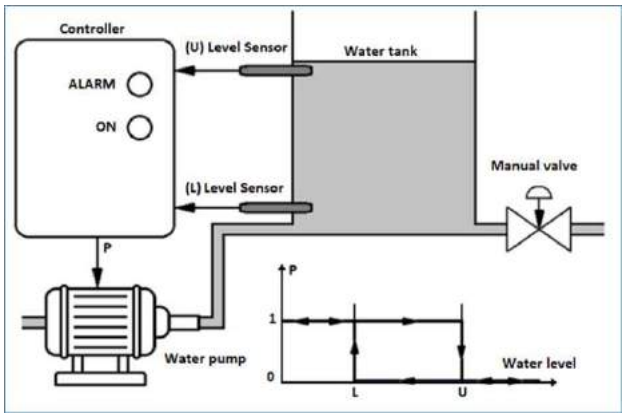
Q14—Q24: The figure to the right shows the connection between a water tank and its level Controller. The controller has two inputs: Upper-Level Sensor (U) and Lower-Level Sensor (L), which sense the level of the water inside the tank. Both sensors are connected through signal conditioning circuits to output logic high/low voltage levels. Controller output (P) is used to control a water pump. ON (Green) and ALARM (Red) LED indicators are used to show the state of the pump. The controller runs PROG1 (on the next page) on a TM4C microcontroller. The pump should be turned on (by applying a logic high on signal P), if the water level in the tank is below the low level until reaching the upper level, then it is turned off until the water level drops again below the lower level. If it happens that the pump is on for a specific time duration without having the water reaching the upper level, the pump is turned off and the ALARM LED is switched on for a specific duration. Then, the pump is switched on again (with ALARM LED off), continuing as normal.



14. In PROG1, line 1, constant CONST1 (used to reach the ALARM LED through PF1), should be

A) 0x01	B) 0x02	C) 0x03	D) 0x04

15. In PROG1, line 2, constant CONST2 (used to reach the water pump through PF2), should be

A) 0x01	B) 0x02	C) 0x03	D) 0x04

16. In PROG1, line 3, constant CONST3 (used to reach the ON LED through PF3), should be

A) 0x01	B) 0x04	C) 0x06	D) 0x08

17. In PROG1, line 5, condition C1 (used to switch pump states from PUMP_OFF to PUMP_ON) should be

A) !current_L && !old_L	B) !current_L && old_L	C) current_L && !old_L	D) current_L && old_L

18. In PROG1, line 8, condition C2 (used to switch pump states from PUMP_ON to PUMP_OFF) should be

A) !current_U && !old_U	B) !current_U && old_U	C) current_U && !old_U	D) current_U && old_U

19. In PROG1, line 9, condition C3 (used to switch pump states from PUMP_ON to PUMP_ALARM) should be

A) !current_L && on_state_counter>COUNTER_TH)	B) !current_U && on_state_counter>COUNTER_TH)
C) current_L && on_state_counter<COUNTER_TH)	D) current_U && on_state_counter<COUNTER_TH)

20. In PROG1, line 13, condition C4 (used to switch pump states from PUMP_ALARM to PUMP_ON) should be

A) current_U==1	B) old_U==0	C) on_state_counter<COUNTER_TH	D) on_state_counter>COUNTER_TH

21. In PROG1, line 6, V1, V2, V3 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, OFF, ON	C) OFF, ON, OFF	D) ON, OFF, OFF

22. In PROG1, line 11, V4, V5, V6 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, ON, ON	C) ON, ON, OFF	D) ON, OFF, ON

23. In PROG1, line 14, V7, V8, V9 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, ON, ON	C) ON, ON, OFF	D) OFF, OFF, ON

24. In PROG1, line 19, statement <S1> (used to get pins values) should be

A) return ((GPIO_PORTF_DATA_R||PIN)<0);	B) return ((GPIO_PORTF_DATA_R&PIN)>0);
C) return ((GPIO_PORTF_DATA_R|PIN)==0);	D) return ((GPIO_PORTF_DATA_R&PIN)<0);

