MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING AND TECHNOLOGY

COURSE NAME : Name: PBL_1
(CSE2PR04A)

ACADEMIC YEAR : 2024-25	SEMESTER : III
CLASS : SY BTECH (CSE)	PANEL : I
DATE OF ASSIGNMENT: 20/09/2024	DATE OF SUBMISSION : 07/11/2024
CATEGORY OF ASSIGNMENT : CCA 3	TYPE OF ASSIGNMENT : GROUP ASSIGNMENT

NAME	ROLL NO	PRN
ZEESHAN KHAN	53	1032233386
LUV PARTANI	48	1032233368
PURV MEGHPURIA	46	1032233364
ADITYA SHARMA	47	1032233365

DIRECTED BY : Dr. Ranjana Agarwal

WATER LEVEL CONTROLLER

8051 MICROCONTROLLER ASSEMBLY LANGUAGE PROGRAMMING

ABSTRACT

This project addresses the common challenge of efficiently managing water levels in overhead tanks to prevent overflow and water wastage. Manual monitoring of water tanks can be inefficient and prone to human error, leading to resource wastage and increased operational costs. To address this issue, an automated water level control system was developed using the 8051 microcontroller programmed in assembly language.

The system continuously monitors the water level within the tank using sensor inputs, triggering the water pump when levels fall below a specified threshold and turning it off once the tank reaches its maximum level. The 8051 microcontroller's efficient handling of inputs and outputs enables reliable and real-time response to water level changes. The project was implemented and tested on the edsim51 simulator, validating the control logic and ensuring precise functionality.

Key outcomes include a cost-effective, automated solution that reduces manual intervention, optimizes resource use, and prevents water overflow. This project demonstrates the applicability of microcontroller-based automation for sustainable water management in residential, agricultural, and industrial settings.

CONTENTS

INTRODUCTION

Objective:

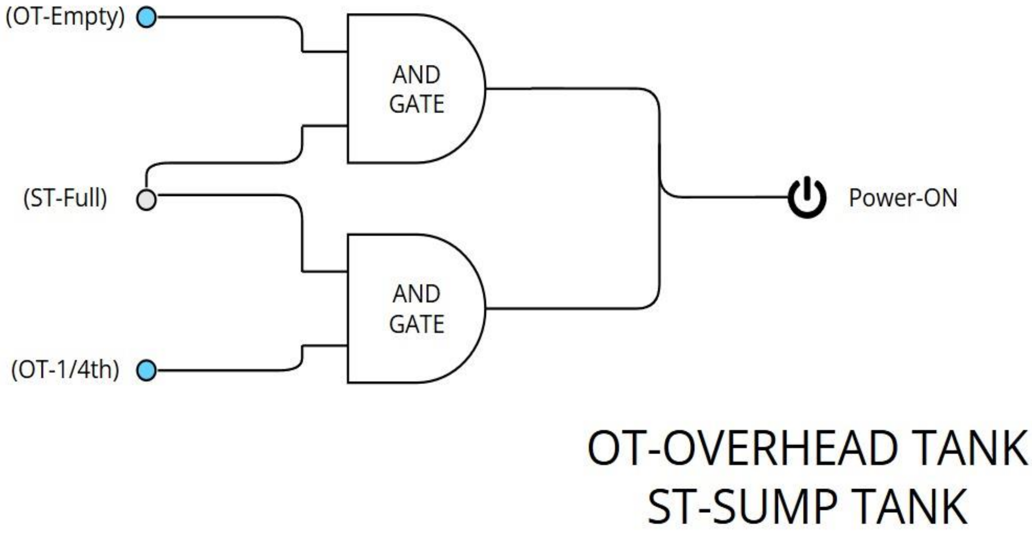
The objective of this project is to design and implement an automated **Water Level Controller** using the 8051 microcontroller assembly language programming. This system is intended to monitor the water level in an overhead tank and automatically activate the water pump when the level falls below a pre-set threshold, deactivating it once the tank is full.

Motivation:

