Microprocessors and Interfacing

FLOUR PACKING MACHINE

Submitted by: Group 80

Group Members:

Archit Rungta (2019A3PS0450G)

Rohan Choudhary (2019AAPS0402G)

Shubham Kalantry (2019A7PS0141G)

Yash Bagrecha (2019A3PS0387G)

Smit Sawant (2019A3PS0394G)

Shreyans Singh (2019A7PS0078G)

DATE:18/4/21
(f20190141@goa.bits-pilani.ac.in)

Contents:

User Requirement & Technical Specifications:

Design a Micro-Processor based flour packing machine.

The Technical Specifications are as follows:

1. Flour is stored in the silo.
2. The user keys in the required amount of flour per packet.
3. The user interface has a keypad with keys 0-9, Weight, Temp, Enter, Back Space and Start Key.
4. The user will press the Weight key and then provide the required weight via the keypad.
5. The system is also required to monitor the temperature of the flour where packing is going on. This temperature range (in Celsius) is user settable.
6. To set the temperature the user will press the Temp key followed by the temperature (2 digits).
7. While the user is pressing a key, it should be displayed on 7-segment displays available with the keyboard.
8. The measured temperature value of the flour should be displayed on a seven-segment display.
9. On pressing of START key the system should pack the specified amount of flour based upon the user input.
10. The system is also required to display the number of packets packed in every hour on a seven-segment display (The number of packets per hour will not exceed 99).
11. An alarm must be provided if the temperature rises 5 degrees above the set temperature.
12. If the user presses the START key without configuration, then the system should pack for the preset/default weight and temperature.

Assumptions & Justifications:

Justification:

1. As the no. of packages per hour does not exceed 99 and the temperature value can be a two-digit number only, so two seven segment displays are needed to display each data.
2. As only the packing temperature needs to be measured, only a temperature sensor placed near the dispenser is required.

Assumptions:

1. User inputs the temperature in °C.
2. Maximum temperature entered is 99 °C.
3. Weigh per packet is entered in kg(s) and is less than 99kgs.
4. All user inputs should be whole numbers.
5. The preset value of weight is 5 kg, and temperature is 25 °C.
6. Rate of flow of flour is constant with time (0.25 kg/s), when the valve is kept open.
7. For a single digit weight value, it should be entered along with a zero in front e.g., 5 should be entered as 05. Similarly for the temperature input as well.
8. Temperature of the flour will be checked before each packet of flour is filled.
9. STOP key needs to be pressed when the alarm is raised.

LIST OF COMPONENTS:

COMPONENTS USED	QUANTITY	PURPOSE
8086	1	Central Processor
8255	3	PPI for I/O
8254	1	Programmable interval timer
8259	1	Programmable interrupt controller
8284	1	Clock generator
6116 RAM	2	RAM for the Memory
2716 ROM	4	EPROM
74LS138	1	Address Decoder
74LS373	3	Latching the Bus
74LS244	1	Uni-Directional Buffer
74LS245	2	Bi-Directional Buffer
L293D	1	Motor Driver
ADC0808	1	ADC 8 channel 8 bit
7447	6	BCD to Seven segment Display
LM35	1	Temperature sensor
OR GATE	9	
NOT GATE	2	
4x4 KEYPAD	1	16 Key Matrix
7-Segment common anode Display	6	O/p Display
LED	3	Output Status/Alarm
BUZZER	1	Alarm
RESISTORS		
12V Bipolar stepper motor	1	Controlling dispenser valve

ADDRESS MAP:

ITEM	ADDRESS	REMARK
ROM1	00000H-00FFFH	2x 2KB ROM
ROM2	FF000H-FFFFFH	2x 2KB ROM
RAM	01000H-01FFFH	2x 2KB RAM
8255(#1)		
PORT A	00H	Connects two 7-segment displays through 7447 decoders displaying digit inputs from keypad
PORT B	02H	Keypad status LEDS
PORT C	04H	4x4 Keypad
CONTROL WORD REGISTER	06H	
8255(#2)		
PORT A	08H	Connects two 7-segment displays through 7447 decoders displaying current temperature
PORT B	0AH	Connects two 7-segment displays through 7447 decoders displaying count of floor packets
PORT C	0CH	Connects buzzer and bulb to raise alarm
CONTROL WORD REGISTER	0EH	
8255(#3)		
PORT A	10H	Connects stepper motor of valve through motor driver
PORT B	12H	Data lines of ADC
PORT C	14H	Generates SOC and EOC of ADC
CONTROL WORD REGISTER	16H	
8254		To generate ADC clock and 1 hour timer
COUNTER 0	18H	Input CLK 5MHz and count 05d (Mode 3)
COUNTER 1	1AH	Input CLK 1MHZ and count 60000d (Mode 2)
COUNTER 2	1CH	Input CLK 100/6Hz and count 60000d (Mode 2)
CONTROL WORD REGISTER	1EH	
8259	20H,22H	To generate interrupt every 1 hr

MEMORY AND I/O INTERFACING:

	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BHE’	M/IO’
RAM1e	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
RAM1o	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
ROM1e	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1
ROM1o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1
ROM2e	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
ROM2o	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
8255(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
8255(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
8255(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
8254	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0
8259	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

SYSTEM DESIGN:

When the system is turned on, the user is expected to set the weight and temperature values to be used while packaging using the WEIGHT and TEMP buttons on the 4x4 keypad. When the temperature is being set, an orange led glows, indicating that the temperature setting is in progress, similarly for the weight setting. A valid input followed by an ENTER is expected from the user once he has pressed the TEMP or WEIGHT button. If the start key is pressed without setting the weight or temperature, the default values are loaded, and the system starts packaging.

