**Automated Hydroponic Cattle Feed Generator**



CEP Report

By

|  |  |
| --- | --- |
| NAME | Registration Number |
| Sheikh Ata Ur Rehman Saqib | CUI/FA22-BCE-061 /ATD |
| Syed Iftikhar Shah | CUI/FA22-BCE-057 /ATD |
| Sahibzada Daniyal | CUI/FA22-BCE-082 /ATD |

**CPE-342 Microprocessor Systems & Interfacing**

Semester Spring 2025

Supervised by: Muhammad Naveed Shaikh

CEP Course Teacher’s Name: Muhammad Naveed Shaikh

Department of Computer Engineering

COMSATS University Islamabad – Abbottabad Campus

**DECLARATION**

We Sheikh Ata Ur Rehman Saqib (CUI/FA22-BCE-061/ATD), Syed Iftikhar Shah (CUI/FA22-BCE-057/ATD) and Sahibzada Daniyal (CUI/ FA22-BCE-082/ ATD) hereby declare that we have produced the work presented in this report, during the scheduled period of study. We also declare that we have not taken any material from any source except referred to wherever due. If a violation of rules has occurred in this report, we shall be liable to punishable action.

Date: 20 July, 2025

Sheikh Ata Ur Rehman Saqib

Student 1

(CUI/ FA22-BCE 061/ATD)

Syed Iftikhar Shah

Student 2

(CUI/ FA22-BCE 057/ATD)

Sahibzada Daniyal

Student 3

(CUI/ FA22-BCE 082/ATD)

**ABSTRACT**

This project presents a fully automated hydroponic cattle feed generator that enhances sustainable livestock feeding practices using embedded systems. By leveraging the PIC18F452 microcontroller, programmed entirely in assembly language, the system controls the entire fodder growth cycle. It automates watering intervals, tray movements, and provides sensor-based monitoring for water level and tray positioning. The setup includes a 10-tray inclined structure with integrated sensors, motors, and LCD feedback, offering a cost-effective, low-maintenance solution for consistent green fodder production. The system ensures timely operations using internal timers and interrupt-based control logic, achieving a practical, reliable automation model.

**TABLE OF CONTENTS**

[1 Introduction 7](#_Toc139461550)

[2 Literature Survey 8](#_Toc139461551)

[3 Proposed Methodology 8](#_Toc139461552)

[4 Simulation Results 11](#_Toc139461553)

[5. Conclusions 14](#_Toc139461554)

[6. References 14](#_Toc139461555)

[7. Appendix 14](#_Toc139461556)

**LIST OF FIGURES**

Fig: 1.1 Hydroponic Cattle Feed Generator 7

Fig: 3.1 System Design N/A

Fig: 3.2 Hardware model 09

# Fig: 3.4 Flow Chart Diagram…………………………………………………………………….4

Fig: 3.3 Circuit Diagram 10

Fig: 4.1 Simulation results1 11

Fig: 4.2 Simulation results2 11

Fig: 4.3 Simulation results3 12

Fig: 4.4 Simulation results4 12

Fig: 4.5 Simulation results5 13

**LIST OF ABBREVIATIONS**

* **LCD** - Liquid Crystal Display
* **MCU** - Microcontroller Unit
* **RTC** - Real Time Clock
* **ISR** - Interrupt Service Routine.

# Introduction

Modern dairy farming requires consistent access to fresh fodder. Traditional methods depend on manual labor and are often impacted by weather and resource availability. Our proposed system addresses these challenges by automating the production of hydroponic cattle feed using a microcontroller-based setup. It uses sensors and actuators to control water pumps and motors, providing real-time automation and minimal human intervention.