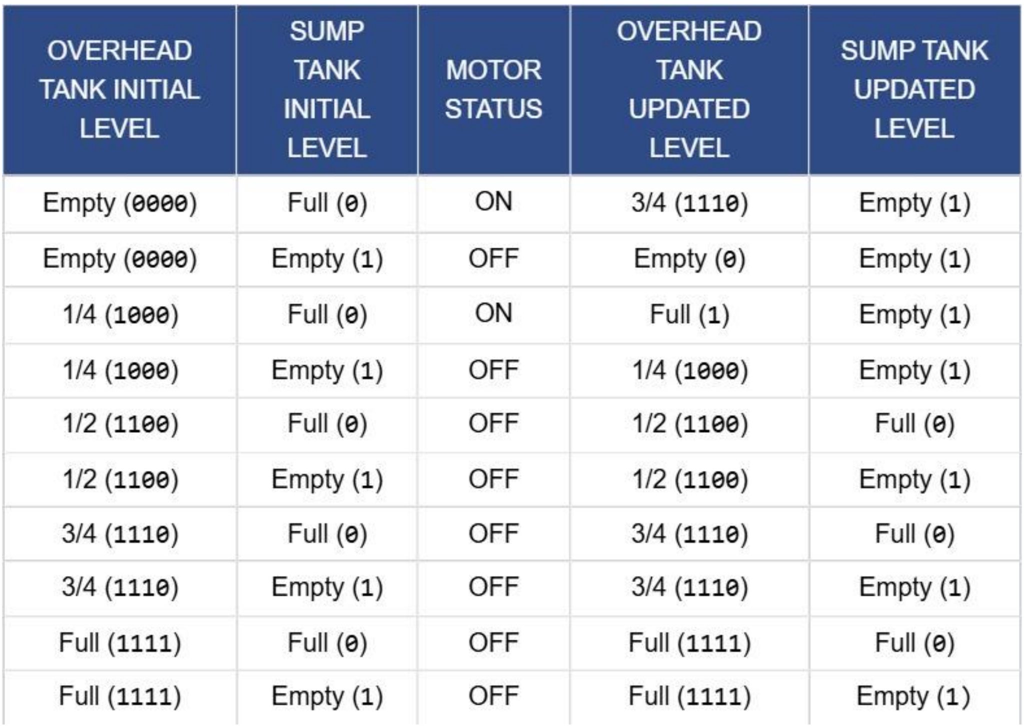
The motivation behind choosing this problem stems from the widespread challenges associated with manual water tank management. In households, industries, and agricultural areas, water overflow and pump overuse can lead to resource wastage, higher costs, and operational inefficiencies. An automated water level control system addresses these issues by providing a practical and sustainable solution. Using the 8051 microcontroller for this task allows for precise control and cost-efficiency, making the system accessible for various applications. Additionally, programming the system in assembly language on the edsim51 simulator enhances our understanding of microcontroller operations, low-level programming, and embedded system design.

IMPLEMENTATION

Logic Diagram:



Truth Table:



OVERHEAD TANK INITIAL LEVEL	SUMP TANK INITIAL LEVEL	MOTOR STATUS	OVERHEAD TANK UPDATED LEVEL	SUMP TANK UPDATED LEVEL
Empty (0000)	Full (0)	ON	3/4 (1110)	Empty (1)
Empty (0000)	Empty (1)	OFF	Empty (0)	Empty (1)
1/4 (1000)	Full (0)	ON	Full (1)	Empty (1)
1/4 (1000)	Empty (1)	OFF	1/4 (1000)	Empty (1)
1/2 (1100)	Full (0)	OFF	1/2 (1100)	Full (0)
1/2 (1100)	Empty (1)	OFF	1/2 (1100)	Empty (1)
3/4 (1110)	Full (0)	OFF	3/4 (1110)	Full (0)
3/4 (1110)	Empty (1)	OFF	3/4 (1110)	Empty (1)
Full (1111)	Full (0)	OFF	Full (1111)	Full (0)
Full (1111)	Empty (1)	OFF	Full (1111)	Empty (1)

Components:

1. LEDs

In this project, LEDs serve as visual indicators connected to the bits of Port 1 (P1) of the 8051 microcontroller. Each bit represents a different water level status in the overhead tank as well as sump tank. Specifically:

- **P1.0** indicates that the overhead tank is empty (0%).
- **P1.1** indicates a water level of 25%.
- **P1.2** indicates a water level of 50%.
- **P1.3** indicates a water level of 75%.
- **P1.4** indicates that the overhead tank is full (100%).
- **P1.6** indicates that the sump tank is full.
- **P1.7** indicates that the sump tank is empty.

The microcontroller turns on the corresponding LED when the water level reaches a specific threshold by setting the appropriate bits, while it clears bits to indicate lower levels.

2. Motor (Water Pump)

The motor, representing the water pump, is controlled via Port 3 (P3.0). The 8051 microcontroller activates or deactivates the motor based on the water level:

- The motor is turned **off** when the water level is full or when a low sump condition is detected.
- The motor is turned **on** when the water level drops below 25%, indicating that water needs to be pumped into the tank.

This control logic is crucial for maintaining optimal water levels and preventing overflow or depletion.

3. LCD

The LCD display in this project provides a real-time, textual representation of the updated water levels in both the overhead and sump tanks. Connected to the 8051 microcontroller, the LCD is programmed to display messages indicating the exact water level, helping users monitor both tanks' status at a glance.

- **Overhead Tank Level Display**: The LCD displays messages when threshold, condition is met (empty) to reflect the updated water level (3/4th).
- **Sump Tank Level Display**: The display also shows the status of the sump tank, indicating whether the water source (sump) is empty or full.

