

프로젝트

Solenoid_Braille

- 시각 장애인을 위한 점자 장치 -

- 강민성 , 김동주 -

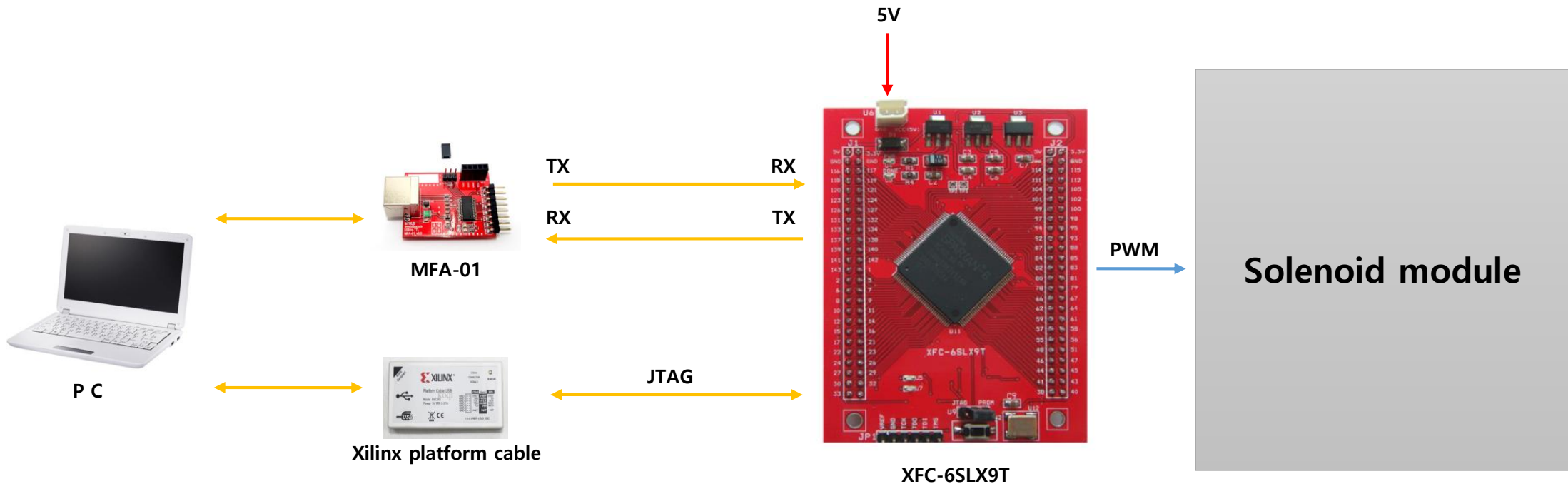
진행 상황(4.22 ~ 4.26)

- BOM (Bill Of Material)

| 품 번 | 품 명 | 구 분 | 수 량 | 단 가 (원) | 총 액 (원) |
|-----|--------------------------|---------------------------|-------|---------|---------|
| 1 | FPGA 베이직 모듈 | XFC-6SLX9T | 1 EA | 68,000 | 68,000 |
| 2 | USB to UART TTL 모듈 | MFA-01 | 1 EA | 15,000 | 15,000 |
| 3 | Xilinx-ic FPGA Cable USB | Xilinx platform USB cable | 1 EA | 46,020 | 46,020 |
| 4 | 51004-0200 | Crimp housing | 5 EA | 230 | 1,150 |
| 5 | SMPS 아답터 DC 5V | AC/DC Adaptor | 1 EA | 7,535 | 7,535 |
| 6 | 50011-8000 | Crimp Terminal | 20 EA | 19.8 | 396 |
| 7 | YYT-11 | 탭커넥터 크리핑툴 | 1 EA | 33,000 | 33,000 |
| 8 | DC-005 | DC Power Jack | 3 EA | 99 | 297 |
| 9 | FH01(2.54)-DS20P | 핀헤더소켓 | 2 EA | 225 | 550 |
| 10 | PH01(2.54)-SS40P | 핀헤더 | 2 EA | 159.5 | 319 |
| | | | | 합 계 | 17,2267 |

진행 상황(4.22 ~ 4.26)

