

A Circuit for Water Level Detection and Control

Author: Shao-Quan Lu

**Graduate Student, Institute of Electrical Engineering,
National Yunlin University of Science and Technology, Touliu, Yunlin 640, Taiwan**

Introduction

For the residents who live in high buildings, low water pressure is a very common problem, especially when many people use the water at the same time. Therefore, many people install a water tower on the roof of the building. The water is stored at a water tank at the basement first. Then it is pumped into the water tower on the roof using a pumping motor. In this manner, the low water-pressure problem can be improved and the water can be used in a more efficient way.

Traditional water level controller can control the water between two levels. However, since the controller is usually placed at the top of the water tower, corrosion and mal-function may occur due to the effect of humidity to the controller. Meanwhile, if the water level of the basement tank is too low, the pumping motor may be burned because there is no water to pump.

Traditional Water Level Controller

Traditional water level controller uses floating balls as the sensors for level detection, which is depicted in Figure 1. As shown in the figure, the controller has two floating balls (will be called ball A and ball B hereafter) placed at two levels (will be called level H and level L hereafter). According to the positions of the two balls, the switch of the pumping motor is set to be on or off. The operation theory is as shown in the following table:

Step	Description
1	When the water level of the water tank is lower than level L, due to the effect of gravity, the weights of ball A and ball B will be heavy enough to make the switch be closed. This will turn on the pumping water to pump water into the water tower.
2	When the water level reaches level L, due to the effect of floating force, ball B will float on the water. However, the weight of ball A is still enough to make the switch be on. Therefore, the pumping motor keeps on pumping water.
3	When the water reaches level H, due to effect of floating force, ball A and ball B have no weight on the switch. Therefore, the motor stops pumping water.
4	When the residents use the water, the water will be lower than level H first. In this case, though the weight of ball A is on the switch, it is not heavy enough to turn on the motor. If the water is lower then level L, the weights of ball A and ball B will make the switch be closed and turn on the pumping motor.
5	Repeating the above procedure, the water in the water tower can be controlled between level H and level L.

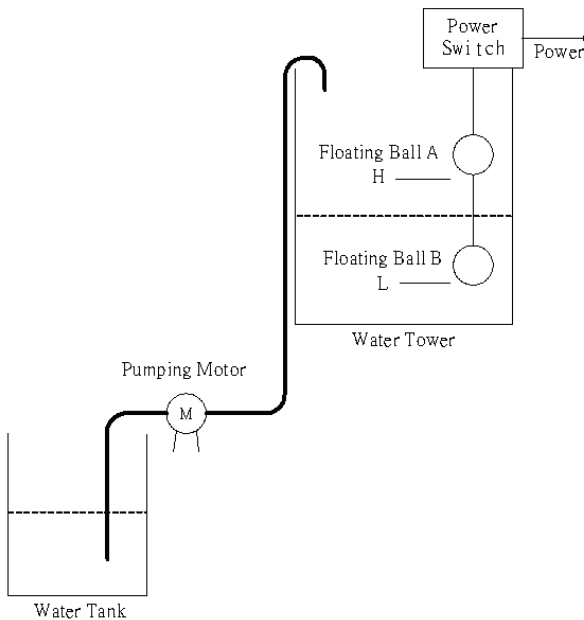


Figure 1. Block Diagram of a Traditional Water Level Controller

Using floating balls as the sensors for water level control has the advantages of simple structure and low cost. However, as mentioned above, since the contact switch of the pumping motor is usually placed at the top of water tower, the humidity may corrode the contact point of the switch. This will cause the switch to mal-function. Meanwhile, since there are no means to detect the water level of the basement water tank, it is possible that the pumping water will be burned if there is water level in the basement water tank is too low.

Using HT48R05A-1 to Design a Circuit for Water Level Detection and Control

Since the traditional water level controller has the above-mentioned problems, many people have tried to detect and control the water level in a digital manner. The usually way is to install a sensor to detect the water level of the basement tank. When the water level is less than a prescribed value, the power of the pumping motor will be shut down to prevent burning the motor.

Digital circuits to detect and control water level uses no switches. Therefore, no corrosion and mal-function problem will exist. However, such kind of circuits still has the following disadvantages:

- The water levels of the water tower and the water tank are not displayed.
- When the water level of the water tank is too low, only the power is shut down. However, status of the water tank is still unknown.
- Several digital IC components are needed in implementation. This will increase the complexity in designing.