Code:

```
MOV P2,#11111100B
MOV P1,#10111111B ; bit 6 denotes sump level (0-full, 1-empty)
MOV A,#00000000B
MAIN:ACALL SMPCK
MOV A,P2
CJNE A,#11111111B,LABEL1 ;full
SETB P1.1
SETB P1.2
SETB P1.3
SETB P1.4
CLR P1.0
setb P3.0   ;motor off
LABEL1:MOV A,P2
CJNE A,#11111110B,LABEL2 ;3/4
SETB P1.0
SETB P1.2
SETB P1.3
SETB P1.4
CLR P1.1
LABEL2:MOV A,P2
CJNE A,#11111100B,LABEL3 ;1/2
SETB P1.0
SETB P1.1
SETB P1.3
SETB P1.4
CLR P1.2
LABEL3:MOV A,P2
CJNE A,#11111000B,LABEL4 ;1/4
SETB P1.0
SETB P1.1
SETB P1.2
SETB P1.4
CLR P1.0
clr p1.7
setb p1.6
JB P1.6,LABEL4
clr P3.0  ;motor on
LABEL4:MOV A,P2
CJNE A,#11110000B,MAIN ;empty
SETB P1.0
SETB P1.1
SETB P1.2
SETB P1.3
CLR P1.4
JB P1.6,MAIN
clr P3.0 ;motor on
MOV 2EH, #'U'
MOV 2FH, #'-'
MOV 30H, #'O'
MOV 31H, #'T'
MOV 32H, #'='
MOV 33H, #'3'
MOV 34H, #'/'
MOV 35H, #'4'
MOV 36H, #','
MOV 37H, #'S'
MOV 38H, #'T'
MOV 39H, #'='
MOV 3AH, #'E'
MOV 3BH, #'M'
MOV 3CH, #'T'
MOV 4DH, #0
CLR P1.3
CLR P1.7
CLR P1.6
SETB P1.5
CLR P1.4
SETB P1.2
CLR P1.2
CALL delay
SETB P1.2
CLR P1.2
SETB P1.7
SETB P1.2
CLR P1.2
CALL delay
CLR P1.7
CLR P1.6
CLR P1.5
CLR P1.4
SETB P1.2
CLR P1.2
SETB P1.6
SETB P1.5
SETB P1.2
CLR P1.2
CALL delay
CLR P1.7
CLR P1.6
CLR P1.5
CLR P1.4
SETB P1.2
CLR P1.2
SETB P1.7
SETB P1.6
SETB P1.5
SETB P1.4
SETB P1.2
```

```
CLR P1.2
CALL delay
SETB P1.3
MOV R1, #2EH
loop:
MOV A, @R1
JZ finish
CALL sendCharacter
INC R1
JMP loop
finish:
JMP $
sendCharacter:
MOV C, ACC.7
MOV P1.7, C
MOV C, ACC.6
MOV P1.6, C
MOV C, ACC.5
MOV P1.5, C
MOV C, ACC.4
MOV P1.4, C
SETB P1.2
CLR P1.2
MOV C, ACC.3
MOV P1.7, C
MOV C, ACC.2
MOV P1.6, C
MOV C, ACC.1
MOV P1.5, C
MOV C, ACC.0
MOV P1.4, C
SETB P1.2
CLR P1.2
CALL delay
delay:
MOV R0, #50
DJNZ R0,$
ret
JMP MAIN
SMPCK:JB P1.6,LABEL5 ; sump
SETB P1.7
SJMP LABEL6
LABEL5:setb P3.0 ;motor off
CLR P1.7 ; sump low indicator
LABEL6:RET
END
```

Software Specifications:

Programming Language: The primary programming language used for this project is **Assembly Language** specifically designed for the 8051 microcontroller architecture.

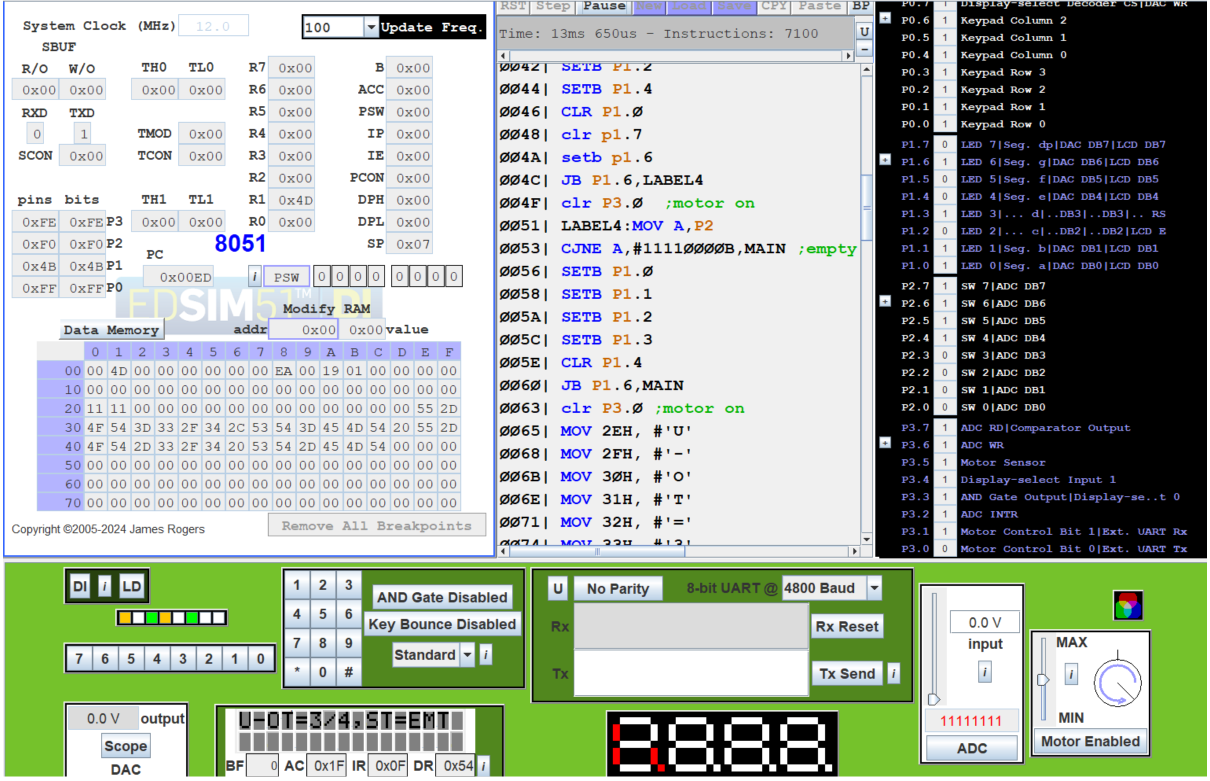
Development Environment: The project was implemented using the **Edsim51 Simulator**, an integrated development environment (IDE) that simulates the 8051 microcontroller.

