PROBLEM

Most older sump pumps have a single level float control to regulate when the pump is turned on and off. When the water level rises to the float control, the pump turns on. The pump quickly pumps the water down to just below the float control in about a minute, shutting off the pump. Since the water level is just below the float control, the water rises enough to turn the sump pump on in just a couple of minutes. This control system makes the sump pump turn on 20 to 30 times an hour during a rain storm. This constant turning on and off of the motor leads to premature motor failure resulting in a flooded basement.

SOLUTION

Newer sump pumps have a two level control system. This project is intended to retrofit sump pumps that are already installed. The solution presented here uses a two water level control system. Using two water level sensors and the PIC12C508 microcontroller, we are able to introduce hysteresis into the control loop. This allows the pump to stay on longer and stay off longer during a rain storm. This reduction in power cycling leads to prolonged motor life, reducing the chance of having a failed sump pump lead to a flooded basement. The pump turns on when the water level reaches the top water level sensor and stays on until enough water is pumped out to lower the level to below the bottom sensor. The bottom sensor is positioned just above the lowest level the pump can operate at, so that the pump never dry pumps. The pump does not turn back on until the water level rises to the top sensor. This circuit is very cost effective allowing the consumer to retrofit an older sump pump instead of buying a new model.

APPLICATION OPERATION

Two water level sensors are used. One sensing the desired top water level and the other sensing the desired bottom water level. The bottom water level sensor is used to ensure that the sump is never pumped completely dry, which could damage the sump pump. The PIC12C508 uses reads both sensor states and acts upon them according to the following table.

<table>
<thead>
<tr>
<th>Top</th>
<th>Off</th>
<th>On</th>
<th>Off</th>
<th>On</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>On</td>
</tr>
<tr>
<td>Action</td>
<td>Turn On</td>
<td>No Change</td>
<td>Invalid</td>
<td>Turn Off</td>
</tr>
</tbody>
</table>

The condition that the bottom sensor is off and the top sensor is an invalid state. For this condition the pump is turned on as a fail safe measure. In case the bottom switch failed and was always reading off, this fail safe measure would still prevent sump from overflowing though functionality will be diminished to the standard type sump pump control.
Sensor Interface

**BLOCK DIAGRAM**

```
Vdd
3 V

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GP</td>
<td>GP</td>
<td></td>
<td>VPP</td>
<td>TOCK1</td>
</tr>
</tbody>
</table>

Top Level Switch

Bottom Level Switch

Triac Control
```

**FLOW CHART:**

1. Power On Setup
2. Read Top Switch
3. **Top Switch = ON?**
   - Yes: Turn Pump ON
   - No: Read Bottom Switch
4. **Bottom Switch = ON?**
   - Yes: Delay 1 sec
   - No: Turn Pump OFF
GRAPHICAL HARDWARE REPRESENTATION

BILL OF MATERIALS

<table>
<thead>
<tr>
<th>Ref</th>
<th>Part#</th>
<th>Manufacture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12C508</td>
<td>Microchip Technology</td>
</tr>
<tr>
<td>Q1</td>
<td>L4004L3</td>
<td>Teccor</td>
</tr>
</tbody>
</table>

MICROCHIP TOOLS USED

Assembler/Compiler version:

MPLAB 3.22.02
APPENDIX A:  SOURCE CODE

Title "Sump Pump Controller"
Subtitle "Version 1.0"

; Written by Brian Iehl 7/12/97
; Last modified 7/25/97

list p=12C508

INCLUDE c:\apps\mplab\p12c508.inc

SetIO equ B'00001100' ; 0 for output, 1 for input

GPIO0  equ 0
GPIO1  equ 1
GPIO2  equ 2
GPIO3  equ 3
GPIO4  equ 4
GPIO5  equ 5

TopState equ GPIO3 ; Top water level switch Input
BotState equ GPIO2 ; Bottom water level switch Input
SwOn   equ 0 ; Switch closed GPIO is Low
SwOff  equ 1 ; Switch open GPIO is high
TopSwValue equ B'00001000' ; Used to test GPIO3 bit
BotSwValue equ B'00000100' ; Used to test GPIO2 bit

TriacCntl equ GPIO0 ; Output Triac Control
TriacOn  equ 1 ; Hi to turn Triac on
TriacOff equ 0 ; Lo to turn Triac off

ReadDelay equ D'15' ; S to wait for next reading

ScratchPadRam equ 0x07
DelayValue equ ScratchPadRam+0; For DelayRoutine
SDelayValue equ ScratchPadRam+1; For SDelayRoutine

;********************  Macros **************************************

MOVLF MACRO LL, FF ; Move Literal to register file
    MOVLW LL ; Load literal
    MOVWF FF ; Store in register file
    ENDM ; end MOVLF

mSDelay MACRO mS ; Number of mS to delay up to 255 mS
    LOCAL
    Loop, SetTmr
    MOVLF mS, DelayValue ; store number of mS delay
    CLRWD ; avoid unintentional reset
    SETMR ; Set prescaler to 256, clear PSA, Clear T0CS
    MOVWF B'00000111' ; store prescaler value
    LOOP ; 4 * 256 = 1024 uS ~ 1 mS
    MOVF TMRO, w ; force check zero
    BTFSS STATUS, Z ; w = 0 if same, so Z is set
    GOTO Loop ; not 0 so loop again
    DECFSZ DelayValue, f ; one more mS passed
    GOTO SetTmr ; if DelayValue = 0 then done
    ENDM ; end mSDelay

SDelay MACRO S ; Number of Seconds delay up to 63
LOCAL Loop
MOVLF S'4, SDelayValue ; store number of S delay
Loop
mSDelay D'250' ; Delay 0.25 sec
DECFSZ SDelayValue, f ; count down number of S
goto Loop ; not done reset timer
; if DelayValue = 0 then done
ENDM ; end SDelay

;********************************************************************
org 0x0A ;start address 0x0A
goto Start ;start address 0x0A
org 0x10

; Start

Setup
MOVlw SetIO ; Load IO configuration byte
TRIS GPIO ; Set GPIO with contents of w
CLRF DelayValue
CLRF SDelayValue

MainLoop
MOVF GPIO, w ; read GPIO register
ANDLW TopSwValue ; Clear all bits except TopSwState
BTFSS STATUS, Z ; if TopSwState = Lo = ON
goto ReadBot ; if TopSwState = Lo = ON

TurnOn
BSF GPIO, TriacCntl ; Turn Triac on
goto Cont ; Turn Triac on

ReadBot
MOVF GPIO, w ; read GPIO register
ANDLW BotSwValue ; Clear all bits except TopSwState
BTFSS STATUS, Z ; if BotSwState = Lo = on

TurnOff
BCF GPIO, TriacCntl ; Turn Triac Off

Cont
SDelay ReadDelay ; Wait before next reading
goto MainLoop ; Return to top of main loop

END
NOTES: