```c
/*
 * Lab 4-5: Washing Machine
 * main.c
 *
 * Created: 2/14/2021
 * Author : Alec Derwent
 *
 * This main.c file is used to control a stepper motor
 * and multiple inputs/outputs from an IO shield in order to power a
 * washing machine. It contains 4 functions. The first function is
 * main(), used to execute actions on the microcontroller. Second is
 * io_init(). No IO is directly controlled by this file, instead managed by
 * stepper_motor.c and shieldIO.c. This function therefore
 * calls on the IO initialization functions in those two files
 * to set up the correct IO. The third function is Valve_Control().
 * This function has two parameters, "mode" of type char and "temp"
 * of type uint8_t. This was used to control the valves on the washing
 * machine, using "mode" within a switch statement to simplify the main
 * function. The final function is Set_Temp(). This is used to determine
 * the temperature set on the toggle switches. Using a switch statement,
 * this function prevents illegal temperature inputs by having a default
 * option that will not finish until a proper input is provided. Once finished,
 * this function returns the temperature as the bits defined below (HOT, COLD,
 * or WARM).
 */

#define F_CPU 16000000UL    // define baud rate

// introduce CHECK_BIT macro to check if desired bit in a number is 1.
// outputs 1 if true, 0 if false. Usage: CHECKBIT(var, n-1)
#define CHECK_BIT(var,pos) (((var)>>(pos)) & 1)

// define LED/valve output bits
#define HOT     0x01
#define WARM    0x02
#define COLD    0x04
#define DRAIN   0x08
#define DONE    0x10

// include files
#include <avr/io.h> // standard IO
#include <util/delay.h> // delay functions
#include "stepper_motor.h" // includes functions and values from header file
#include "shieldIO.h" // include IO shield functions for pushbuttons and
                      // toggle switches

// prototypes
void io_init(void);    // initialize all IO
void Valve_Control(char mode, uint8_t temp); // control valves for different times
uint8_t Set_Temp(void);  // Control temperature settings

int main(void)
```

```
{
    io_init();  // initialize IO for Motor and IO Shield
    while (1)
    {
        uint8_t checkStart = CHECK_BIT(Read_PB(), 0); // initialize checkStart
        while (checkStart == 0)         // if start button has not been pressed
        {
            // update checkStart until start button is pressed
            checkStart = CHECK_BIT(Read_PB(), 0);
        }

        uint8_t checkDoor = CHECK_BIT(Read_TS(), 3); // initialize checkDoor
        if(checkDoor == 0)
        {
            // exit if door is open, return to check start button
        }
        else
        {
            // use Set_Temp(); to determine the desired washing temperature
            uint8_t temp = Set_Temp();

            // FILL
            Valve_Control('F', temp);

            // WASH
            Washing_Machine('A', 8);

            // DRAIN
            Valve_Control('D', temp);

            // FILL
            Valve_Control('F', temp);

            // RINSE
            Washing_Machine('A', 6);


            // SPIN
            Valve_Control('S', temp);
            Washing_Machine('S', 8);
            Set_LED(0x00);

            // DONE LED ON
            Set_LED(DONE);  // turn on DONE LED
            checkDoor = CHECK_BIT(Read_TS(), 3);  // update checkDoor variable
            while (checkDoor == 1)  // while the door is still closed
            {
                // update checkDoor until it is opened
                checkDoor = CHECK_BIT(Read_TS(), 3);
            }
            Set_LED(0x00); // turn off done LED
        }
```

```
    }
}

// A function in order to set up IO ports
void io_init(void)
{
    Stepper_init(); // initialize stepper motor IO
    shield_io_init(); // initialize IO shield IO
}

// A function to control the valve LEDs
void Valve_Control(char mode, uint8_t temp)
{
    switch (mode)
    {
        case 'F': // if set to fill
        {
            Set_LED(temp); // open desired valve
            _delay_ms(4000); // delay 4 seconds
            Set_LED(0x00); // turn off all valves
            break;
        }

        case 'D': // if set to drain
        {
            Set_LED(DRAIN); // open drain valve
            _delay_ms(4000); // delay 4 seconds
            Set_LED(0x00); // turn off all valves
            break;
        }

        case 'S': // if set to spin cycle
        {
            Set_LED(DRAIN); // open drain valve
            _delay_ms(1000); // delay 1 second
            // DRAIN valve turned off in code after spin is completed
            break;
        }
    }
}

// A function used to determine the desired temperature setting
uint8_t Set_Temp(void)
{
    // initialize input variable as ONLY the temperature bits from the toggle switches
    uint8_t input = Read_TS() & 0x07;
    switch (input)
    {
        // if hot is selected, return hot
        case HOT:
        {
            return HOT;
```

```
            break;
        }

        // if warm is selected, return warm
        case WARM:
        {
            return WARM;
            break;
        }

        // if cold is selected, return cold
        case COLD:
        {
            return COLD;
            break;
        }

        // if input is invalid
        default:
        {
            input = Read_TS() & 0x07;
            // while the input is invalid, keep checking the input
            while(input != HOT || input != COLD || input != WARM)
            {
                // continuously update the input as ONLY the temperature bits
                input = Read_TS() & 0x07;

                // if input is correct (added for redundancy and bugs with while())
                if(input == HOT || input == COLD || input == WARM)
                {
                    break; // break out of while loop
                }
            }

            // once the input is valid, return it
            return input;
            break;
        }
    }
}
```