
[Microprocessor Applications]

Lab 4: I/O Peripherals

Chester Sungchung Park (박성정)
SoC Design Lab, Konkuk University
Webpage: <http://soclab.konkuk.ac.kr>

Outline

- ❑ Creating projects
- ❑ Running C applications
- ❑ Programming C applications

Creating Projects

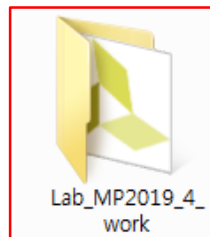
❑ Download the Vivado project

- Zybo-7010
 - ✓ <https://drive.google.com/open?id=1xXMAuQ7SO6848Wyn6y450C8tRLrPYDBU>
- Zybo-7020
 - ✓ <https://drive.google.com/open?id=1ly8K7flslqJKMX58dPmhk66mrx2keZF0>
- Download the project from the link address to the desktop.
- Unzip the downloaded project.

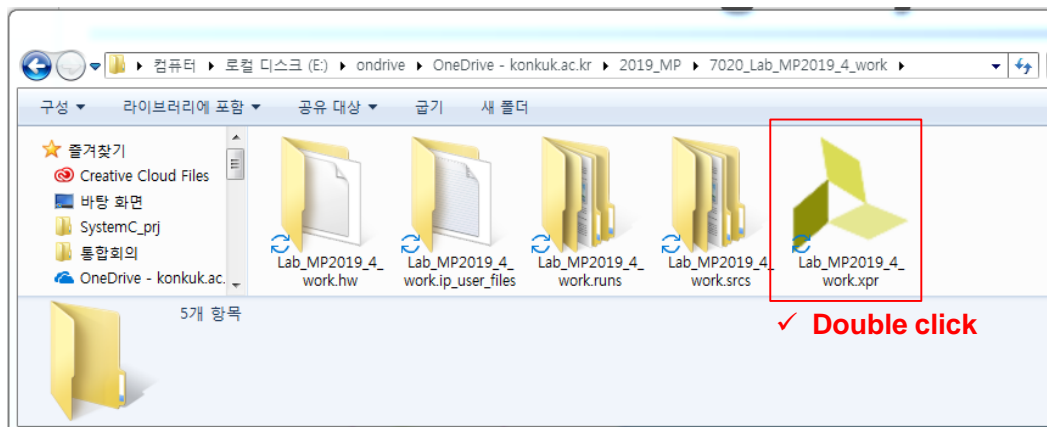
Creating Projects

❑ Run the Vivado

- Check the workspace folder in the Windows
- Double-click the '**Lab_MP_2019_4_work**' folder
- Double-click the '**Lab_MP_2019_4_work.xpr**' icon