To deal with the above disadvantages, the HT48R05A-1 I/O type MCU will be used to design a circuit for water level detection and control. This MCU is a product of the Holtek Semiconductor, which is a professional IC design company in Taiwan, especially in the design, promotion, and applications of MCUs. This company has its own core design capabilities. By capitalizing on the substantial accumulated skills within its dedicated MCU development department, Holtek has been able to release a comprehensive range of high-quality low-cost MCU devices for wide range of application areas. Among the products of Holtek Semiconductor, one MCU with the serial number HT48R05A-1 is designed for to deal with multi-input/multi-output applications. This controller is a high-performance, cost-effective, 8-bit RISC (Reduced Instruction Set Computer) architecture microcontroller. This I/O type microcontroller has many features such as 8-bit programmable input timer/event counter with overflow interrupt, 8-stage prescaler, half function and wake-up feature to reduce power consumption, on-chip crystal and RC oscillator, buzzer driving pair, and I/O ports that can be controlled by software instructions. With careful use of these MCUs, the users should result in trouble free designs. Meanwhile, maximum benefit should be gained from the many features of Holtek MCUs and these features also promise the need of least components in practical applications. Since the number of external hardware devices can be minimized, one will get lower prime cost of the products.

In the following paragraphs, the author will introduce the features of HT48R05A-1 briefly first. Then the specifications, pin assignment, and internal block diagram of this MCU will be depicted as shown in Table I, Figure 2, and Figure 3, respectively. Then the author will show how to use the HT48R05A-1 MCU to design a circuit for water level detection and control. Detailed description of hardware and software will be included. Then in the final paragraph, a conclusion is given.

HT48R05A-1 Specifications

The HT48R05A-1 is 8-bit high-performance, cost-effective RISC architecture MCU, which has 63 powerful instructions and two-level hardware stack. The Watchdog Timer (WDT) is provided to prevent program sequences from jumping to unknown location, which may be caused by certain uncontrollable external events such as electrical noises. If it happened, the program will reset the MCU. The low voltage reset circuit monitors the voltage supply. If the voltage supply is too low, it will automatically reset the device internally to avoid the error of MCU. Though the Program Memory is 512×14 only, it is enough to apply to normal cases of I/O ports. It also has 32×8 Data Memory. There are two kinds of interrupts, which are the external interrupt and the internal timer/event counter interrupt, respectively.

There are 13 bidirectional I/O, which include 8 PA ports, 3 PB ports, and 2 PC ports. It can be controlled by software instructions to determine whether the pin is a CMOS output or Schmitt Trigger input with pull-high register. PB0 and PB1 are pin-shared with BZ and , respectively. and TMR are pin-shared with PC0 and PC1, respectively. If the program wants to set PC0 as an external interrupt port, PC0 must be set to be an input port. If it wants to set as external counter-up timer, PC1 must also be set as an input port.

HT48R05A-1 includes 8-bit count-up timer input. One can select input external count-up timer, or let the internal counter-up timer as the source of the countering frequency. When the system frequency passes the prescaler, it can provide eight reference frequencies to the 8 bit counter-up timer. The range of the system frequency is from equal to the system frequency to the 1/128 of the system frequency.

Package	16 SSOP-A, 18 DIP or 18 SOP-A OTP only
I/O	13
External Interrupt	Pin-share with PC0
External Counter	Pin-share with PC1
System Oscillator	RC or Crystal
Operating Voltage	3.3V~5.5V
Program Memory	512×14
Data Memory	32×8
Stack	2 layer
Timer	8-bit

