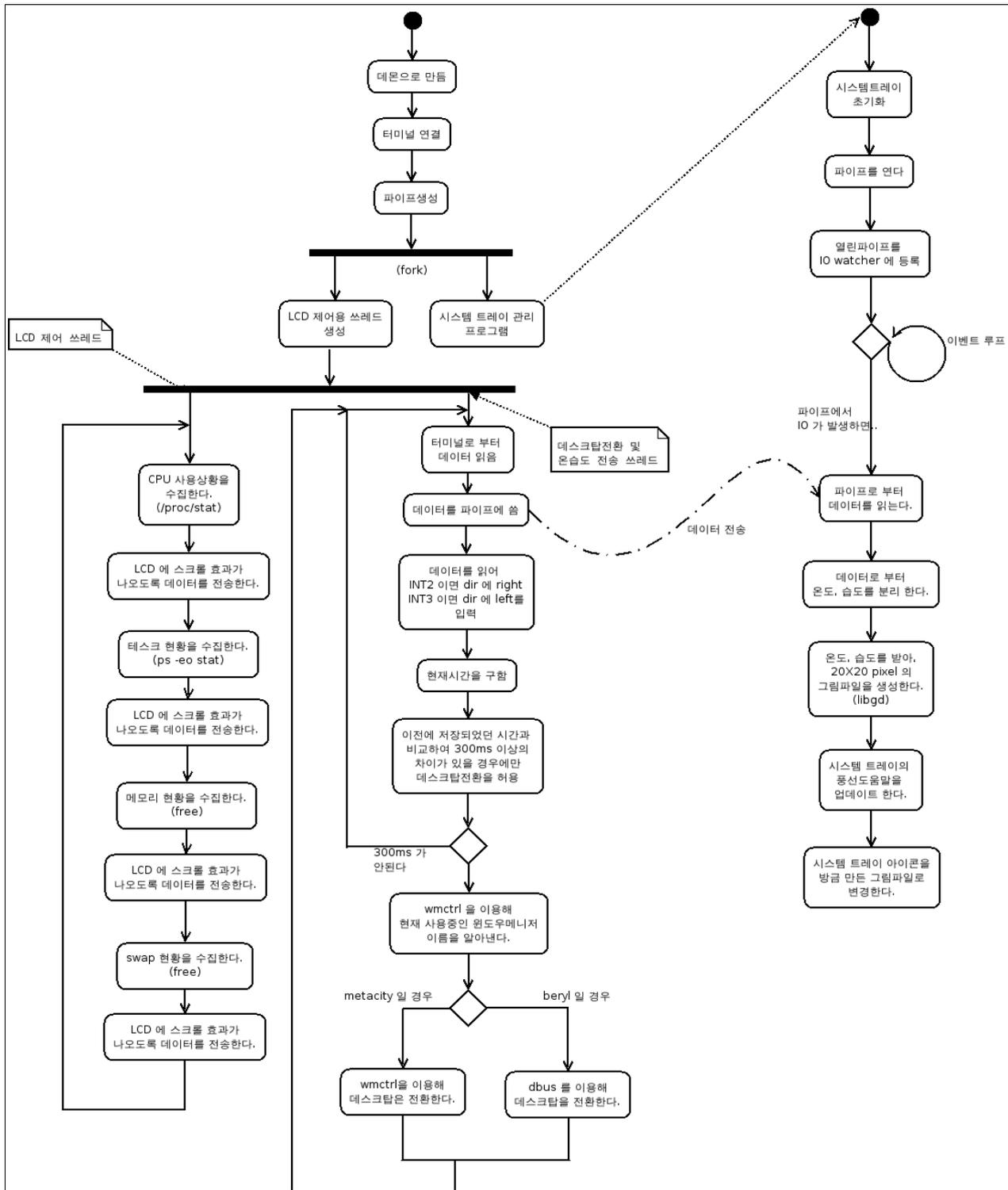
두개의 프로그램을 각각 메인프로그램, 서브프로그램으로 명명하겠습니다. 메인프로그램은 위에서 설명한 ttyUSB를 통해 AVR 과 데이터를 주고 받습니다. 받는 데이터는 터치스위치가 눌렸을 때 어떤 스위치가 눌렸는가(EINT2, EINT3), 그리고 현재 온도, 습도 데이터 입니다. 보내는 데이터는 현재 시스템의 정보들(cpu 사용량, 메모리 사용량, 로드에버리지등..) 입니다.