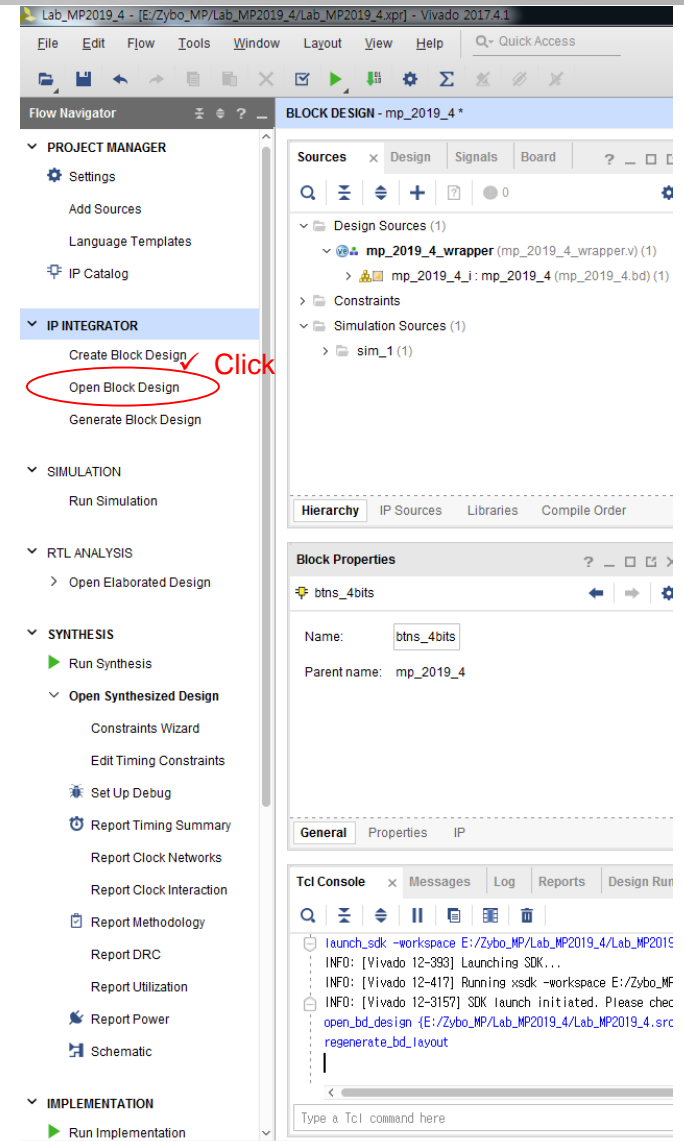
✓ Double click



✓ Double click

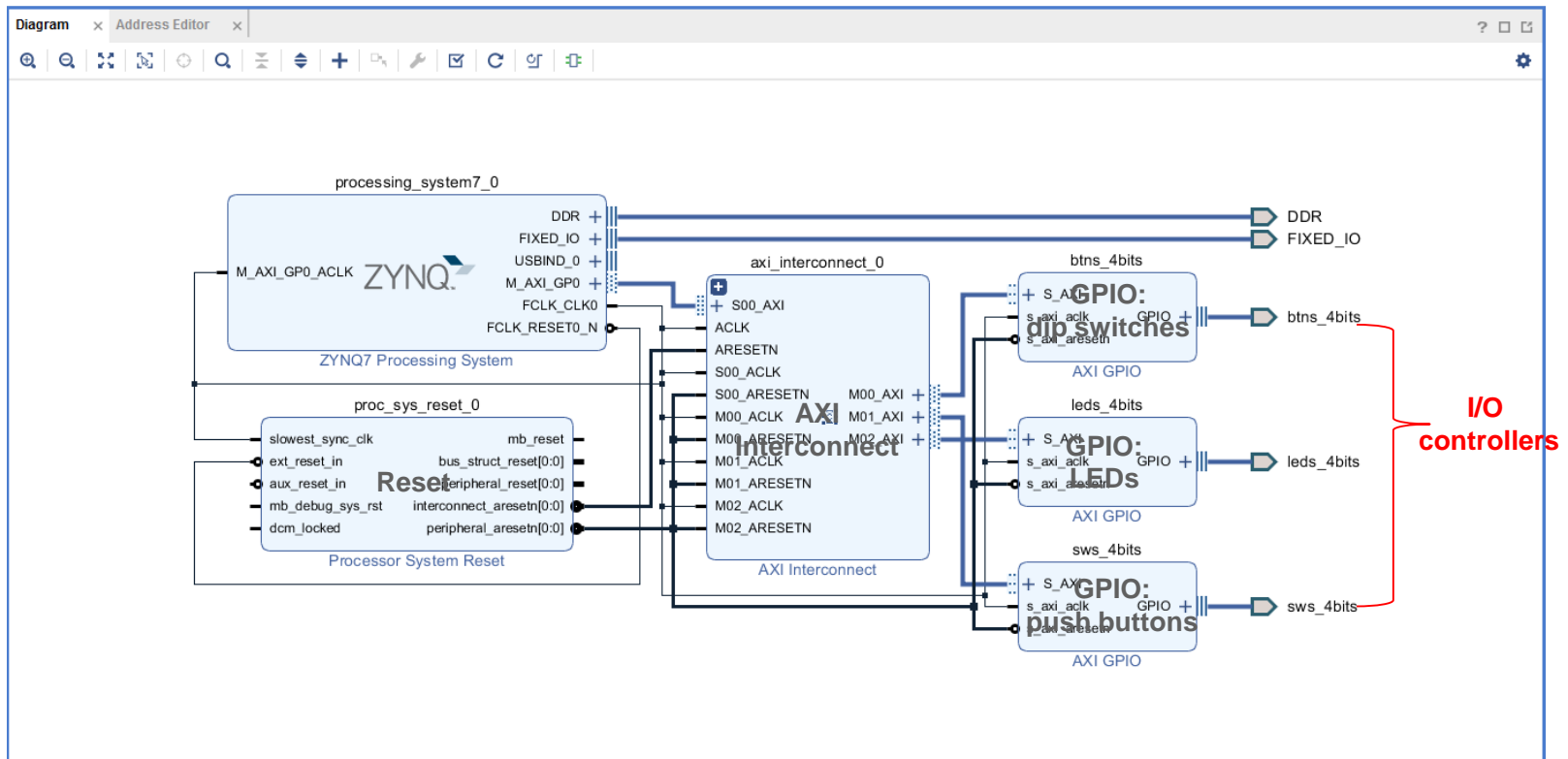
Creating Projects

- ❑ Check the block design