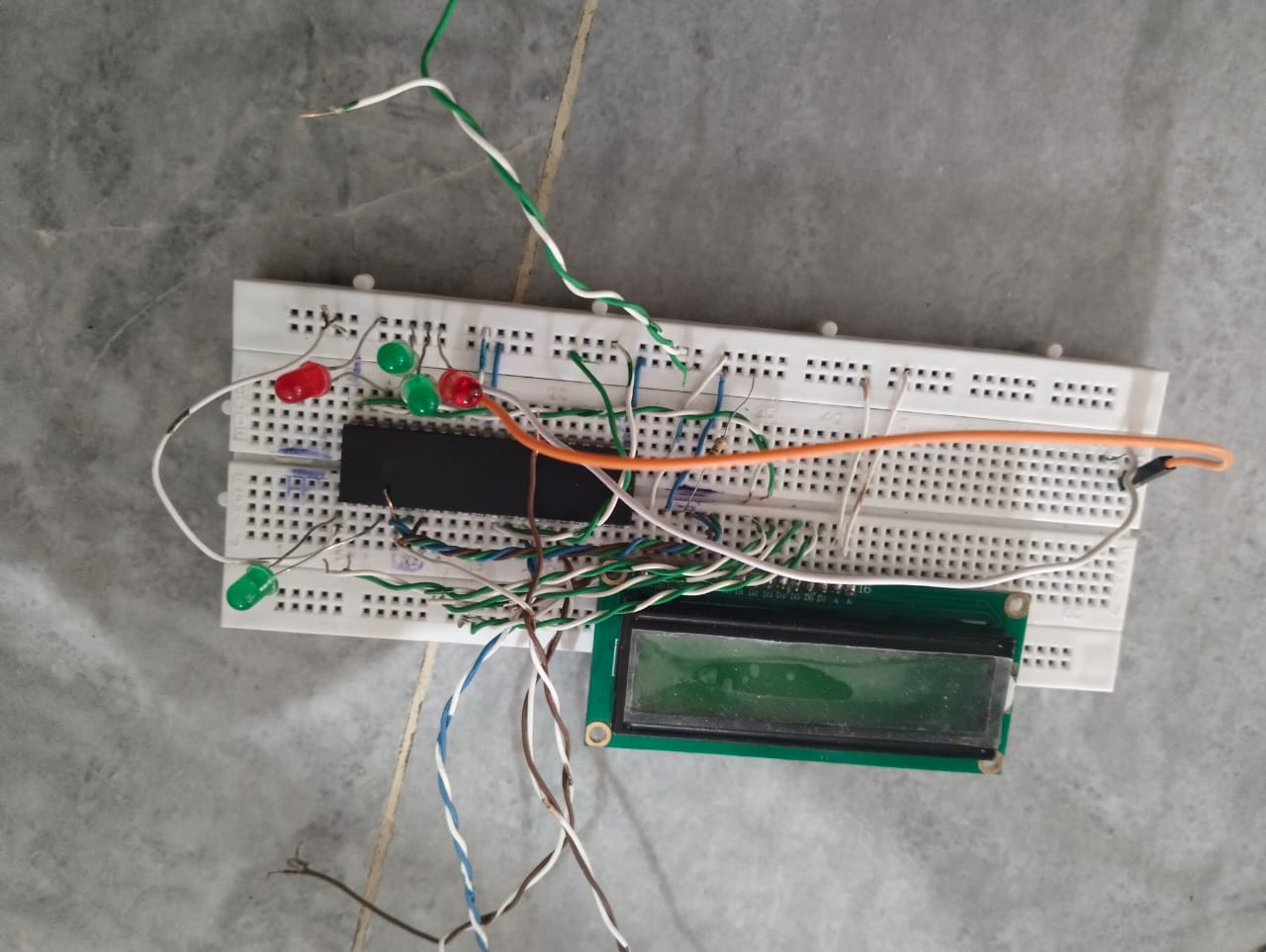


Fig: 1.1 Hydroponic Cattle Feed Generator

* 1. **Objectives**
* To design a fully automatic hydroponic feed system.
* To enhance control through assembly-programmed microcontroller.
* To integrate sensors and actuators with reliable performance.
* To enable timing precision for watering and tray movements.
  1. **Features and Cost Estimate of our Project**

**Features:**

* Real-time monitoring.
* Visual alert on LCD.
* Threshold-based activation.

**Estimated Cost:**

* MQ-2 Sensor: PKR 600
* PIC18F452: PKR 2500
* LCD (16x2): PKR 400
* LED (4): 100
* Miscellaneous (wires, breadboard): PKR 800

**Total Estimated Cost:** PKR 4400

# Literature Survey

Various automated farming systems have been proposed using microcontrollers and IoT modules. However, hydroponic automation using PIC18F452 with full assembly programming is less common. This project fills that gap with a resource-efficient solution. Hydroponic fodder systems typically use sensors for water levels and motorized systems for tray shifting. Integrating all modules under precise control using PIC18F452 enables cost-effective and scalable development.

* 1. **Microcontroller-Based Safety Systems**  
     Embedded systems utilizing PIC microcontrollers provide reliable processing and easy integration of sensors and actuators.