EINT2의 신호가 올 경우에는 윈도우매니저의 데스크탑을 왼쪽으로 전환하고, EINT3이 올 경우에는 오른쪽으로 전환 합니다. metacity, kwin, wmaker 등 EWMH(Extended Window Manager Hints) 스펙을 지원하는 윈도우 매니저는 wmctrl 이라는 외부 프로그램을 사용하여 데스크탑을 전환 하고, 3D 데스크탑 환경을 만들어주는 beryl 은 직접 dbus 명령을 만들어 dbus-send 라는 외부 프로그램을 사용하여 데스크탑을 전환 합니다.

시스템 정보는 ps 를 통해, 현재 프로세스의 통계(전체 프로세스갯수, 동작중인 프로세스, sleep 상태, zombie 상태)를 계산해, 만들어 내고, free 명령으로 메모리 사용량, proc/stat 파일을 분석하여, cpu 사용량등을 만듭니다. proc/loadavg 파일에서는 로드에버리지 정보를 받아 ttyUSB 를 통해 AVR 에 전송합니다. AVR 에서는 이 데이터를 CLCD 로 보내서 외부 LCD 에 시스템 정보가 출력 됩니다.

온,습도 센서를 통해 들어온 정보는 메인프로그램에서 IPC(FIFO) 를 이용해 서브 프로그램으로 전달 합니다. 서브프로그램은 python 으로 작성되어 있습니다. FIFO 를 통해 받은 온도,습도 정보를 서브프로그램에서는 pyGD 라이브러리를 통해, 20X20 의 작은 아이콘으로 만들고, 만들어진 아이콘은 pyGTK 를 이용해서 system tray 에 등록 됩니다. FIFO 를 통해 데이터가 갱신될 때 마다, system tray 에 아이콘도 실시간으로 변경됩니다.