Algorithm:

- Start
- If the sump tank is empty:
  - Turn on the sump indicator LED (P1.7).
  - Turn off the motor.
- If the sump tank is full, turn on LED (P1.6) and proceed to the next step.
- Read the water level in the overhead tank from Port 2.
- Determine the water level based on specific bit patterns (full, 3/4, 1/2, 1/4, or empty).
- If the overhead tank is 1/4th or empty:
  - Turn on the pump by clearing the motor control bit (P3.0).
- If the overhead tank is ½ or ¾ or full:
  - Turn off the pump by setting the motor control bit (P3.0).
- Based on the updated water level:
  - Set the corresponding LEDs (P1.0 to P1.4) / LCD to indicate the overhead tank level.
- End program

RESULTS

1. Overhead tank: empty, Sump tank: Full



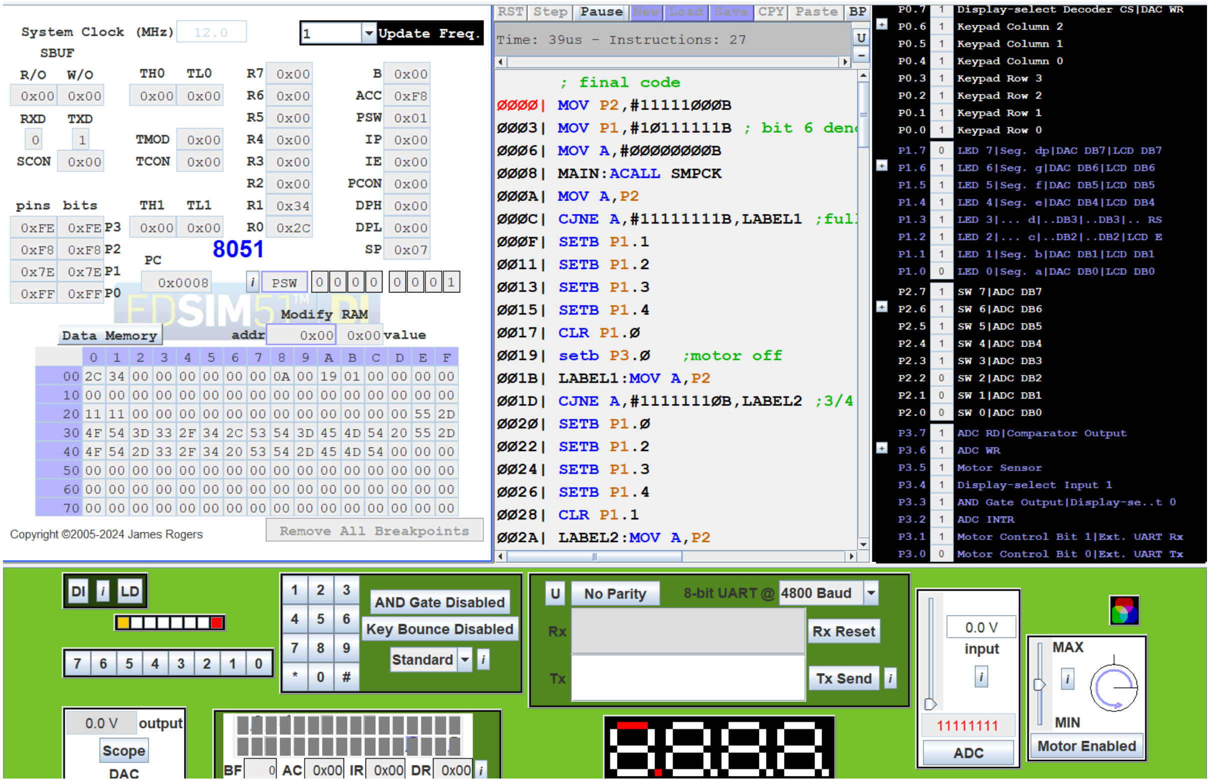
```
System Clock (MHz) 12.0      100   Update Freq.
SBUF
R/O   W/O      TH0   TL0    R7 0x00     B    0x00
0x00  0x00     0x00  0x00   R6 0x00     ACC  0x00
RXD   TXD                   R5 0x00     PSW  0x00
0     1        TMOD  0x00   R4 0x00     IP   0x00
SCON  0x00     TCON  0x00   R3 0x00     IE   0x00
                            R2 0x00     PCON 0x00
pins  bits     TH1   TL1    R1 0x4D     DPH  0x00
0xFE  0xFE P3  0x00  0x00   R0 0x00     DPL  0x00
0xF0  0xF0 P2          8051             SP   0x07
0x4B  0x4B P1  PC
0xFF  0xFF P0  0x00ED      PSW  0 0 0 0  0 0 0 0
                            Modify RAM
Data Memory          addr  0x00  0x00 value
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 00 4D 00 00 00 00 00 00 EA 00 19 01 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 11 11 00 00 00 00 00 00 00 00 00 00 00 00 55 2D
30 4F 54 3D 33 2F 34 2C 53 54 3D 45 4D 54 20 55 2D
40 4F 54 2D 33 2F 34 20 53 54 2D 45 4D 54 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Copyright ©2005-2024 James Rogers    Remove All Breakpoints
```

```
Time: 13ms 650us - Instructions: 7100
ØØ42| SETB P1.2
ØØ44| SETB P1.4
ØØ46| CLR P1.Ø
ØØ48| clr p1.7
ØØ4A| setb p1.6
ØØ4C| JB P1.6,LABEL4
ØØ4F| clr P3.Ø  ;motor on
ØØ51| LABEL4:MOV A,P2
ØØ53| CJNE A,#1111ØØØØB,MAIN ;empty
ØØ56| SETB P1.Ø
ØØ58| SETB P1.1
ØØ5A| SETB P1.2
ØØ5C| SETB P1.3
ØØ5E| CLR P1.4
ØØ6Ø| JB P1.6,MAIN
ØØ63| clr P3.Ø ;motor on
ØØ65| MOV 2EH, #'U'
ØØ68| MOV 2FH, #'-'
ØØ6B| MOV 3ØH, #'O'
ØØ6E| MOV 31H, #'T'
ØØ71| MOV 32H, #'='
ØØ74| MOV 33H, #'3'
```

```
U-OT=3/4,ST=EMT
BF 0   AC 0x1F   IR 0x0F   DR 0x54
DAC 0.0 V output     ADC 0.0 V input 11111111     Motor Enabled
No Parity   8-bit UART @ 4800 Baud
```