# Proposed Methodology

* 1. **Mathematical Model**

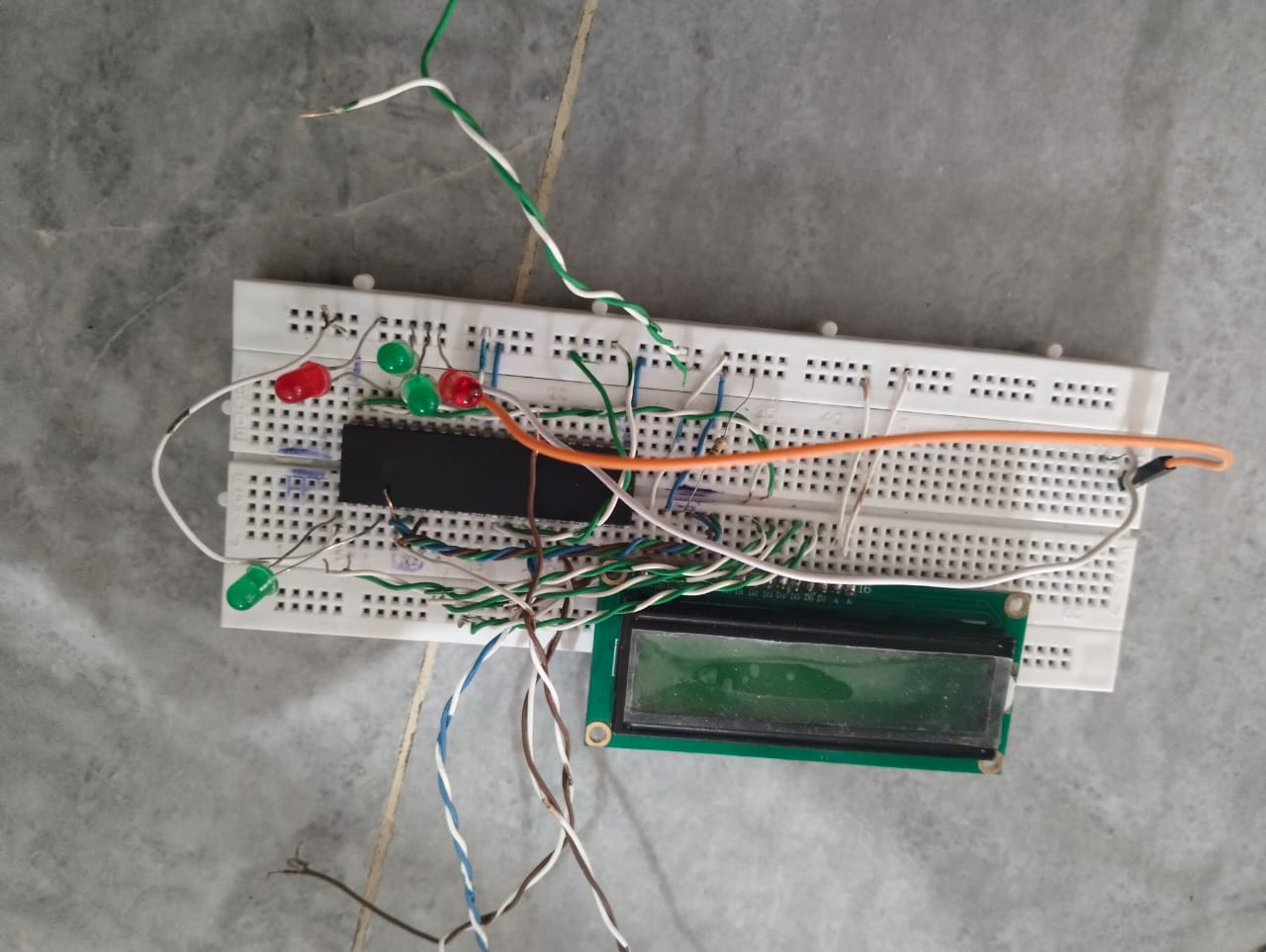
N/A

* 1. **System Design / Block diagram**

N/A

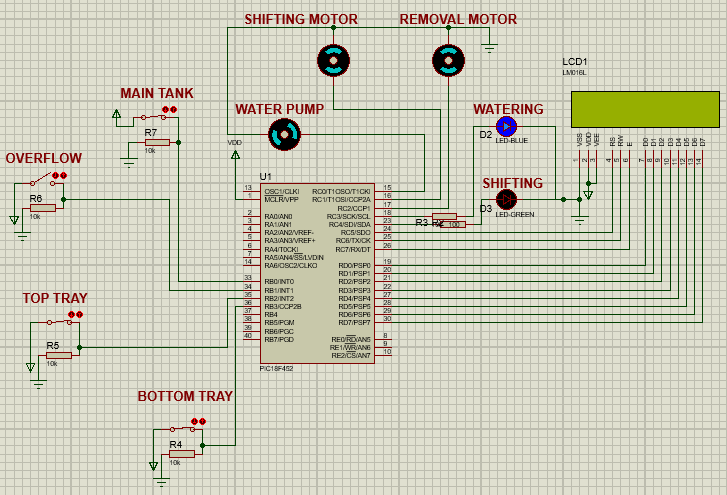