 - Click '*Open Block Design*'



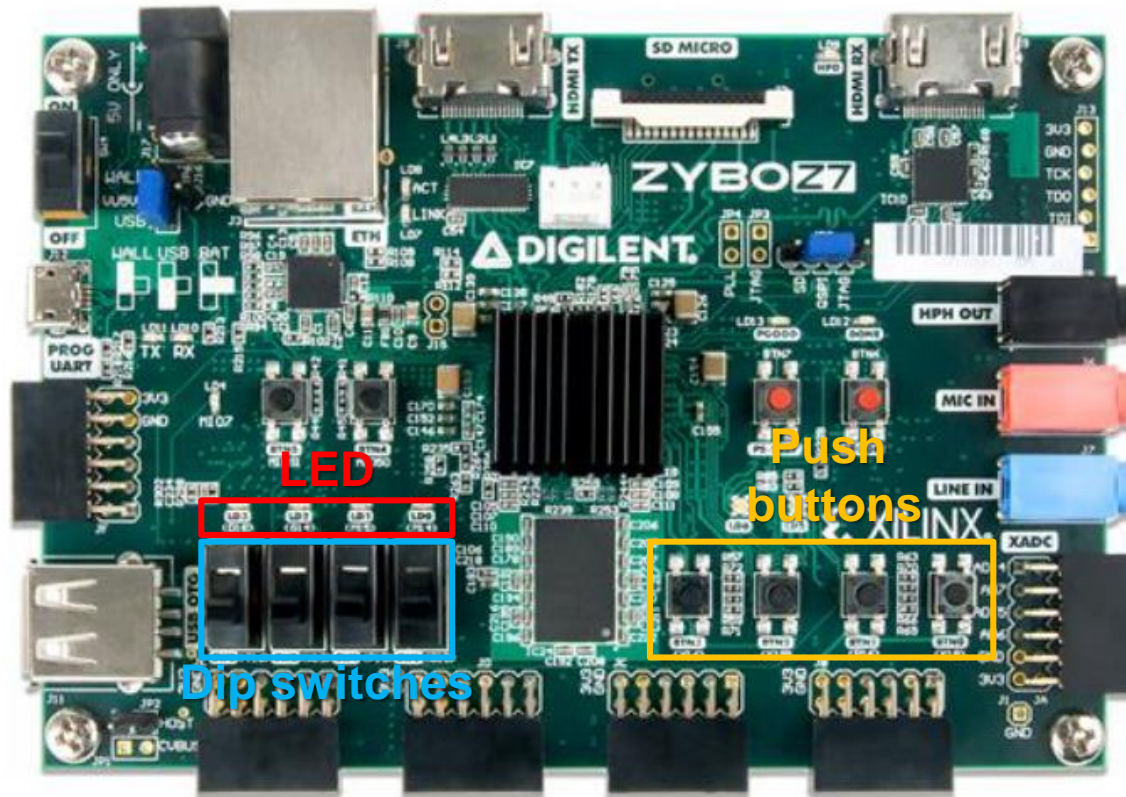
Creating Projects

- ❑ Check the block design (cont'd)
 - Check the block design that has already been designed