RST
Step
Pause
New
Load
Save
CPY
Paste
BP
P0.7 Display-select Decoder CS|DAC WR
P0.6 Keypad Column 2
P0.5 Keypad Column 1
P0.4 Keypad Column 0
P0.3 Keypad Row 3
P0.2 Keypad Row 2
P0.1 Keypad Row 1
P0.0 Keypad Row 0
P1.7 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 LED 3|... d|..DB3|..DB3|.. RS
P1.2 LED 2|... c|..DB2|..DB2|LCD E
P1.1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 SW 7|ADC DB7
P2.6 SW 6|ADC DB6
P2.5 SW 5|ADC DB5
P2.4 SW 4|ADC DB4
P2.3 SW 3|ADC DB3
P2.2 SW 2|ADC DB2
P2.1 SW 1|ADC DB1
P2.0 SW 0|ADC DB0
P3.7 ADC RD|Comparator Output
P3.6 ADC WR
P3.5 Motor Sensor
P3.4 Display-select Input 1
P3.3 AND Gate Output|Display-se..t 0
P3.2 ADC INTR
P3.1 Motor Control Bit 1|Ext. UART Rx
P3.0 Motor Control Bit 0|Ext. UART Tx
DI
LD
7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 * 0 #
AND Gate Disabled
Key Bounce Disabled
Standard
Rx
Tx
Rx Reset
Tx Send
Scope
MAX
MIN