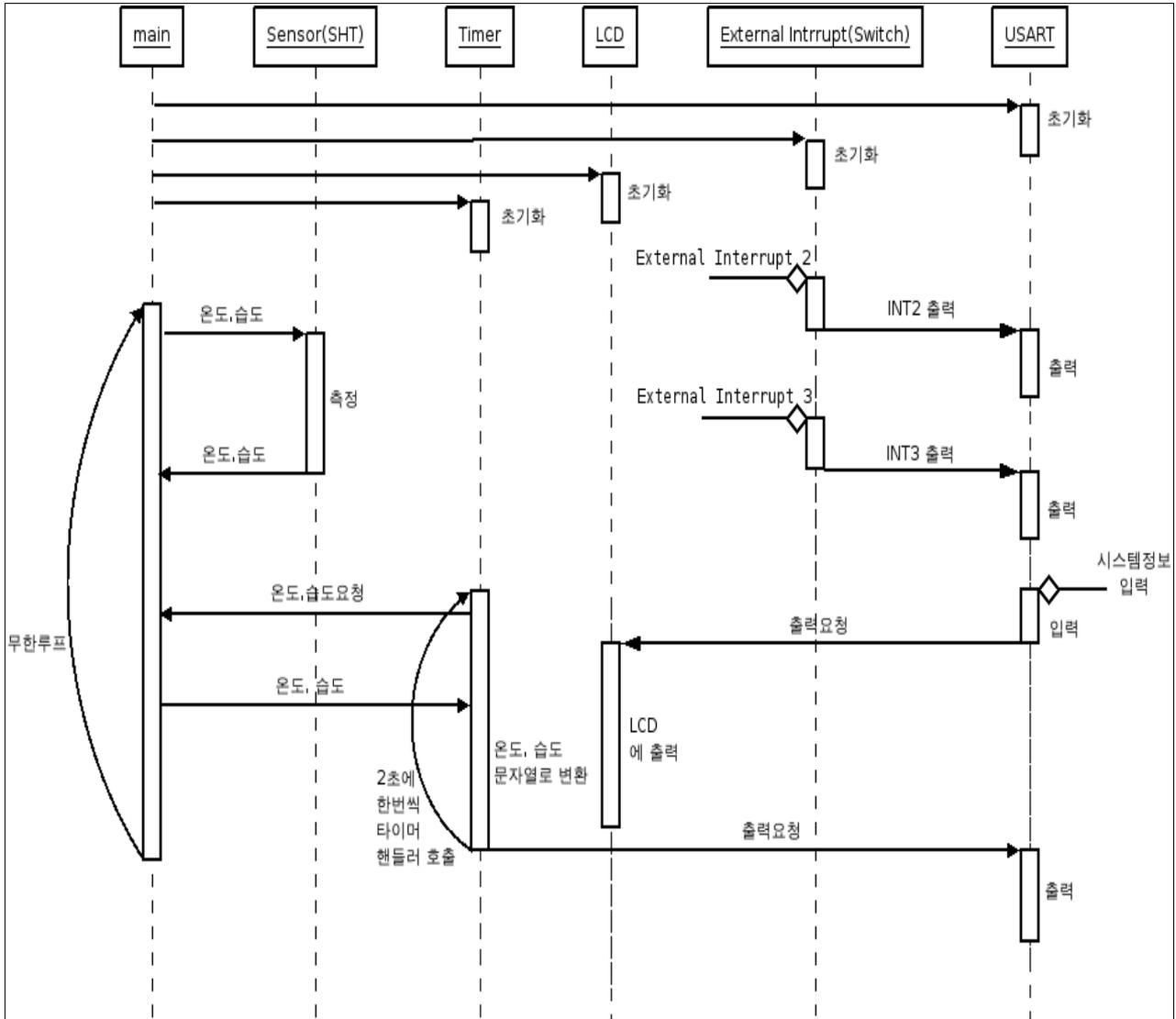
# Software Overview



software 의 flowchart 입니다. 메인 프로그램은 두개의 쓰레드로 나뉘어, 한 쓰레드는 윈도우 매니저 전환을 담당하고 다른 한 쓰레드는 CLCD 제어를 담당 합니다.

또한 다른 프로세스를 fork 하여, FIFO 로 통신을 합니다. fork 된 서브프로그램은 메인프로그램으로 부터 FIFO 로 온도, 습도를 받아, system tray 에 온도, 습도를 표시하는 아이콘을 만듭니다.

# Firmware Overview



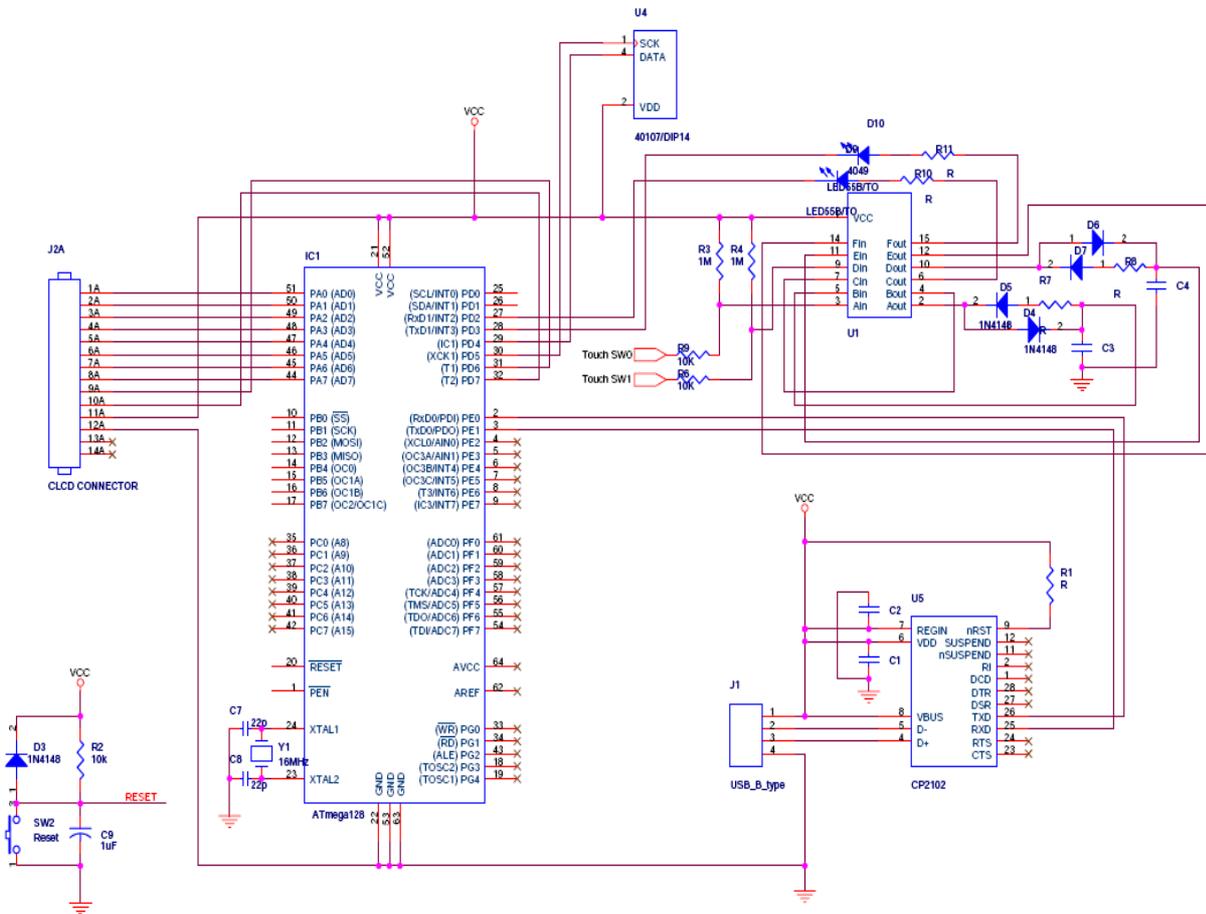
펌웨어의 동작을 시퀀스 다이어그램의 형식을 빌어 그린 그림입니다.  
AVR 에 전원이 들어오면 일단 하드웨어를 구성하는 각 부분을 초기화 합니다.