- 연결 구성도



진행 상황(4.22 ~ 4.26)

- ISE 코딩 설정

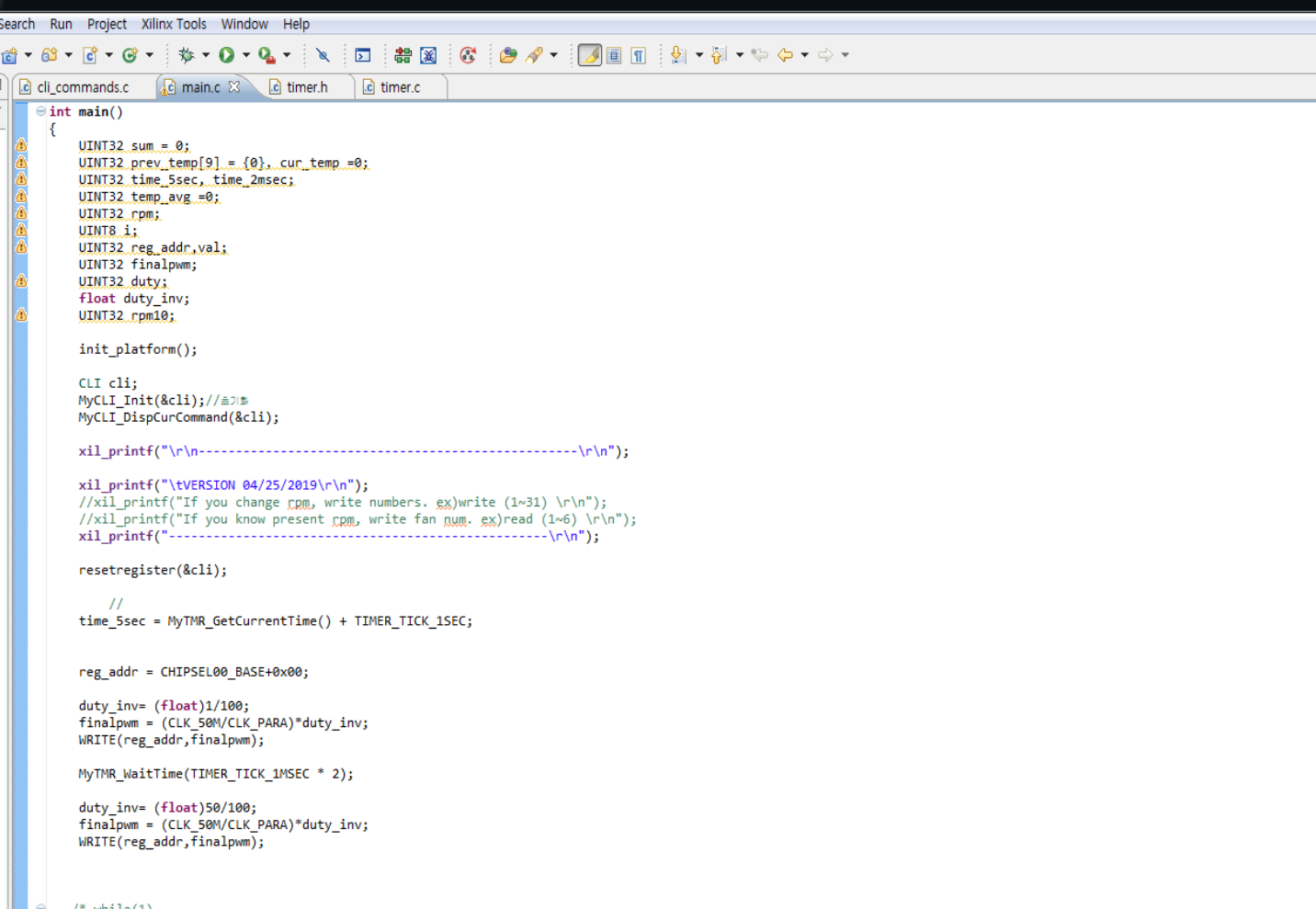
- 솔레노이드 동작에 필요한 PWM 파형
- 필요한 duty 값을 SDK로 부터 불러 받아와서 카운터를 이용하여 PWM 파형을 생성한다.

```
23
24 attribute keep : string;
25 attribute keep of PULSE_TEM : signal is "true";
26
27 begin
28
29 rpm_spd_proc : process(fan_clk)
30 begin
31     if (fan_reset='0') then
32         PULSE_TEM <= (others=>'0');
33         COUNT_CLK <= (others=>'0');
34     elsif rising_edge(fan_clk) then
35         if (fan_wren='1') then --FAN입력 값이 들어오면
36             PULSE_TEM <= (others=>'0');
37             COUNT_CLK <= (others=>'0');
38         else
39             COUNT_CLK <= COUNT_CLK + '1';
40             if (COUNT_CLK>=fan_in) then
41                 PULSE_TEM <= (others=>'0');
42                 if (COUNT_CLK= BASICHZ) then
43                     COUNT_CLK <= (others=>'0');
44
45                 end if;
46             elsif (COUNT_CLK< fan_in) then
47                 PULSE_TEM <= (others=>'1');
48             elsif (fan_pwr_sw_in='1') then --전면 스위치 때문에 0에서 1로 바꿈
49                 PULSE_TEM <= (others=>'0');
50             end if;
51         end if;
52         fan_out(0) <= not PULSE_TEM(0); --pulse임의 값 out으로 넘김
53         fan_out(1) <= not PULSE_TEM(1); --pulse임의 값 out으로 넘김
54         fan_out(2) <= not PULSE_TEM(2); --pulse임의 값 out으로 넘김
55     end if;
56 end process;
57
58 end Behavioral;
59
```