# Fig: 3.1 System Design

* 1. **Simulation/hardware model (if any)**



# Fig: 3.2 Hardware model

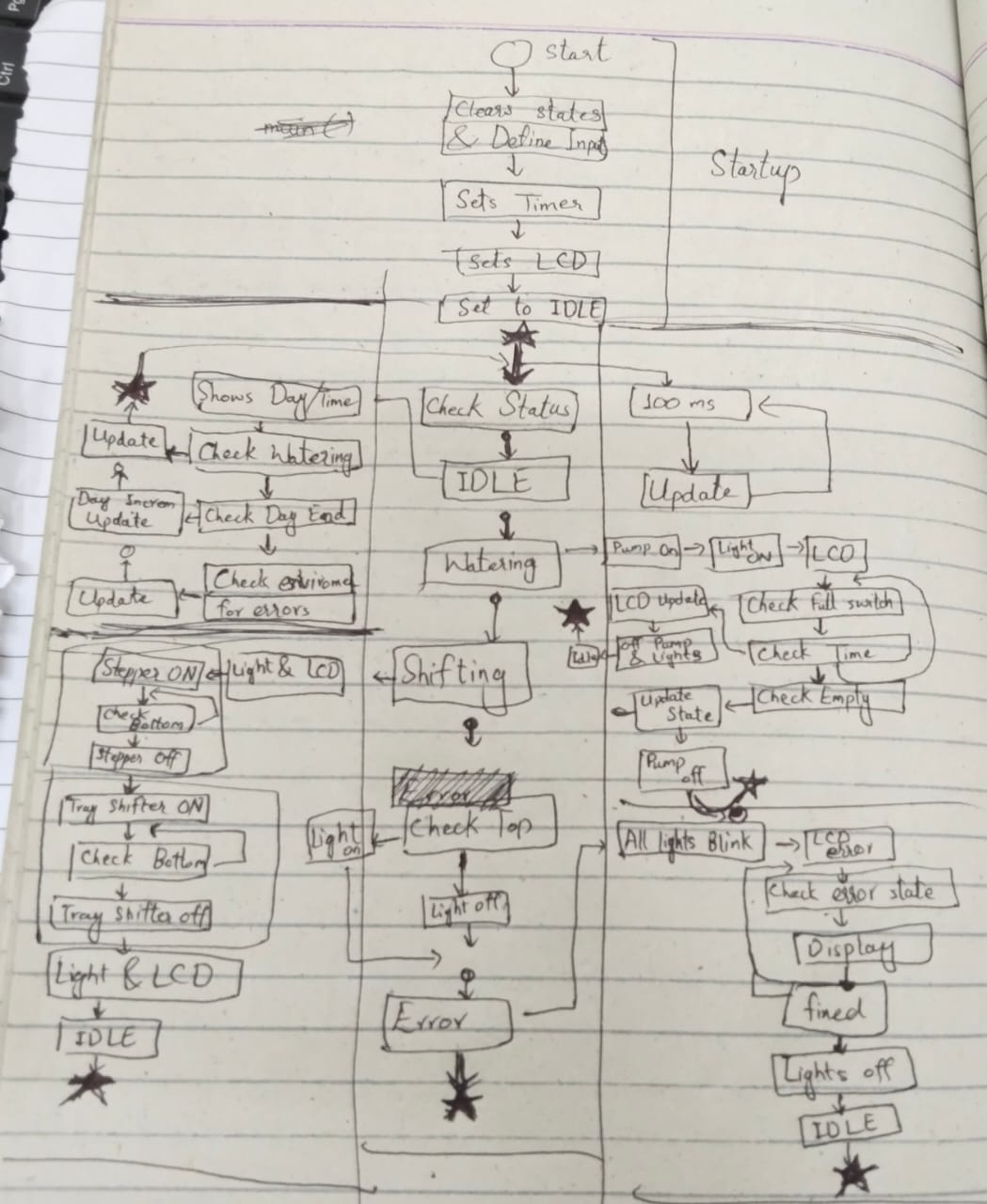
* 1. **Circuit diagram / components / simulation diagram (if any)**