Table 1. HT48R05A-1 Specifications

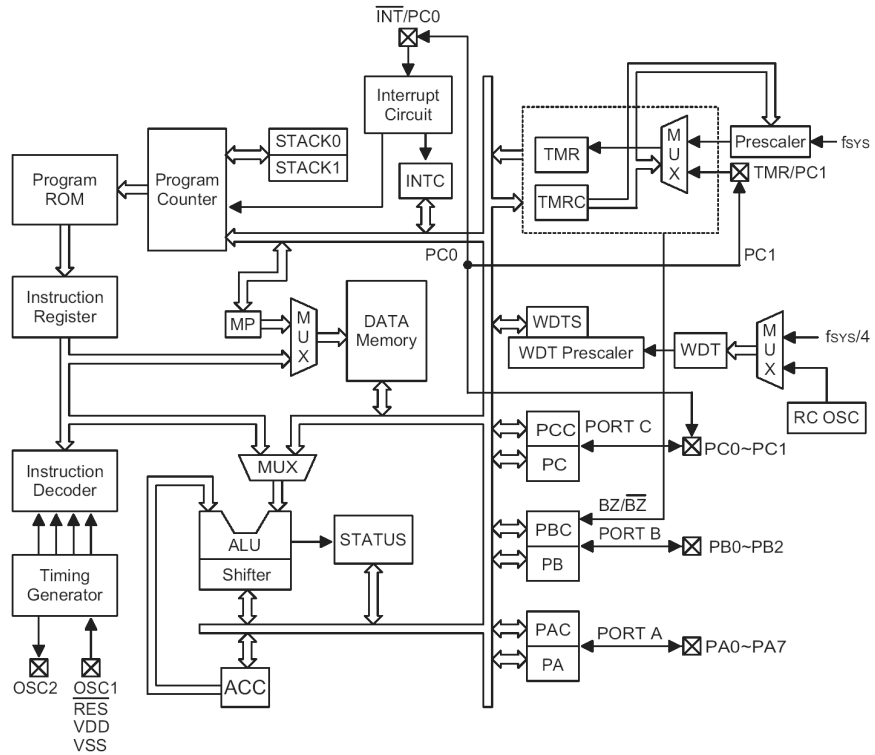


Figure 2. HT48R05A-1 Block Diagram

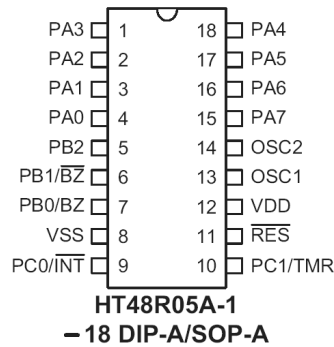


Figure 3. HT48R05A-1 Pin-assignment

The Hardware Circuit

The designed circuit is shown in Figure 4, which includes an HT48R05A-1MCU as the kernel, 4 LED to display the water level of the water tower and the water tank, an SSR (Solid State Relay) as the driver of the pumping motor, and a few external components. The basic design idea is to set PA0~PA6 as the input ports for level detection. When the water level reaches the any one of the input ports PA0~PA6, the DC supply voltage VDD is sent to the individual port. From the reading of each port, the internal software can determine the water level. Ports PC0 and PC1 are set as output ports, in which PC0 is to drive the pumping motor and PC1 is to display the status of the water level of the basement water tank. The operation theory is as follows:

- When there is enough water in the water tank, LED D4 is on. Whether the pumping motor is to turn on or not depends on the water level of the water tower.
- When the water reaches level B, LED D1 begins to flash. If the water level reaches level C, LED D1 is on.
- When the water level rises to reach level D, LED D1 is on and LED D2 begin to flash. When the water level reaches level E, LEDs D1 and D2 are on for indication.
- When the water level reaches level F, LED D1 and LED D2 are on and LED D3 begins to flash. When the water level reaches level G, all the three LEDs are on to indicate that the water level is on the full level.
- When the water level is lower than level B, the MCU will turn on the pumping motor only when the level of the water tank is higher then level I. This will prevent the damage of the pumping motor.
- When the level of the water tank is higher than level I, the motor will start to pump the water and the LEDs will flashing in the sequence D1→D2→D3→D1→D2→D3 to indicate the operation of the motor.
- When the level of the water tank is lower than level I, the motor stops the pumping operation. LED D4 is off and LEDs D1, D2, and D3 are to indicate the water level of the water tower according to the above-mentioned rules.