PROG1 Q14—Q24

```
1  #define L_PIN              0x01U      // PF0 is connected to Lower-Level (L) sensor
   #define ALARM_LED_PIN      <CONST1>   // PF1 is connected to ALARM LED (Red)
2  #define P_PIN              <CONST2>   // PF2 is connected to water pump
3  #define ON_LED_PIN         <CONST3>   // PF3 is connected to ON LED (Green)
   #define U_PIN              0x10U      // PF4 is connected to Upper-Level (U) sensor
   #define COUNTER_TH         100000     // Threshold counter value
   #define PUMP_OFF           0
   #define PUMP_ON            1
   #define PUMP_ALARM         2
   #define ON                 1
   #define OFF                0

   void Controller_Init(void);
   void Set_PIN(unsigned char, unsigned char);
   unsigned char Get_PIN(unsigned char);
4  int main(void){
       unsigned char old_L, old_U; current_L, current_U, state;
       unsigned long on_state_counter;
       Controller_Init();
       state=PUMP_OFF;
       on_state_counter=0;
       for(;;){
           current_L=Get_PIN(L_PIN); current_U=Get_PIN(U_PIN);
           if (state==PUMP_OFF){
5              if (<C1>) state=PUMP_ON;
6              Set_PIN(P_PIN,V1); Set_PIN(ON_LED_PIN,V2); Set_PIN(ALARM_LED_PIN,V3);
7          } else if(state==PUMP_ON){
8              if (<C2>) state=PUMP_OFF;
9              if (<C3>){state=PUMP_ALARM; on_state_counter=0;}
10             if(!current_L) on_state_counter+=1;
11             Set_PIN(P_PIN,V4); Set_PIN(ON_LED_PIN,V5); Set_PIN(ALARM_LED_PIN,V6);
12         } else if(state==PUMP_ALARM){
               on_state_counter+=1;
13             if (<C4>){state=PUMP_ON; on_state_counter=0;}
14             Set_PIN(P_PIN,V7); Set_PIN(ON_LED_PIN,V8); Set_PIN(ALARM_LED_PIN,V9);
15         }
           old_L=current_L; old_U=current_U;
       }
   }
16 void Controller_Init(void){
       SYSCTL_RCGCGPIO_R |= 0x00000020;                    // Activate PORTF
17     while((SYSCTL_PRGPIO_R&0x00000020)==0);
       GPIO_PORTF_DIR_R = 0x0E; GPIO_PORTF_PUR_R = 0x11; GPIO_PORTF_DEN_R = 0x1F;
   }
18 unsigned char Get_PIN(unsigned char PIN){
19     <S1>
   }
20 void Set_PIN(unsigned char PIN, unsigned char value){
       if (value) GPIO_PORTF_DATA_R |= PIN;
21     else GPIO_PORTF_DATA_R &= ~PIN;
   }
```

Q33—Q38: The following figure illustrates the linear relationship between the input wave to a servo motor and the angle the motor rotates to. The servo motor controller runs PROG3 below on a TM4C microcontroller. Some statements in PROG3 are not in the right order.

PROG3 Q33—Q38

1
```
#define MOTOR_PIN       0x02U
#define ON              1
#define OFF             0
#define PERIOD_MS       20          // Period of the input wave to the servo motor
```

2
```
void Controller_Init(void);
void delay_ms(<T1> v);
void Set_PIN(unsigned char, unsigned char);
float angle_2_time(unsigned char);
unsigned char Angles_Array []={00,45,90,180};

int main(void){
    int i;
    <T1> delay_time; unsigned char angle;
    Controller_Init();
    for(;;){
        for (i=0;i<4;i++){
```

3
```
            delay_time=PERIOD_MS-delay_time; delay_ms(delay_time);
```

4
```
            delay_time=angle_2_time(angle); delay_ms(delay_time);
```

5
```
            Set_PIN(MOTOR_PIN,OFF);
```

6
```
            angle=Angles_Array[i];
```

7
```
            Set_PIN(MOTOR_PIN,ON);
```

8
```
        }
    }
}
```

9
```
<T1> angle_2_time(unsigned char angle){
```

10
```
    return (<E1>);
```

11
```
}
```

12
```
void delay_ms(<T1> v){
```

13
```
    unsigned long i=0;
```

14
```
    unsigned long d=v*(<S1>);
```

15
```
    while(i<d) i++;
}
```

16
```
void Set_PIN(unsigned char PIN, unsigned char val){
    if (val) GPIO_PORTF_DATA_R |= PIN;
    else GPIO_PORTF_DATA_R &= ~PIN;
}
void Controller_Init(void){
    SYSCTL_RCGCGPIO_R |= 0x00000020;            // Activate PORTF
    while((SYSCTL_PRGPIO_R&0x00000020)==0);
    GPIO_PORTF_DIR_R=0x0E;                      // PF0 and PF4 as input and PF3-1 as output
    GPIO_PORTF_PUR_R=0x11;                      // Enable pull-up on PF0 and PF4
    GPIO_PORTF_DEN_R=0x1F;                      // Enable digital I/O on PF4-0
}
```

33. In PROG3, line 10, equation <E1> (used to convert an angle to input pulse duration in ms) should be

A) 1-angle*(1.0/180)	B) 1+angle*(2.0/360)	C) 1+angle*(1/180)	D) 1+angle*(2.0/180)

34. In PROG3, if the input pulse duration is 1.75 ms, the angle the motor will rotate to will be

A) -135°	B) -67.5°	C) 67.5°	D) 135°

35. In PROG3, block lines 2 and lines 9 and 12, datatype <T1> should best be set to

A) int	B) unsigned int	C) long	D) float

36. In PROG3, if the controller uses a 16-MHz clock, function delay_ms(1.0) is called to insert a 1-ms delay, and a single loop iteration at line 15 takes 5 cycles, integer value <S1> in line 14 should be

A) 80	B) 800	C) 1600	D) 3200

37. In PROG3, assume that <S1> is set as in the previous question and that the loop at line 15 is changed so that a single iteration takes 6 cycles instead of 5. If function delay_ms(1.5) is called to produce a 1.5-ms pulse to have the motor rotate 90°, the actual angle the motor will rotate to will be

A) 36°	B) 91°	C) 95°	D) 144°

38. In PROG3, the correct order for lines 3—7 should be

A) 6-7-3-5-4	B) 6-7-4-5-3	C) 7-6-4-5-3	D) 6-5-4-7-3