2. Overhead tank: 1/4, Sump tank: Full



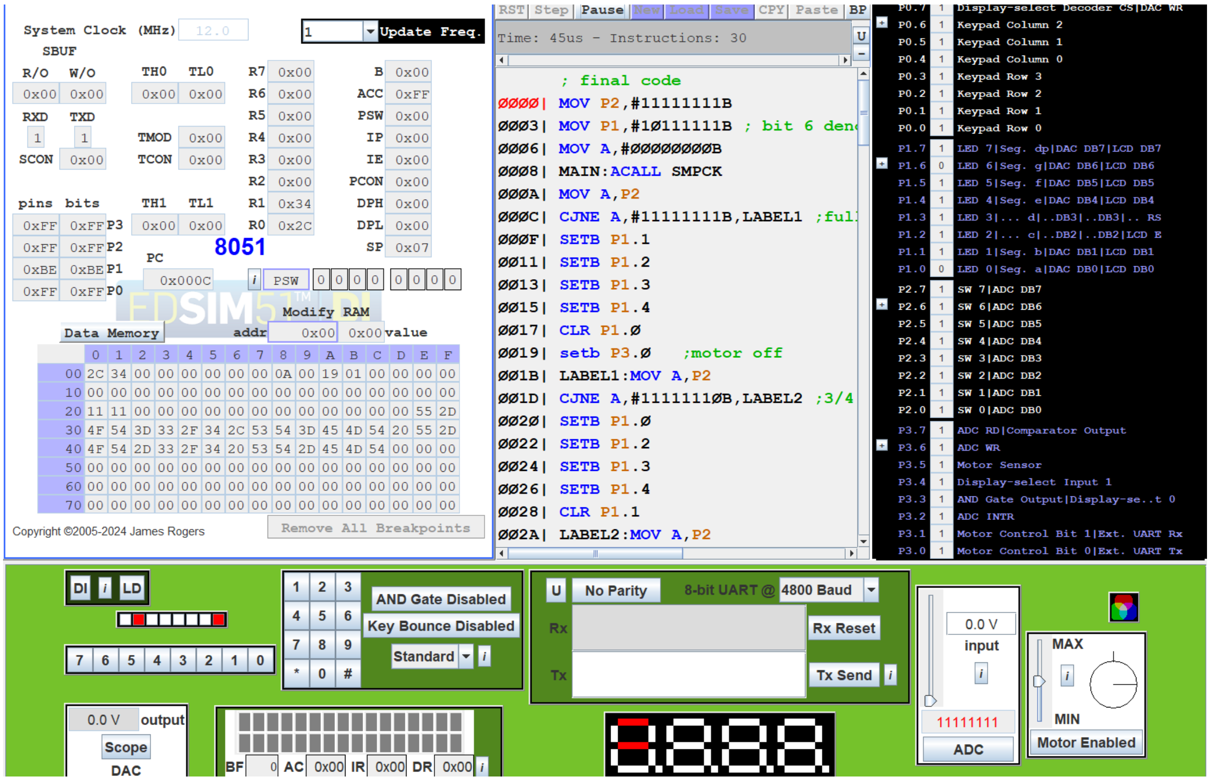
```
System Clock (MHz) 12.0      1     Update Freq.
SBUF
R/O   W/O      TH0   TL0    R7 0x00     B    0x00
0x00  0x00     0x00  0x00   R6 0x00     ACC  0xF8
RXD   TXD                   R5 0x00     PSW  0x01
0     1        TMOD  0x00   R4 0x00     IP   0x00
SCON  0x00     TCON  0x00   R3 0x00     IE   0x00
                            R2 0x00     PCON 0x00
pins  bits     TH1   TL1    R1 0x34     DPH  0x00
0xFE  0xFE P3  0x00  0x00   R0 0x2C     DPL  0x00
0xF8  0xF8 P2          8051             SP   0x07
0x7E  0x7E P1  PC
0xFF  0xFF P0  0x0008      PSW  0 0 0 0  0 0 0 1
                            Modify RAM
Data Memory          addr  0x00  0x00 value
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 2C 34 00 00 00 00 00 00 0A 00 19 01 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 11 11 00 00 00 00 00 00 00 00 00 00 00 00 55 2D
30 4F 54 3D 33 2F 34 2C 53 54 3D 45 4D 54 20 55 2D
40 4F 54 2D 33 2F 34 20 53 54 2D 45 4D 54 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Copyright ©2005-2024 James Rogers    Remove All Breakpoints
```

```
Time: 39us - Instructions: 27
      ; final code
ØØØØ| MOV P2,#11111ØØØB
ØØØ3| MOV P1,#1Ø111111B ; bit 6 den
ØØØ6| MOV A,#ØØØØØØØØB
ØØØ8| MAIN:ACALL SMPCK
ØØØA| MOV A,P2
ØØØC| CJNE A,#11111111B,LABEL1 ;ful
ØØØF| SETB P1.1
ØØ11| SETB P1.2
ØØ13| SETB P1.3
ØØ15| SETB P1.4
ØØ17| CLR P1.Ø
ØØ19| setb P3.Ø   ;motor off
ØØ1B| LABEL1:MOV A,P2
ØØ1D| CJNE A,#1111111ØB,LABEL2 ;3/4
ØØ2Ø| SETB P1.Ø
ØØ22| SETB P1.2
ØØ24| SETB P1.3
ØØ26| SETB P1.4
ØØ28| CLR P1.1
ØØ2A| LABEL2:MOV A,P2
```

```
BF 0   AC 0x00   IR 0x00   DR 0x00
DAC 0.0 V output     ADC 0.0 V input 11111111     Motor Enabled
No Parity   8-bit UART @ 4800 Baud
```

RST
Step
Pause
New
Load
Save
CPY
Paste
BP
P0.7 Display-select Decoder CS|DAC WR
P0.6 Keypad Column 2
P0.5 Keypad Column 1
P0.4 Keypad Column 0
P0.3 Keypad Row 3
P0.2 Keypad Row 2
P0.1 Keypad Row 1
P0.0 Keypad Row 0
P1.7 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 LED 3|... d|..DB3|..DB3|.. RS
P1.2 LED 2|... c|..DB2|..DB2|LCD E
P1.1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 SW 7|ADC DB7
P2.6 SW 6|ADC DB6
P2.5 SW 5|ADC DB5
P2.4 SW 4|ADC DB4
P2.3 SW 3|ADC DB3
P2.2 SW 2|ADC DB2
P2.1 SW 1|ADC DB1
P2.0 SW 0|ADC DB0
P3.7 ADC RD|Comparator Output
P3.6 ADC WR
P3.5 Motor Sensor
P3.4 Display-select Input 1
P3.3 AND Gate Output|Display-se..t 0
P3.2 ADC INTR
P3.1 Motor Control Bit 1|Ext. UART Rx
P3.0 Motor Control Bit 0|Ext. UART Tx
DI
LD
7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 * 0 #
AND Gate Disabled
Key Bounce Disabled
Standard
Rx
Tx
Rx Reset
Tx Send
Scope
MAX
MIN

3. Overhead tank: Full, Sump tank: Full



```
System Clock (MHz) 12.0      1  Update Freq.
SBUF
R/O   W/O     TH0   TL0     R7 0x00      B    0x00
0x00  0x00    0x00  0x00    R6 0x00      ACC  0xFF
RXD   TXD                   R5 0x00      PSW  0x00
1     1       TMOD  0x00    R4 0x00      IP   0x00
SCON  0x00    TCON  0x00    R3 0x00      IE   0x00
                            R2 0x00      PCON 0x00
pins  bits    TH1   TL1     R1 0x34      DPH  0x00
0xFF  0xFF P3 0x00  0x00    R0 0x2C      DPL  0x00
0xFF  0xFF P2               8051         SP   0x07
0xBE  0xBE P1    PC
0xFF  0xFF P0    0x000C     PSW  0 0 0 0 0 0 0 0
                                 Modify RAM
Data Memory              addr  0x00   0x00 value

    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 2C 34 00 00 00 00 00 0A 00 19 01 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 11 11 00 00 00 00 00 00 00 00 00 00 00 00 55 2D
30 4F 54 3D 33 2F 34 2C 53 54 3D 45 4D 54 20 55 2D
40 4F 54 2D 33 2F 34 20 53 54 2D 45 4D 54 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2024 James Rogers
```