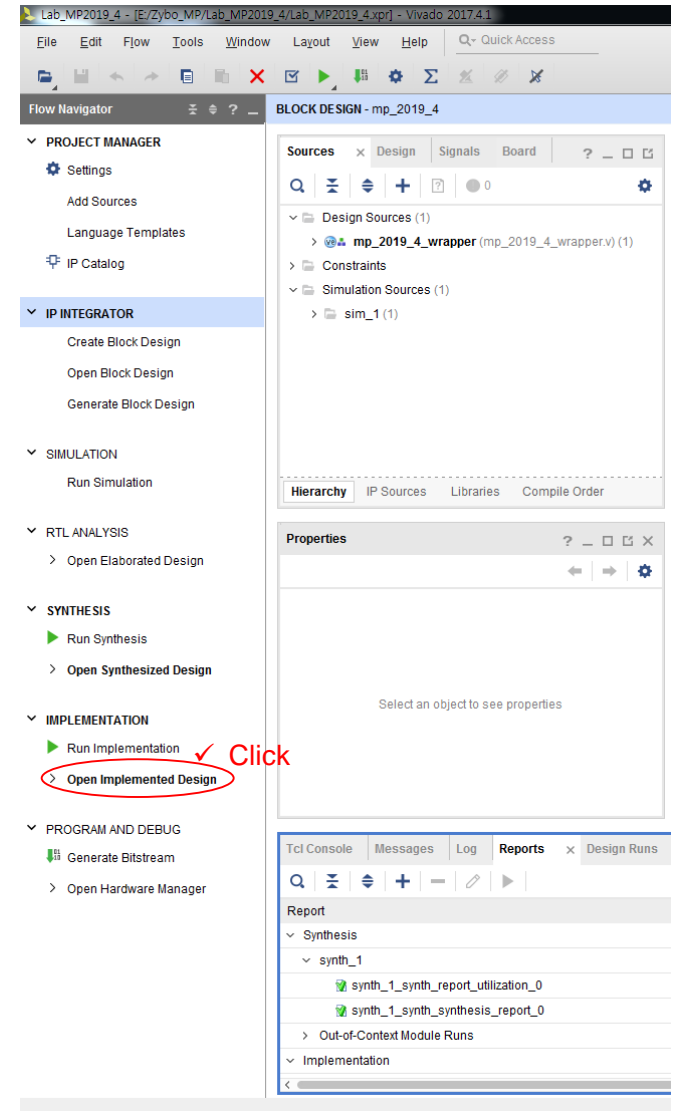
Creating Projects

- ❑ On-board I/O components



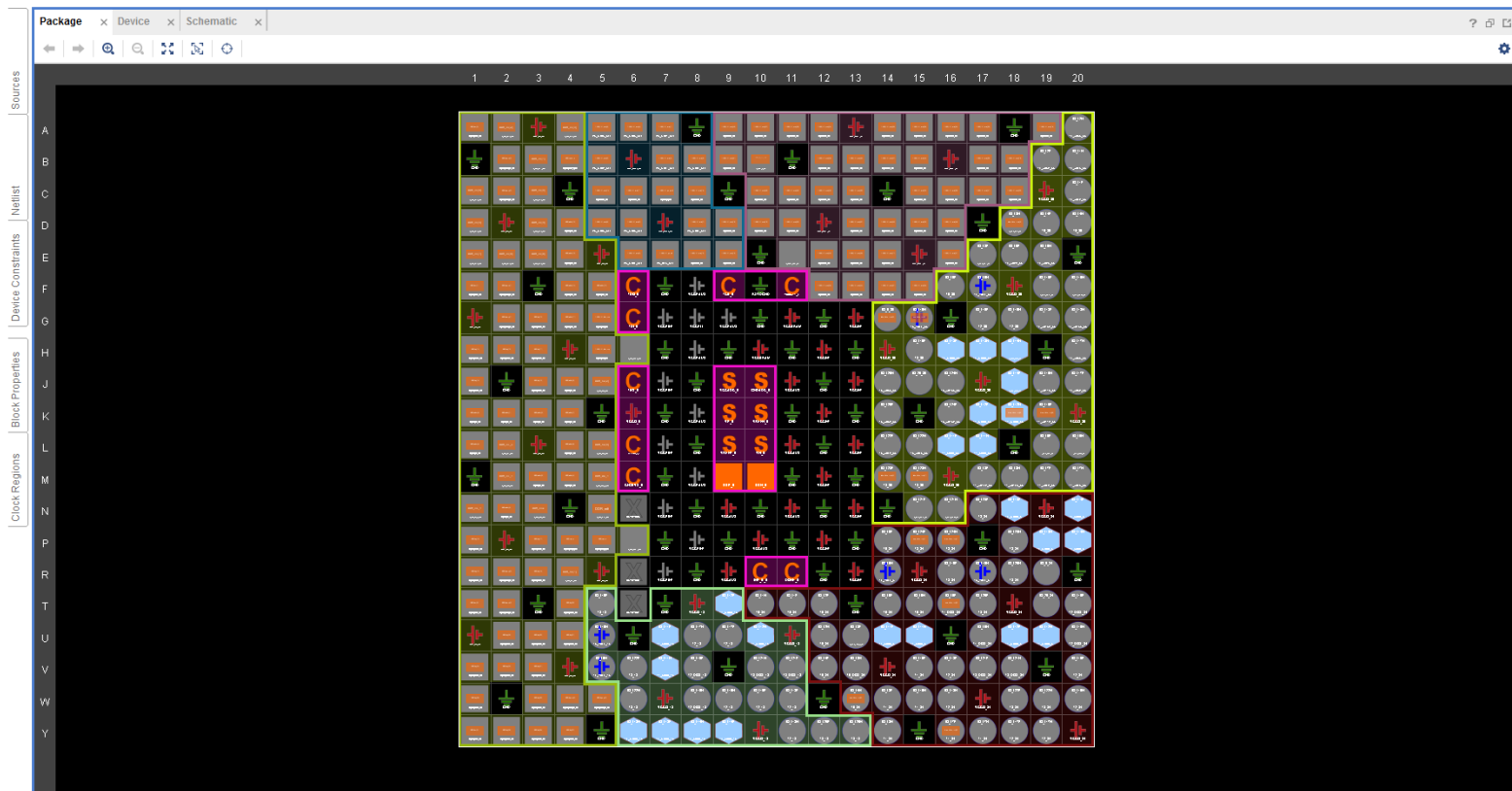
Creating Projects

- ❑ Check the implemented design
 - Click '*Open Implemented design*'



Creating Projects

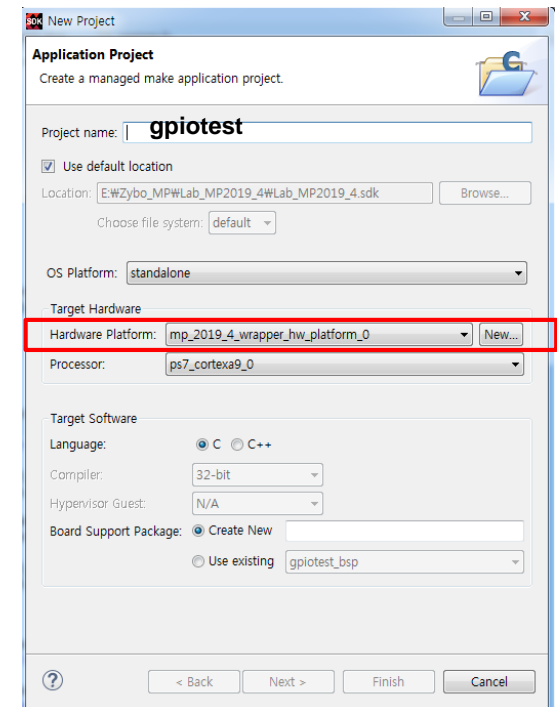
- ❑ Check the implemented design (cont'd)
 - Check that the GPIO pins are mapped.



Creating Projects

❑ Create a new application project

- Follow pp. 22~26 of the following lab workbook:
[Lab_MP2021_1_work_r3.pdf](#)
- Use the provided '**gpiotest.c**' file for this lab.
 - ✓ You should **not** set the '**Hardware Platform**' field to '**zed_hw_platform**'.



Running C Applications

□ Review the source code

- ① Delay the main function
- ② Initialize the GPIO controllers
- ③ Read/write through GPIO

① {

② {

③ {

```
#include <stdio.h>
#include <stdlib.h>
#include <xtime_l.h>
#include "xparameters.h"
#include "xgpio.h"
// #include "xutil.h"

//=====

void delay()
{
    int u = 0, c = 0, p = 0;
    for (u=0; u<9999999; u++);
    for (c=0; c<9999999; c++);
    for (p=0; p<9999999; p++);
}

int main (void)
{
    XGpio dip, push, led;
    int i = 0;
    int psb_check = 0, dip_check = 0, led_cnt = 0;

    xil_printf("-- Start of the Program --\r\n");

    XGpio_Initialize(&dip, XPAR_SWS_4BITS_DEVICE_ID);
    XGpio_SetDataDirection(&dip, 1, 0xffffffff);

    XGpio_Initialize(&push, XPAR_BTNS_4BITS_DEVICE_ID);
    XGpio_SetDataDirection(&push, 1, 0xffffffff);

    XGpio_Initialize(&led, XPAR_LEDS_4BITS_DEVICE_ID);
    XGpio_SetDataDirection(&led, 1, 0x00000000);

    while(1)
    {
        psb_check = XGpio_DiscreteRead(&push, 1);
        xil_printf("Push Buttons Status %x\r\n", psb_check);

        dip_check = XGpio_DiscreteRead(&dip, 1);
        xil_printf("DIP Switch Status %x\r\n", dip_check);

        led_cnt = dip_check;
        for (i = dip_check; i>=0; --i){
            XGpio_DiscreteWrite(&led, 1, led_cnt);
            --led_cnt;
            delay();
        }

        xil_printf("===== \r\n");
    }
}
```

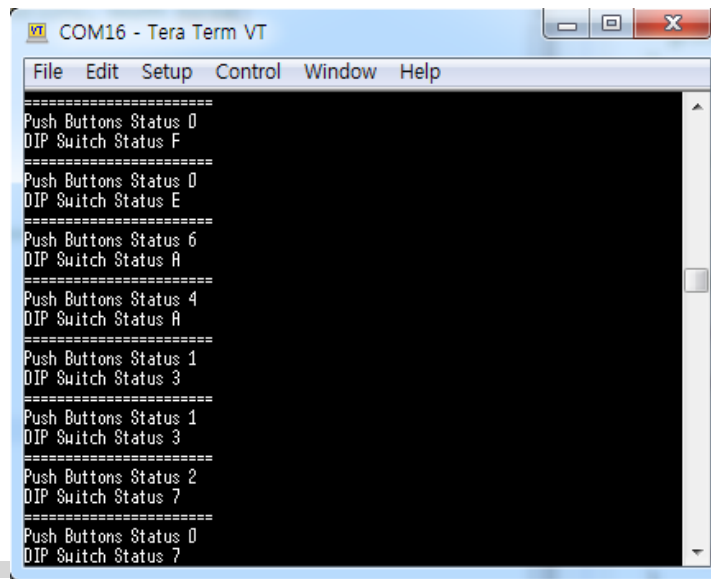
Running C Applications

- ❑ Check the definitions of the following functions
 - XGpio_Initialize
 - XGpio_SetDataDirection
 - XGpio_DiscreteRead
 - XGpio_DiscreteWrite

Running C Applications

❑ Run the application

- Follow pp. 30~33 of the following lab workbook:
[Lab_MP2022_1_work.pdf](#)
- Check the output on '*Tera Term*'
 - ✓ Check the input from the push buttons.
 - ✓ Check the input from the DIP switches.
- Check the output to the LED on the board