이후에 하드웨어 각 부분들은 시그널 핸들러를 통해, 제어됩니다. 온,습도 센서(sht71) 은 타이머 핸들러를 통해 제어 되어 2초에 한번씩, 측정된 결과를 USART 를 통해 PC 로 전송합니다.

터치스위치는 스위치에 접촉이 있을 때 마다, External Interrupt 가 발생하여, EINT 시그널 핸들러가 동작하게 됩니다. EINT 시그널 핸들러는 각각 EINT 번호에 따라 INT2, INT3 이라는 문자열을 USART 를 통해 PC 로 전송 합니다.

USART 를 통해 데이터가 들어오면 이를 버퍼에 모았다가 32바이트가 되었을 때, CLCD로 전송합니다. 전송된 데이터는 CLCD 에 ascii 문자열 형태로 출력 됩니다.

# Schematics



하드웨어 회로도 입니다. 구현과 관계없는 추가적인 회로는 빼고 작품과 관련된 회로만 포함 하였습니다.

하드웨어의 전체 전원은 모드 USB 를 통해 받게 하였습니다. 그래서, 다른 외부 전선 없이 USB 케이블만 꼽으면 모든 하드웨어 장치가 동작합니다.

우측 상단 U1 이 4049 이고, 관련된 회로가 터치스위치 회로입니다. 4049 IC 는 not gate 가 들어가 있는 hex buffer 로 평소에는 전원에서 full up 되어 1 이 흘러 4049 를 거치면 0 이 출력되므로 스위치가 떨어져 있는 것으로 인식합니다. 하지만 사람 몸이 스위치 역할을 하는 철판(혹은 어떤 종류의 도체) 에 닿게 되면, 사람몸이 저항이 되어 전류는 접지되어 땅으로 흘러가 버리므로 0이 흐르게 되고 4049 를 거치면 1이 출력되어 스위치가 ON 된것으로 인식됩니다. AVR 의 EINT 핀으로 들어가기 전에 LED 를 켜게 되어 눈으로 확인 할 수도 있습니다.

CP2102 는 데이터시트를 보고 최소한의 핀만 연결하였습니다. USB 의 D+/- 과 AVR 의 TXD, RXD 를 연결해주는 칩입니다. 이 칩을 사용하여, 펌웨어나 PC 쪽 소프트웨어에서나 별다른 작업없이 일반적인 시리얼 프로그램만으로 통신이 가능합니다.

중앙상단이 온습도센서(sht71) 을 연결한 회로 입니다. 간단하게 AVR 과는 두개의 선이 연결되어 있습니다. 하나는 data 선이고 하나는 clock 선입니다.

좌측이 character LCD 로 연결되는 회로 입니다. 데이터선은 AVR 의 PA0~7 까지의 8bit 를 사용합니다.

Appendix A. PC 부 제어 소프트웨어 소스 - C

Appendix B. PC 부 시스템트레이 아이콘 소스 - Python

Appendix C. AVR 펌웨어 소스 - C

[별첨]