# Fig: 3.3 Simulation Diagram

* 1. **Flow chart / Pseudocode / Algorithm … (if any)**

**Flow Chart:**

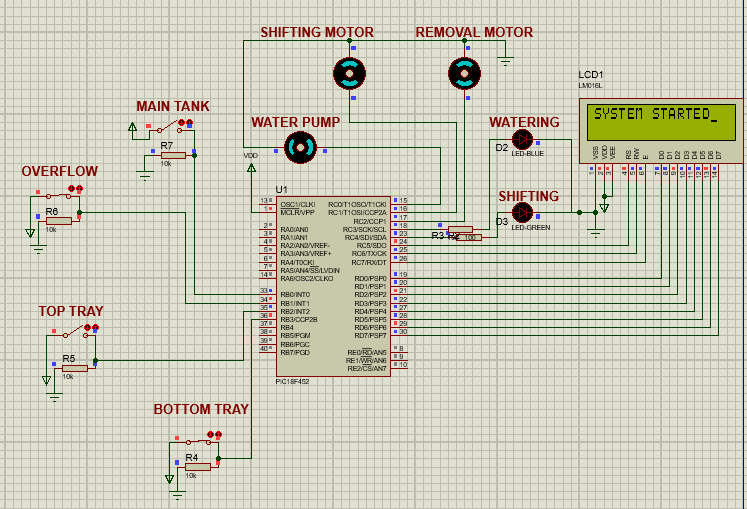


# Fig: 3.4 Flow Chart Diagram

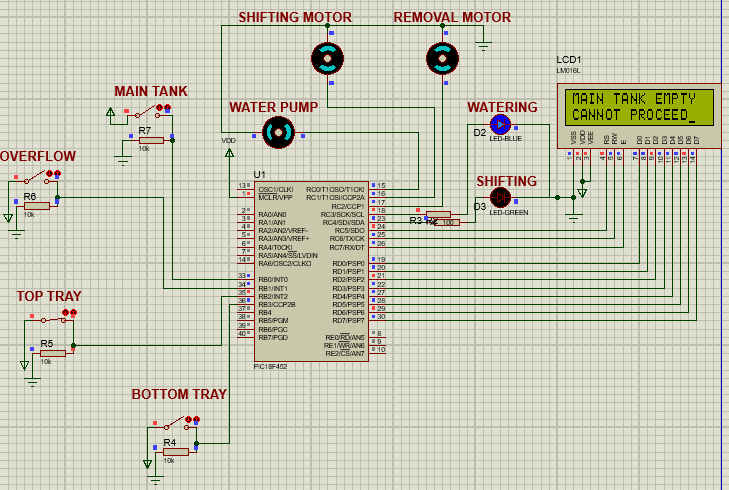
# Simulation Results

* 1. **Hardware/software simulation results**

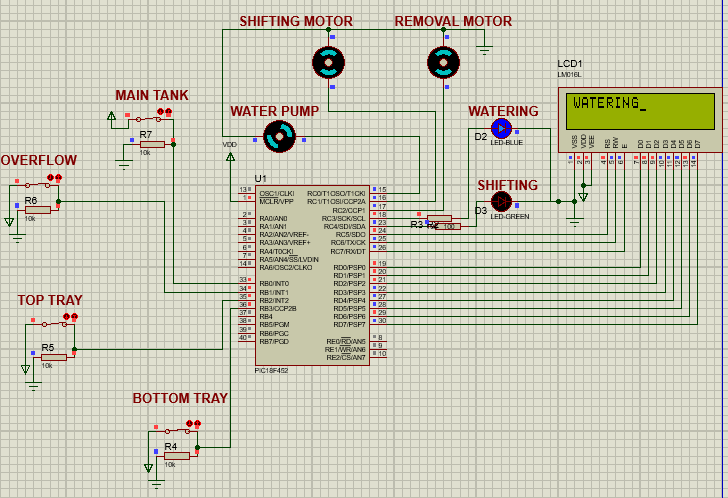
Figures should be properly numbered and with proper captions

****

# Fig: 4.1 simulation results1



# Fig: 4.2 simulation results2



# Fig: 4.3 simulation results3

A diagram of a machine

AI-generated content may be incorrect.

# Fig: 4.4 simulation results4

A diagram of a machine

AI-generated content may be incorrect.

**Fig: 4.5 simulation results5**

* 1. **Design / simulation parameters**

**N/A**

* 1. **Comparison with Analytical Solution (if any)**

**Alternative Solutions:**

**1. Traditional Soil-Based Fodder Farming**

**-** Why not? Requires more land, water, time (7–8 weeks vs. 7 days), and is vulnerable to weather/disease.

**2. Commercial Feed Pellets**

**-** Why not? Expensive, processed, lacks freshness/natural enzymes, may have preservatives.

**Why You Chose Hydroponic System:**

- **Fast Growth:** Ready in 7 days.

- **Water-Efficient:** Uses 80–90% less water than traditional farming.

**- No Soil Needed:** Can be used in any climate or location.

**- Space-Saving:** Vertical trays allow small-area production.

**- Clean & Nutritious:** Fresh, chemical-free, enzyme-rich feed.

* 1. **Discussions etc…**

Simulation validated functionality before hardware implementation.

# Conclusions

The Automated Hydroponic Cattle Feed Generator efficiently demonstrates embedded control of a hydroponic system using PIC18F452. Implemented entirely in assembly language, the system provides precise timing, reliable automation, and scalability. With consistent tray shifting and timely watering cycles, this prototype showcases practical embedded system design for sustainable agriculture.