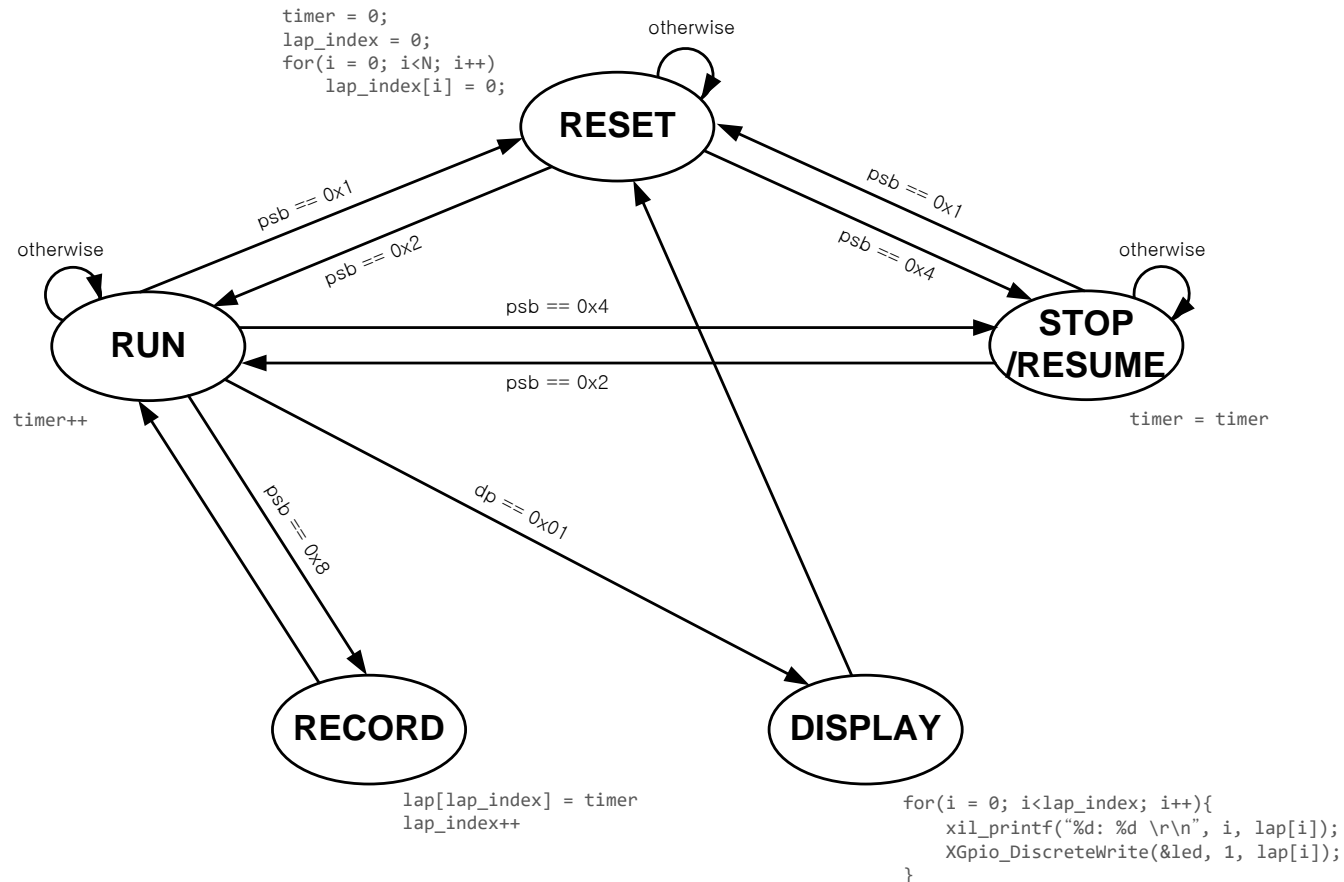


The screenshot shows a Tera Term window titled 'COM16 - Tera Term VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The main text area displays the following serial output:

```
=====
Push Buttons Status 0
DIP Switch Status F
=====
Push Buttons Status 0
DIP Switch Status E
=====
Push Buttons Status 6
DIP Switch Status A
=====
Push Buttons Status 4
DIP Switch Status A
=====
Push Buttons Status 1
DIP Switch Status 3
=====
Push Buttons Status 1
DIP Switch Status 3
=====
Push Buttons Status 2
DIP Switch Status 7
=====
Push Buttons Status 0
DIP Switch Status 7
```