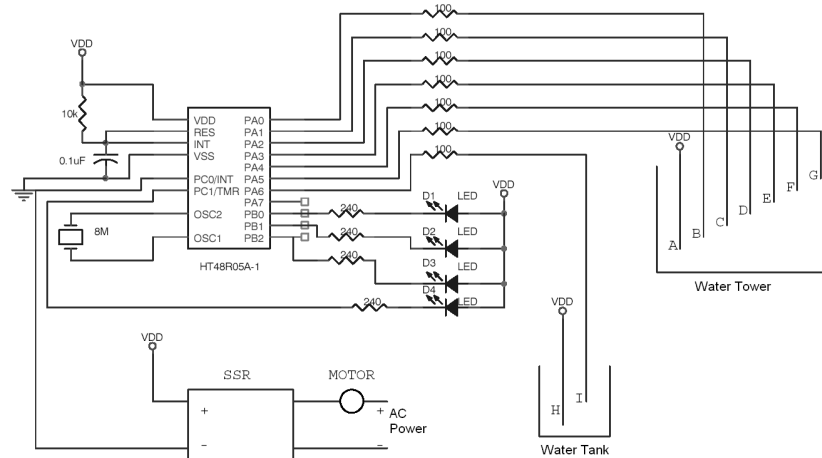


Figure 4. Hardware of the Circuit for Water Detection and Control

Compared with the traditional digital water level controller, the designed system has the following advantages:

- No mechanical switch is used. Therefore, there exists no corrosion problem.
- When the water level is lower than a prescribed value, the power of the pumping motor will be shut down to prevent the damage of the motor.
- The levels of the water tower and water tank can be monitored and displayed by LED.
- Since the HT48R05A-1 MCU is used, very external components are used. This will increase the flexibility and liability of the designed circuit.

Conclusion

In this paper, the author introduced the operation theory of traditional water level controller. Moreover, it is shown how to use a cost-effective I/O type MCU as the kernel to design a circuit for water level detection and control. This circuit monitored and controlled the water level of a water tower, and used LED for indication. From the hardware circuit, one can find that very few external components are needed. Moreover, one can also find that the software is very easy to be developed and modified if more functions are to be added. All these features verify the high performance of the HT48R05A-1 MCU and more application cases can be explored.