# References

Should be in standard IEEE format…

[1] Microchip Technology, "PIC18F452 Datasheet," <https://www.microchip.com/wwwproducts/en/PIC18F452>

[2] Microchip Technology, "MPLAB X IDE Documentation," https://www.microchip.com/mplab/mplab-x-ide

# Appendix

LIST P=PIC1F452, F=INHX32, N=0, ST=OFF, R=HEX

config OSC=HS, OSCS=OFF, WDT=OFF, BORV=45, PWRT=ON, BOR=ON, DEBUG=OFF,LVP=OFF, STVR=OFF

#INCLUDE P18F452.INC

ORG 00H

GOTO MAIN

;-------------------------------------------

ORG 08H

GOTO HIGH\_ISR

;-------------------------------------------

ORG 2AH

MAIN

;---------------FILE REGISTERS--------------

STATUS\_TEMP\_SAVE EQU 20H

WREG\_TEMP\_SAVE EQU 21H

BSR\_TEMP\_SAVE EQU 22H

CURRENT\_STATE EQU 23H

WATERING\_COUNTER EQU 24H

DAY\_COUNTER\_L EQU 25H

DAY\_COUNTER\_H EQU 26H

;---------------INPUT LABELS----------------

INPUTS EQU PORTB

WATER\_LEVEL\_LOW EQU RB0

WATER\_OVERFLOW EQU RB1

TRAY\_TOP EQU RB2

TRAY\_BOTTOM EQU RB3

;---------------OUTPUT LABELS---------------

OUTPUTS EQU PORTC

WATER\_PUMP EQU RC0

SHIFTING\_MOTOR EQU RC1

REMOVAL\_MOTOR EQU RC2

LED\_WATERING EQU RC3

LED\_SHIFTING EQU RC4

LCD\_DATA EQU PORTD

LCD\_RS EQU RC5

LCD\_RW EQU RC6

LCD\_EN EQU RC7

;----------------I/O SETTING----------------

SETF TRISB

CLRF TRISC

CLRF TRISD

;----------------PIC STATES-----------------

IDLE EQU 00H

WATERING EQU 01H

SHIFTING EQU 02H

;===========================================

;-----SETTING TIMER FOR 8.3s INTERRUPTS-----

MOVLW B'00000110'