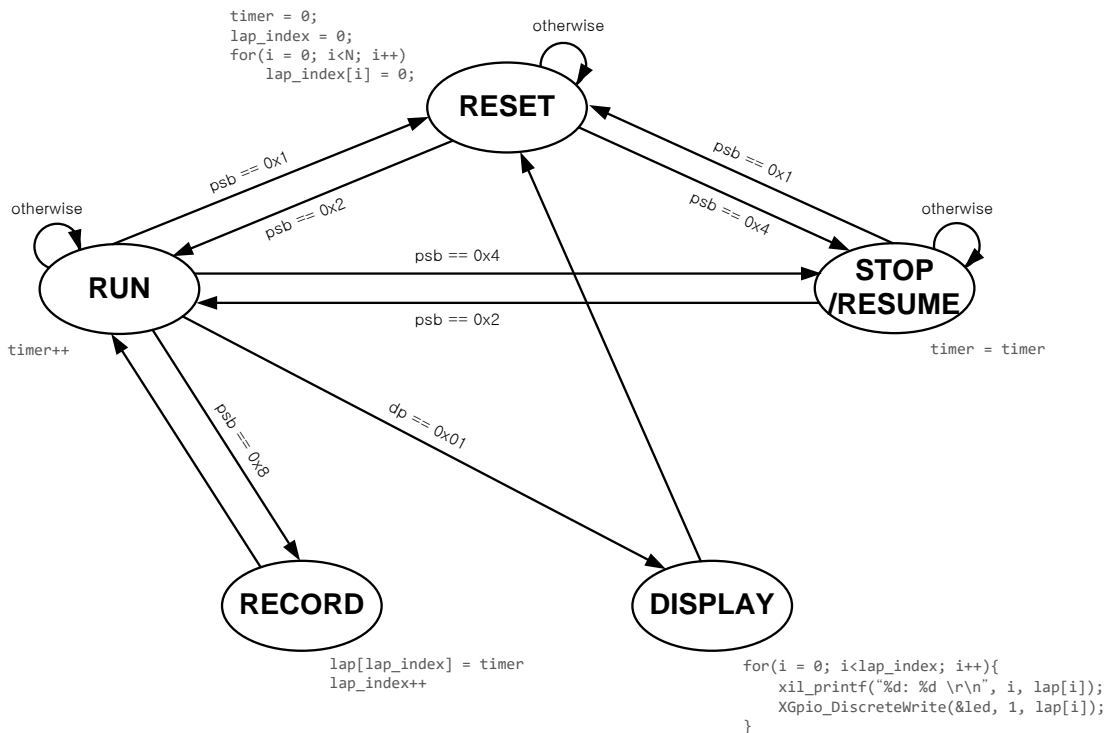
Programming C Applications

□ Stopwatch



Programming C Applications

❑ Complete the source code: '*gpiotest_problem.c*'.



```
while(1)
{
    switch(curr_state)
    {
        case RESET:
            psb = 0;
            dp = 0;
            timer = 0;
            lap_index = 0;
            for (i = 0; i < N; i++)
                lap[i] = 0;
            XGpio_DiscreteWrite(&led, 1, timer);
            psb = XGpio_DiscreteRead(&push, 1);
            if (psb & 0x02)
                next_state = RUN;
            else if (psb & 0x04)
                next_state = STOP_RESUME;
            else
                next_state = curr_state;
            break;

        case RUN:
            timer++;
```

Programming C Applications

☐ Run the application

- Follow pp. 30~33 of the following lab workbook:
[Lab_MP2022_1_work.pdf](#)

☐ Check the operation of stopwatch

- Check the output on '*Tera Term*'