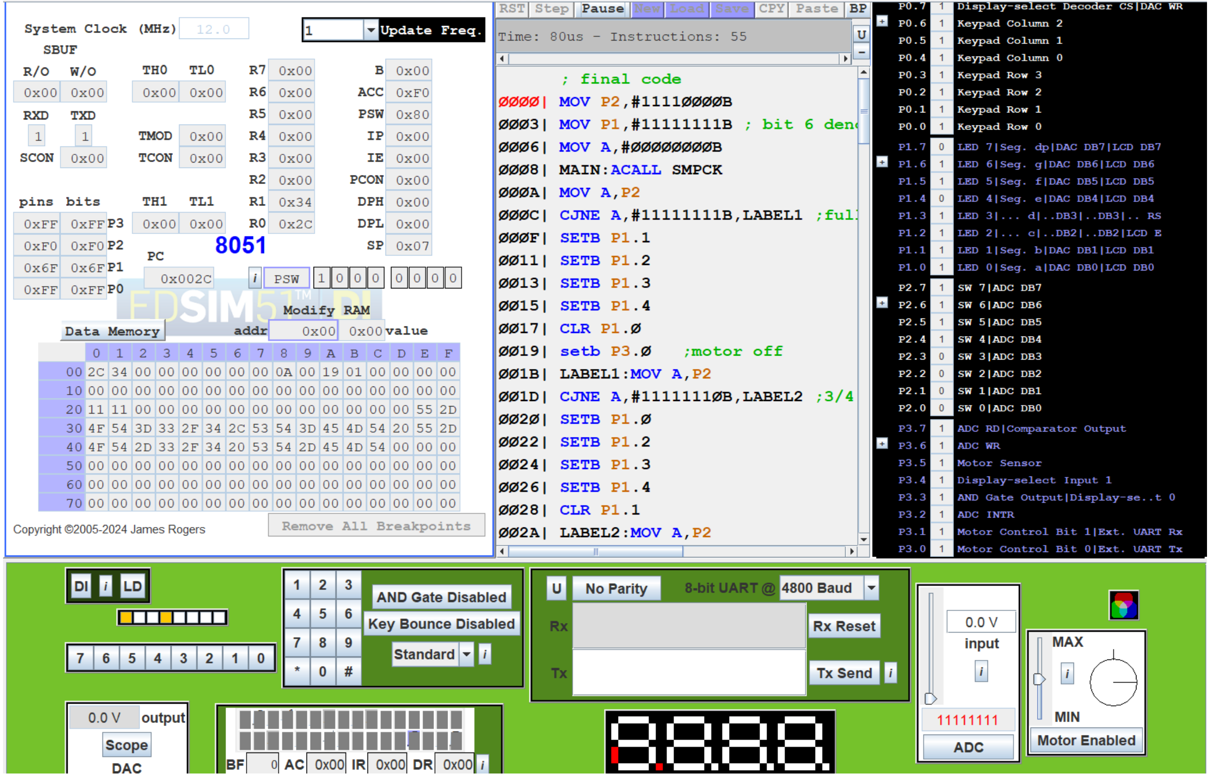
```
Time: 45us - Instructions: 30

       ; final code
ØØØØ| MOV P2,#11111111B
ØØØ3| MOV P1,#1Ø111111B ; bit 6 den
ØØØ6| MOV A,#ØØØØØØØØB
ØØØ8| MAIN:ACALL SMPCK
ØØØA| MOV A,P2
ØØØC| CJNE A,#11111111B,LABEL1 ;ful
ØØØF| SETB P1.1
ØØ11| SETB P1.2
ØØ13| SETB P1.3
ØØ15| SETB P1.4
ØØ17| CLR P1.Ø
ØØ19| setb P3.Ø    ;motor off
ØØ1B| LABEL1:MOV A,P2
ØØ1D| CJNE A,#1111111ØB,LABEL2 ;3/4
ØØ2Ø| SETB P1.Ø
ØØ22| SETB P1.2
ØØ24| SETB P1.3
ØØ26| SETB P1.4
ØØ28| CLR P1.1
ØØ2A| LABEL2:MOV A,P2
```

```
P0.7 1 Display-select Decoder CS|DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 1 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 0 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 1 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 1 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 1 LED 3|... d|..DB3|..DB3|.. RS
P1.2 1 LED 2|... c|..DB2|..DB2|LCD E
P1.1 1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 0 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 1 SW 7|ADC DB7
P2.6 1 SW 6|ADC DB6
P2.5 1 SW 5|ADC DB5
P2.4 1 SW 4|ADC DB4
P2.3 1 SW 3|ADC DB3
P2.2 1 SW 2|ADC DB2
P2.1 1 SW 1|ADC DB1
P2.0 1 SW 0|ADC DB0
P3.7 1 ADC RD|Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output|Display-se..t 0
P3.2 1 ADC INTR
P3.1 1 Motor Control Bit 1|Ext. UART Rx
P3.0 1 Motor Control Bit 0|Ext. UART Tx
```

```
7 6 5 4 3 2 1 0        1 2 3 / 4 5 6 / 7 8 9 / * 0 #
AND Gate Disabled   Key Bounce Disabled   Standard
U  No Parity   8-bit UART @ 4800 Baud
Rx   Tx
0.0 V input   11111111   ADC
Motor Enabled
0.0 V output
BF 0  AC 0x00  IR 0x00  DR 0x00
```

4. Overhead tank: Empty, Sump tank: Empty



```
System Clock (MHz) 12.0      1  Update Freq.
SBUF
R/O   W/O     TH0   TL0     R7 0x00      B    0x00
0x00  0x00    0x00  0x00    R6 0x00      ACC  0xF0
RXD   TXD                   R5 0x00      PSW  0x80
1     1       TMOD  0x00    R4 0x00      IP   0x00
SCON  0x00    TCON  0x00    R3 0x00      IE   0x00
                            R2 0x00      PCON 0x00
pins  bits    TH1   TL1     R1 0x34      DPH  0x00
0xFF  0xFF P3 0x00  0x00    R0 0x2C      DPL  0x00
0xF0  0xF0 P2               8051         SP   0x07
0x6F  0x6F P1    PC
0xFF  0xFF P0    0x002C     PSW  1 0 0 0 0 0 0 0
                                 Modify RAM
Data Memory              addr  0x00   0x00 value

    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 2C 34 00 00 00 00 00 0A 00 19 01 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 11 11 00 00 00 00 00 00 00 00 00 00 00 00 55 2D
30 4F 54 3D 33 2F 34 2C 53 54 3D 45 4D 54 20 55 2D
40 4F 54 2D 33 2F 34 20 53 54 2D 45 4D 54 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2024 James Rogers
```

```
Time: 80us - Instructions: 55

       ; final code
ØØØØ| MOV P2,#1111ØØØØB
ØØØ3| MOV P1,#11111111B ; bit 6 den
ØØØ6| MOV A,#ØØØØØØØØB
ØØØ8| MAIN:ACALL SMPCK
ØØØA| MOV A,P2
ØØØC| CJNE A,#11111111B,LABEL1 ;ful
ØØØF| SETB P1.1
ØØ11| SETB P1.2
ØØ13| SETB P1.3
ØØ15| SETB P1.4
ØØ17| CLR P1.Ø
ØØ19| setb P3.Ø    ;motor off
ØØ1B| LABEL1:MOV A,P2
ØØ1D| CJNE A,#1111111ØB,LABEL2 ;3/4
ØØ2Ø| SETB P1.Ø
ØØ22| SETB P1.2
ØØ24| SETB P1.3
ØØ26| SETB P1.4
ØØ28| CLR P1.1
ØØ2A| LABEL2:MOV A,P2
```

```
P0.7 1 Display-select Decoder CS|DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 0 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 1 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 1 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 0 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 1 LED 3|... d|..DB3|..DB3|.. RS
P1.2 1 LED 2|... c|..DB2|..DB2|LCD E
P1.1 1 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 1 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 1 SW 7|ADC DB7
P2.6 1 SW 6|ADC DB6
P2.5 1 SW 5|ADC DB5
P2.4 1 SW 4|ADC DB4
P2.3 0 SW 3|ADC DB3
P2.2 0 SW 2|ADC DB2
P2.1 0 SW 1|ADC DB1
P2.0 0 SW 0|ADC DB0
P3.7 1 ADC RD|Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output|Display-se..t 0
P3.2 1 ADC INTR
P3.1 1 Motor Control Bit 1|Ext. UART Rx
P3.0 1 Motor Control Bit 0|Ext. UART Tx
```

```
7 6 5 4 3 2 1 0        1 2 3 / 4 5 6 / 7 8 9 / * 0 #
AND Gate Disabled   Key Bounce Disabled   Standard
U  No Parity   8-bit UART @ 4800 Baud
Rx   Tx
0.0 V input   11111111   ADC
Motor Enabled
0.0 V output
BF 0  AC 0x00  IR 0x00  DR 0x00
```

CONCLUSION

The project "Water Level Controller Using 8051 Microcontroller Assembly Language Programming" successfully developed an automated water level management system using the 8051 microcontroller. This system monitors the water level in an overhead tank and activates a water pump when needed, providing visual feedback through LEDs.

Major Findings:

1. The system effectively eliminates manual monitoring, preventing overflow and reducing water wastage by automatically activating or deactivating the pump based on water level thresholds.
2. Integration with water level sensors allows for timely actions to maintain optimal water supply without human intervention.
3. The project demonstrates enhanced water management, making it applicable in households, agriculture, and industrial settings.
4. Testing confirmed that the system responds accurately to various water levels, with clear LED indicators improving usability.
5. This project serves as a practical introduction to microcontroller programming and embedded systems, illustrating the importance of low-level programming in automation.

In summary, the project highlights the potential of microcontroller-based solutions for efficient resource management and automation in diverse applications.

REFERENCES & CONTRIBUTION

- https://edsim51.com/
- https://www.circuitstoday.com/water-level-controller-using-8051
- 8051 manuals

Purv Meghpuria – Implementation, Report, Algorithm, Presentation
Zeeshan Khan – Logic Diagram, Truth Table, Presentation, Implementation
Luv Partani – Presentation
Aditya Sharma – Presentation