MOVWF T0CON

CLRF TMR0H

CLRF TMR0L

BCF INTCON,TMR0IF

BSF INTCON,TMR0IE

BSF INTCON,GIE

BSF T0CON,TMR0ON

;---------CLEARING VARIABLES FIRST TIME-----

CLRF CURRENT\_STATE

CLRF WATERING\_COUNTER

CLRF DAY\_COUNTER\_L

CLRF DAY\_COUNTER\_H

;-------------------------------------------

CALL LCD\_INIT

;DISPLAYING OPENING MESSAGE

CALL DISPLAY\_STARTING\_MESSAGE

CALL WAIT

;================MAIN LOOP==================

MAIN\_LOOP

BCF OUTPUTS,LED\_WATERING

BCF OUTPUTS,LED\_SHIFTING

MOVF CURRENT\_STATE, W

XORLW IDLE

BTFSC STATUS,Z

GOTO IDLE\_STATE

MOVF CURRENT\_STATE,W

XORLW WATERING

BTFSC STATUS,Z

GOTO WATERING\_STATE

MOVF CURRENT\_STATE,W

XORLW SHIFTING

BTFSC STATUS,Z

GOTO SHIFTING\_STATE

GOTO MAIN\_LOOP

;===========================================

IDLE\_STATE

; Check Main Tank Empty

;BEING ON MEANS WATER LEVEL IS HIGHER THAN THIS

BTFSS INPUTS,WATER\_LEVEL\_LOW

GOTO DISPLAY\_TANK\_EMPTY

; Check for Top Tray Missing

BTFSC INPUTS,TRAY\_TOP

GOTO L1

CALL DISPLAY\_TOP\_MISSING

GOTO L2

L1

CALL DISPLAY\_IDLE\_MESSAGE

L2

; EVERYTHING IS OK, proceed

BCF OUTPUTS,LED\_WATERING

BCF OUTPUTS,LED\_SHIFTING

CALL CHECK\_PUMP\_TIMER

MOVLW WATERING

CPFSEQ CURRENT\_STATE

CALL CHECK\_DAY\_TIMER

GOTO MAIN\_LOOP

;-----------------------------------------

CHECK\_PUMP\_TIMER

;SINCE 15\*8s=120s OR 2 min

MOVF WATERING\_COUNTER,W

SUBLW 04H ; 0x0F - WATERING\_COUNTER

BTFSC STATUS,C ; If C=1, then literal >= WREG

RETURN

MOVLW WATERING

MOVWF CURRENT\_STATE

RETURN

;------------------------------------------

CHECK\_DAY\_TIMER

;42\*256 + 48 = 10800\*8 = 24 hours

;MOVLW 29H

;CPFSGT DAY\_COUNTER\_H

;GOTO MAIN\_LOOP

;MOVLW 2FH

;CURRENTLY USING ONLY 6 MINUTES TIMER

MOVLW 08H

CPFSGT DAY\_COUNTER\_L

RETURN

MOVLW SHIFTING

MOVWF CURRENT\_STATE

CLRF DAY\_COUNTER\_H

CLRF DAY\_COUNTER\_L

RETURN

;-------------------------------------------

WATERING\_STATE

BTFSS INPUTS,WATER\_LEVEL\_LOW

GOTO DISPLAY\_TANK\_EMPTY

BCF OUTPUTS,LED\_SHIFTING

BSF OUTPUTS,LED\_WATERING

BSF OUTPUTS,WATER\_PUMP

WATERING\_IN\_PROGRESS

CALL DISPLAY\_WATERING

BTFSC INPUTS,WATER\_OVERFLOW

GOTO WATERING\_DONE

BTFSC INPUTS,WATER\_LEVEL\_LOW

GOTO WATERING\_IN\_PROGRESS

BCF OUTPUTS,WATER\_PUMP

MOVLW IDLE

MOVWF CURRENT\_STATE

GOTO DISPLAY\_TANK\_EMPTY

WATERING\_DONE

BCF OUTPUTS,WATER\_PUMP

CALL DISPLAY\_WATERING\_DONE

CLRF WATERING\_COUNTER

MOVLW IDLE

MOVWF CURRENT\_STATE

GOTO MAIN\_LOOP

;-------------------------------------------

SHIFTING\_STATE

BSF OUTPUTS,LED\_SHIFTING

BTFSS INPUTS,TRAY\_BOTTOM

GOTO SHIFT\_DOWN

CALL DISPLAY\_TRAY\_REMOVAL

BSF OUTPUTS,REMOVAL\_MOTOR

REMOVAL\_IN\_PROGRESS BTFSC INPUTS,TRAY\_BOTTOM

GOTO REMOVAL\_IN\_PROGRESS

BCF OUTPUTS, REMOVAL\_MOTOR

SHIFT\_DOWN

CALL DISPLAY\_TRAY\_SHIFTING

BSF OUTPUTS,SHIFTING\_MOTOR

SHIFTING\_IN\_PROGRESS BTFSS INPUTS,TRAY\_BOTTOM

GOTO SHIFTING\_IN\_PROGRESS

BCF OUTPUTS,SHIFTING\_MOTOR

CALL DISPLAY\_SHIFTING\_DONE

MOVLW IDLE

MOVWF CURRENT\_STATE

GOTO MAIN\_LOOP

;===========================================

TOGGLE

BTG OUTPUTS,LED\_WATERING

BTG OUTPUTS,LED\_SHIFTING

RETURN

;===========================================

LCD\_INIT

BCF OUTPUTS,LCD\_EN

CALL LDELAY

MOVLW 38H

CALL WRT\_INS

CALL LDELAY

MOVLW 0EH ;DISPLAY ON, CULCD\_RSOR ON

CALL WRT\_INS

CALL LDELAY

RETURN

;-------------------------------------------

CLEAR\_SCREEN

MOVLW 01H ;CLEALCD\_RS THE SCRELCD\_EN

CALL WRT\_INS

CALL DELAY

RETURN

;-------------------------------------------

SHIFT\_CURSOR\_RIGHT

MOVLW 06H ;SHIFTS CULCD\_RSOR RIGHT

CALL WRT\_INS

CALL DELAY

RETURN

;-------------------------------------------

LINE\_1\_START

MOVLW 80H ;CURSOR AT LINE 1 POS 0

CALL WRT\_INS

CALL DELAY

RETURN

;-------------------------------------------

LINE\_2\_START

MOVLW 0xC0 ; CURSOR AT LINE 2 POSITION 0

CALL WRT\_INS

CALL DELAY

RETURN

;-------------------------------------------

WRT\_INS

MOVWF LCD\_DATA

BCF OUTPUTS,LCD\_RS

BCF OUTPUTS,LCD\_RW

BSF OUTPUTS,LCD\_EN

CALL SDELAY

BCF OUTPUTS,LCD\_EN

RETURN

;-------------------------------------------

WRT\_DATA

MOVWF LCD\_DATA

BSF OUTPUTS,LCD\_RS

BCF OUTPUTS,LCD\_RW

BSF OUTPUTS,LCD\_EN

CALL SDELAY

BCF OUTPUTS,LCD\_EN

RETURN

;===========================================

;-------------STARTING MESSAGE--------------

DISPLAY\_STARTING\_MESSAGE

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

MOVLW A'Y'

CALL WRT\_DATA

CALL DELAY

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

MOVLW A'M'

CALL WRT\_DATA

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL DELAY

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

MOVLW A'D'

CALL WRT\_DATA

CALL DELAY

RETURN

;---------------IDLE STATE MESSAGE----------

DISPLAY\_IDLE\_MESSAGE

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'I'

CALL WRT\_DATA

CALL DELAY

MOVLW A'D'

CALL WRT\_DATA

CALL DELAY

MOVLW A'L'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL DELAY

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

RETURN

;---------------MAIN TANK EMPTY-------------

DISPLAY\_TANK\_EMPTY

BCF OUTPUTS,LED\_WATERING

BSF OUTPUTS,LED\_SHIFTING

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'M'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'K'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'M'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'P'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'Y'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_2\_START

MOVLW A'C'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'P'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'C'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'D'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