진행 상황(4.22 ~ 4.26)

- SDK 설정

- 오실레이터가 생성하는 50M CLK을 이용
- 기준 5KHZ를 이용하여 공식화 하여 duty비를 레지스터로 넘겨준다.



```

Search Run Project Xilinx Tools Window Help
cli_commands.c main.c timer.h timer.c
int main()
{
    UINT32 sum = 0;
    UINT32 prev_temp[9] = {0}, cur_temp = 0;
    UINT32 time_5sec, time_2msec;
    UINT32 temp_avg = 0;
    UINT32 rpm;
    UINT8 i;
    UINT32 reg_addr, val;
    UINT32 finalpwm;
    UINT32 duty;
    float duty_inv;
    UINT32 rpm10;

    init_platform();

    CLI cli;
    MyCLI_Init(&cli); // 添加
    MyCLI_DispatchCommand(&cli);

    xil_printf("\r\n-----\r\n");

    xil_printf("\tVERSION 04/25/2019\r\n");
    //xil_printf("If you change rpm, write numbers. ex)write (1~31) \r\n");
    //xil_printf("If you know present rpm, write fan num. ex)read (1~6) \r\n");
    xil_printf("-----\r\n");

    resetregister(&cli);

    //
    time_5sec = MyTMR_GetCurrentTime() + TIMER_TICK_1SEC;

    reg_addr = CHIPSEL00_BASE+0x00;

    duty_inv = (float)1/100;
    finalpwm = (CLK_50M/CLK_PARA)*duty_inv;
    WRITE(reg_addr, finalpwm);

    MyTMR_WaitTime(TIMER_TICK_1MSEC * 2);

    duty_inv = (float)50/100;
    finalpwm = (CLK_50M/CLK_PARA)*duty_inv;
    WRITE(reg_addr, finalpwm);

    /* while(1)

```

진행 상황(4.22 ~ 4.26)

- IMPACT 프로그램

- Xilinx 사 Spartan6 보드와의 Jtag 연결 확인

- 프로그램 성공 확인 완료

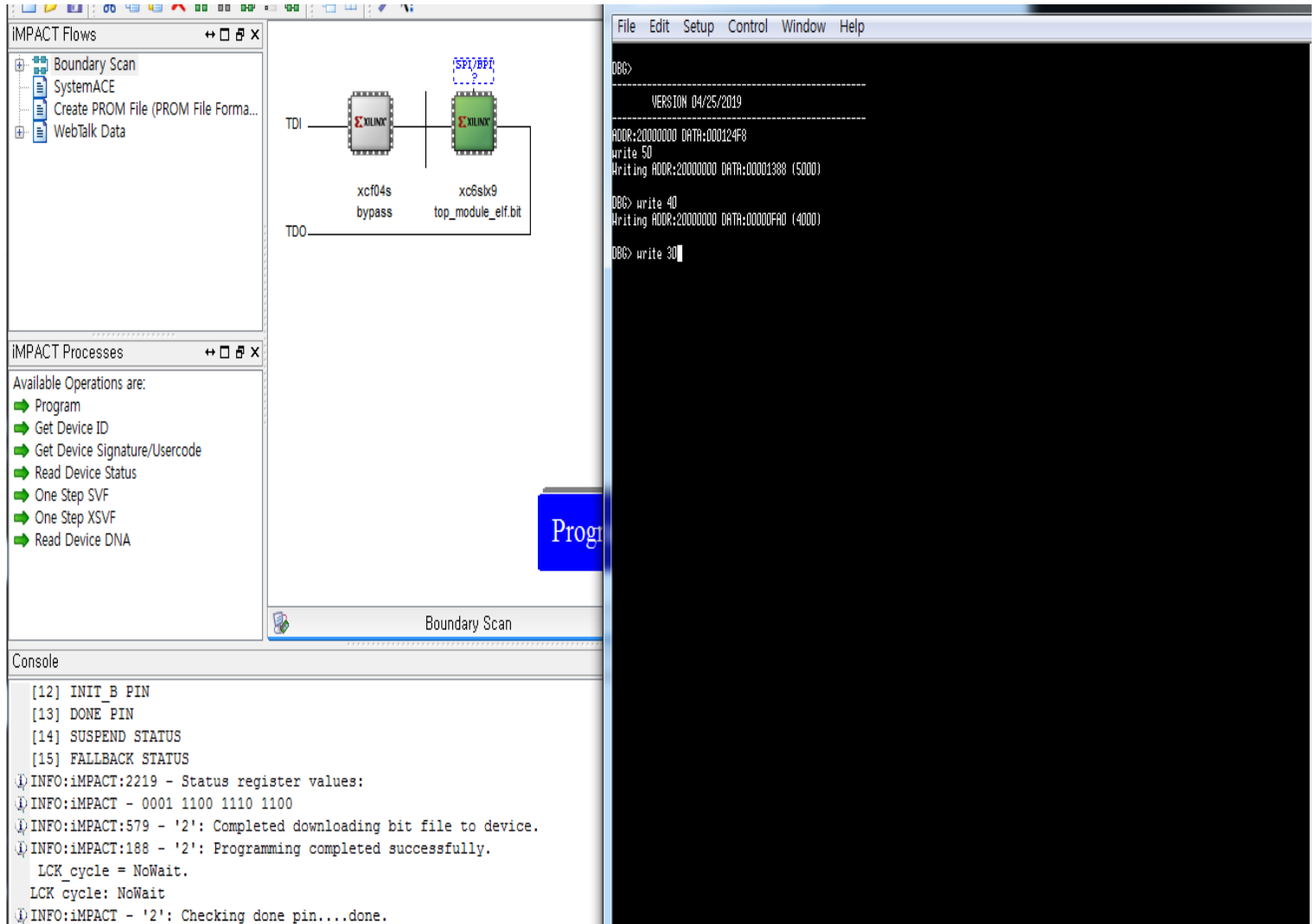
The image shows the Xilinx IMPACT software interface. On the left, the 'IMPACT Flows' pane lists 'Boundary Scan', 'SystemACE', 'Create PROM File (PROM File Forma...', and 'WebTalk Data'. Below it, the 'IMPACT Processes' pane lists available operations: 'Program', 'Get Device ID', 'Get Device Signature/Usercode', 'Read Device Status', 'One Step SVF', 'One Step XSVF', and 'Read Device DNA'. The main workspace displays a schematic diagram of two Xilinx devices connected via JTAG. The first device is labeled 'xc604s bypass' and the second is 'xc6sbx9 top_module_elf.bit'. A blue box with the text 'Program Succeeded' is overlaid on the right side of the workspace. At the bottom, the 'Console' pane shows the following log output:

```
[12] INIT_B_PIN : 1
[13] DONE_PIN : 1
[14] SUSPEND STATUS : 0
[15] FALLBACK STATUS : 0
INFO:IMPACT:2219 - Status register values:
INFO:IMPACT - 0001 1100 1110 1100
INFO:IMPACT:579 - '2': Completed downloading bit file to device.
INFO:IMPACT:188 - '2': Programming completed successfully.
LCK_cycle = NoWait.
LCK_cycle: NoWait
INFO:IMPACT - '2': Checking done pin....done.
'2': Programmed successfully.
```