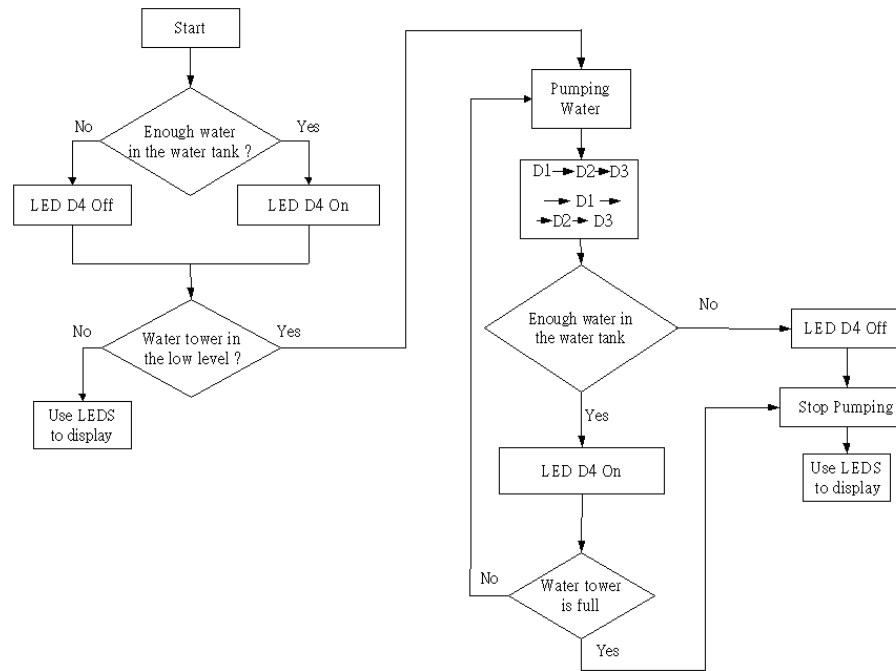


Figure 5. Software Flowchart

Appendix

Source Codes

```

#include ht48r05a-1.inc

data .section 'data'          ; data block
count1 db ?                  ; temporary data variable
count2 db ?                  ; temporary data variable
matrix db ?                  ; temporary data variable
matrix1 db ?                 ; temporary data variable
;-----
code .section at 0 'code'
    org 00h
    jmp start

start:
    clr intc
    clr tmrc                  ;setup initial value register
    clr tmr
    set pac
    set pbc                  ;setup I/O port input mode
    set pcc

main:
    set pac                  ;setup port A to the input mode
    clr pbc                  ;setup port B to the output mode
    clr pcc                  ;setup port C to the output mode
    clr pa                   ;clear port A
    set pc                   ;turn LEDs off
    set pb                   ;stop pumping .led(d4)

keyloop:
    mov a,pa                 ;read input state
    mov matrix,a
    and a,040h
    sz acc                   ;enough water in the water tank ?
    jmp                      ;yes.led(d4) on
    set pc                   ;no.led(d4) off

keyloop1:
    mov a,matrix
    and a,001h
    sz acc                   ;water town in the low level ?
    jmp get_level           ;no. leds display water town's level

```

```

set    pb          ;yes. pumping water
mov    a,matrix
and    a,040h
sz     acc         ;enough water in the water tank ?
jmp    motor      ;yes. pumping water
jmp    keyloop    ;no. use leds to display
motor:
set    pb          ;led(d1.d2.d3)display pumping water
call   delay      ;delay
mov    a,06h
mov    pb,a
call   delay      ;delay
mov    a,04h
mov    pb,a
call   delay      ;delay
mov    a,00h
mov    pb,a
mov    a,00h
mov    pc,a       ;led(d4) bright. water tank is full
call   delay      ;delay
nop    nop
mov    a,pa
and    a,020h
sz     acc         ; enough water in the water tower ?
jmp    LEVEL1     ;yes. stop pumping
jmp    LEVEL2     ;no. pumping water
LEVEL2:
mov    a,pa
and    a,040h
sz     acc         ; enough water in the water tank ?
jmp    motor      ;yes. pump water
set    pc         ;no. stop pumping
jmp    keyloop
LEVEL1:           ;stop pumping
mov    a,01h
mov    a,pc       ;port c.0 output 1
jmp    keyloop
get_water:
mov    a,01h
mov    pc,a       ;port c.1 write 0

```

```

        call    delay      ;delay
        jmp     keyloop1
get_level:
        mov     a,matrix
        and     a,020h
        sz     acc         ;water tower level arrive G level ?
        jmp     dis_6     ;yes. led display height of G water level
        mov     a,matrix
        and     a,010h
        sz     acc         ;water tower level arrive F level
        jmp     dis_5     ;yes. led display height of F water level
        mov     a,matrix
        and     a,008h
        sz     acc         ;water tower level arrive E level
        jmp     dis_4     ;yes. led display height of E water level
        mov     a,matrix
        and     a,004h
        sz     acc         ;water tower level arrive D level
        jmp     dis_3     ;yes. led display height of D water level
        mov     a,matrix
        and     a,002h
        sz     acc         ;water tower level arrive C level
        jmp     dis_2     ;yes. led display height of C water level
        mov     a,matrix
        and     a,001h
        sz     acc         ;water tower level arrive B level
        jmp     dis_1     ;yes. led display height of B water level
        jmp     keyloop
dis_1:
        mov     a,006h     ;led display height of B water level
        mov     pb,a
        call    delay
        mov     a,07h
        mov     pb,a
        call    delay
        jmp     keyloop
dis_2:
        mov     a,006h     ;led display height of C water level
        mov     pb,a
        jmp     keyloop

```

```
dis_3:
    mov     a,004h      ;led display height of D water level
    mov     pb,a
    call    delay
    mov     a,006h
    mov     pb,a
    call    delay
    jmp     keyloop

dis_4:
    mov     a,004h      ;led display height of E water level
    mov     pb,a
    jmp     keyloop

dis_5:
    mov     a,000h      ;led display height of F water level
    mov     pb,a
    call    delay
    mov     a,004h
    mov     pb,a
    call    delay
    jmp     keyloop

dis_6:
    mov     a,000h      ;led display height of G water level
    mov     pb,a
    jmp     keyloop

delay proc          ;delay program
    mov     a,00eh      ;load counter value
    mov     count1,a
    mov     count2,a

dl:
    sdz     count2
    jmp     dl
    sdz     count1
    jmp     dl
    ret

delay endp
end
```