GOTO MAIN\_LOOP

;---------------TOP IS MISSING--------------

DISPLAY\_TOP\_MISSING

BCF OUTPUTS,LED\_WATERING

BSF OUTPUTS,LED\_SHIFTING

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'P'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'Y'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

RETURN

;--------------DISPLAYING WATERING----------

DISPLAY\_WATERING

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'W'

CALL WRT\_DATA

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL DELAY

MOVLW A'G'

CALL WRT\_DATA

CALL DELAY

RETURN

;---------DISPLAYING WATERING DONE----------

DISPLAY\_WATERING\_DONE

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'W'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'G'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'D'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

RETURN

;--------DISPLAYING TRAY REMOVAL----------

DISPLAY\_TRAY\_REMOVAL

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'R'

CALL WRT\_DATA

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL DELAY

MOVLW A'M'

CALL WRT\_DATA

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL DELAY

MOVLW A'V'

CALL WRT\_DATA

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL DELAY

MOVLW A'G'

CALL WRT\_DATA

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL DELAY

MOVLW A'B'

CALL WRT\_DATA

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL DELAY

MOVLW A'M'

CALL WRT\_DATA

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL DELAY

MOVLW A'Y'

CALL WRT\_DATA

CALL DELAY

RETURN

;--------DISPLAYING TRAY SHIFTING-----------

DISPLAY\_TRAY\_SHIFTING

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

MOVLW A'H'

CALL WRT\_DATA

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL DELAY

MOVLW A'F'

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL DELAY

MOVLW A'G'

CALL WRT\_DATA

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL DELAY

MOVLW A'R'

CALL WRT\_DATA

CALL DELAY

MOVLW A'A'

CALL WRT\_DATA

CALL DELAY

MOVLW A'Y'

CALL WRT\_DATA

CALL DELAY

MOVLW A'S'

CALL WRT\_DATA

CALL DELAY

RETURN

;---------DISPLAYING SHIFTING DONE----------

DISPLAY\_SHIFTING\_DONE

CALL CLEAR\_SCREEN

CALL SHIFT\_CURSOR\_RIGHT

CALL LINE\_1\_START

MOVLW A'S'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'H'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'F'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'T'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'I'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'G'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A' '

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'D'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'O'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'N'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

MOVLW A'E'

CALL WRT\_DATA

CALL TOGGLE

CALL DELAY

RETURN

;===========================================

SDELAY

NOP

RETURN

;-------------------------------------------

LDELAY

;LETS HAVE 8ms DELAY USING TIMER 1

;T1C0N = 10000000

MOVLW 80H

MOVWF T1CON

; preload = 65536 - 8000 = 57536 = E0 C0h

MOVLW 0xE0

MOVWF TMR1H

MOVLW 0XC0

MOVWF TMR1L

BCF PIR1,TMR1IF

BSF T1CON,TMR1ON

CHK BTFSS PIR1,TMR1IF

GOTO CHK

BCF T1CON,TMR1ON

RETURN

;-------------------------------------------

DELAY

;LETS HAVE 250ms BETWEEN CHARACTERS

;USING PS 4

MOVLW B'10100000'

MOVWF T1CON

;preload = 65536 - 62500 = 3036 = 0B DC H

MOVLW 0x0B

MOVWF TMR1H

MOVLW 0xDC

MOVWF TMR1L

BCF PIR1,TMR1IF

BSF T1CON,TMR1ON

WAIT\_DELAY BTFSS PIR1,TMR1IF

GOTO WAIT\_DELAY

BCF T1CON,TMR1ON

RETURN

;----------WAITING FOR SOME TIME------------

WAIT

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

RETURN

;===========================================

HIGH\_ISR

MOVWF WREG\_TEMP\_SAVE

MOVFF STATUS,STATUS\_TEMP\_SAVE

MOVFF BSR,BSR\_TEMP\_SAVE

BTFSS INTCON,TMR0IF

GOTO ISR\_EXIT\_RESTORE

;JUST FOR TESTING IF TIMER WORKS OR NOT

;BTG OUTPUTS,WATER\_PUMP

BCF INTCON,TMR0IF

INCF WATERING\_COUNTER,F

INCF DAY\_COUNTER\_L

BNZ ISR\_EXIT\_RESTORE

INCF DAY\_COUNTER\_H

ISR\_EXIT\_RESTORE

MOVFF BSR\_TEMP\_SAVE,BSR

MOVFF STATUS\_TEMP\_SAVE,STATUS

MOVF WREG\_TEMP\_SAVE,W

RETFIE

;===========================================

END

Teachers should assess CLO2, CLO3 and CLO4 based on the given rubrics

(overall weightage 20%)

Recommended Percentage Breakdown

|  |  |
| --- | --- |
| **CLO** | **Percentage** |
| CLO2 (Investigation) | 10% |
| CLO3 (Referencing/Citations)  *(Turnitin report should be generated.)* | 5% |
| CLO4 (Communication) | 5% |