진행 상황(4.22 ~ 4.26)

- Terminal 확인 (Tera Term)

- Tera Term을 통한 duty비 입력
- duty비 입력에 따른 올바른 데이터 값 출력 확인

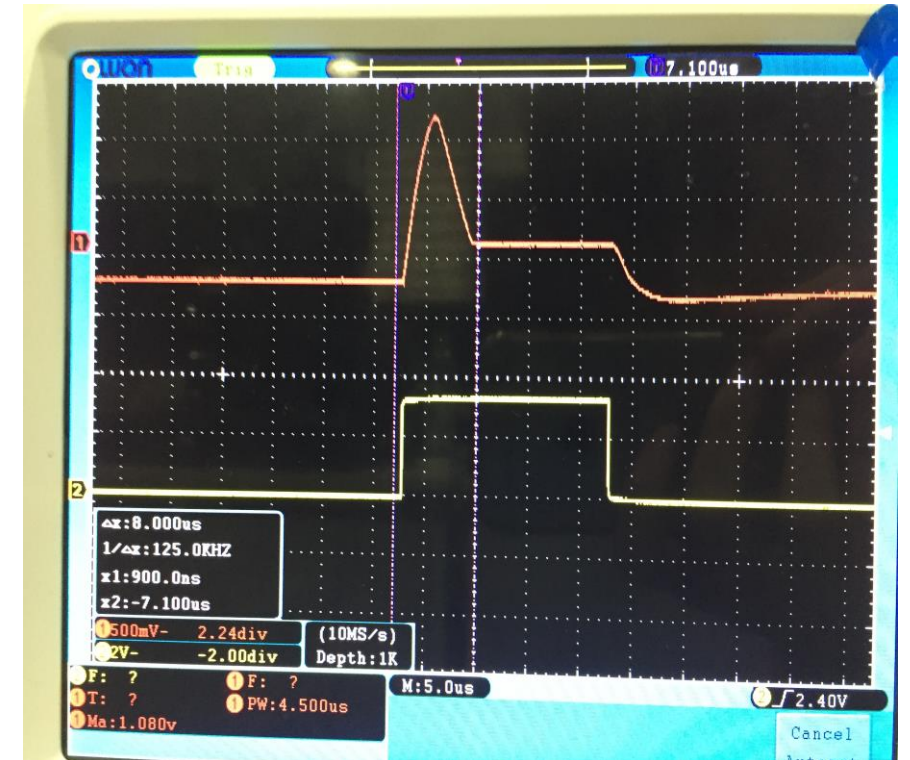


진행 상황(4.22 ~ 4.26)

- PWM 파형 측정

Period : 200us

Duty Cycle : 10%



정리 및 문제점

- PWM을 사용하여 코일을 자화 상태로 만드는 것은 성공 하였으나 코일에 코어가 없으면 자기 포화 상태를 만들수 없어 코어를 구해야하는 상황.
- PWM 파형 측정 시 FET의 Drain 파형이 PWM 파형과 다르게 전압 파형이 나오는 것을 볼수 있는데 이 파형을 해석을 해야 하고 해석 후 회로도 변경이 필요함.
- 코일의 역기전력 때문에 오버쇼트가 생기게 되는데 오버쇼트를 잡기 위해서 적분기가 필요하기 때문에 이부분에 대한 계산 및 회로도 변경이 필요함.
- 코일에 아직 코어가 없는 상태로 실험을 진행하였기 때문에 코어를 장착 후에 PWM의 Period 값과 Duty Cycle을 조율해 줄 필요가 있음.
- FPGA 보드를 사용하기로 확정을 하였음.