The system checks if the temperature is within 5 °C of the required temperature and then starts packaging. The following temperature check is done before the next packet is packed. If the temperature exceeds the limit, an alarm is raised, the user is expected to press the STOP button to stop the alarm. To resume the packing process, the user should press START again. The number of packets packed is displayed after every hour. The dispenser valve is open only for a calculated amount of time to dispense the required weight into the packet. Once the required amount has been dispensed, the user needs to press start again to pack the next bag.

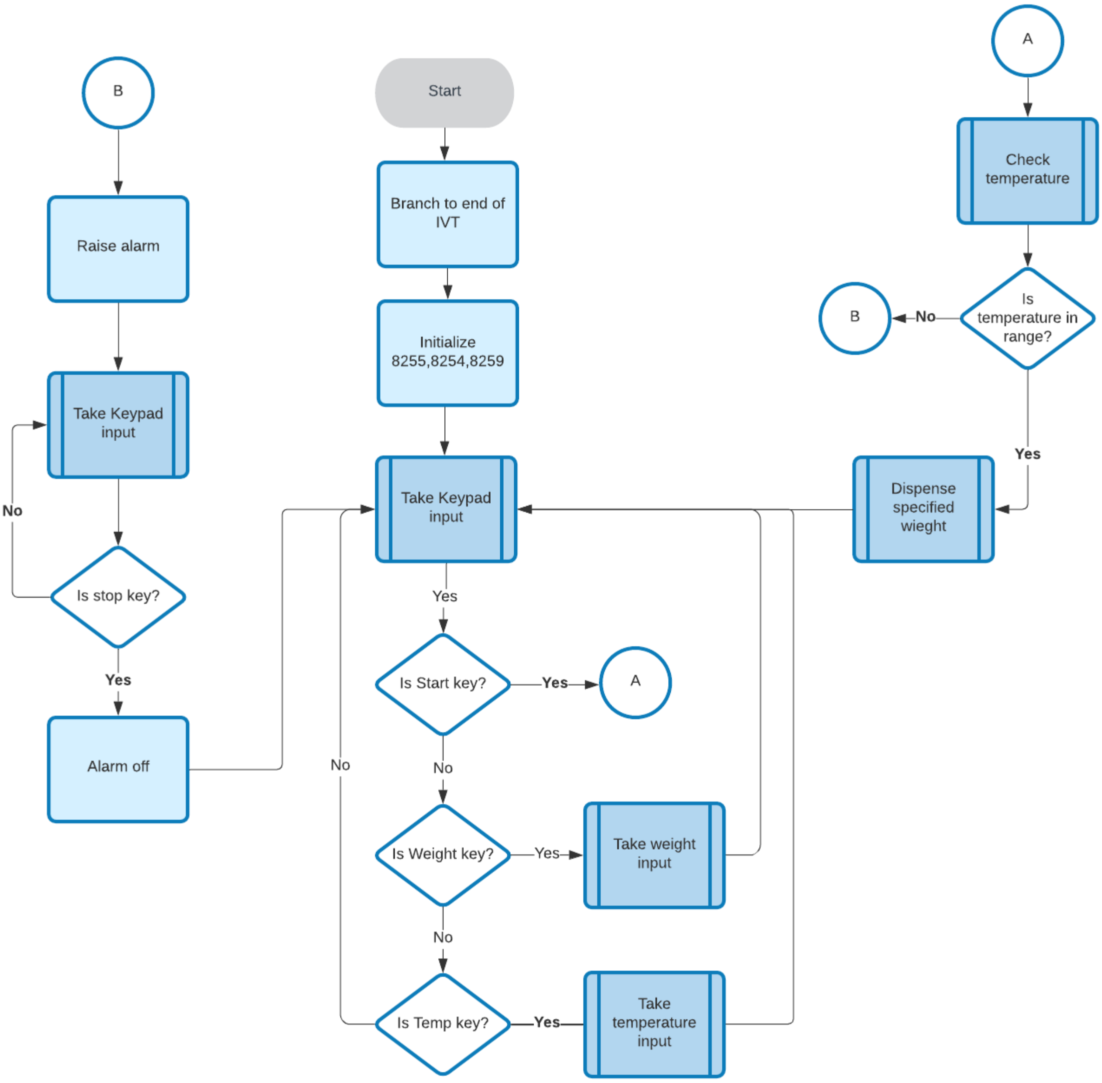
While setting the temperature or weight, the user is expected to enter the tens place value and then the units place value, respectively.

DESIGN:

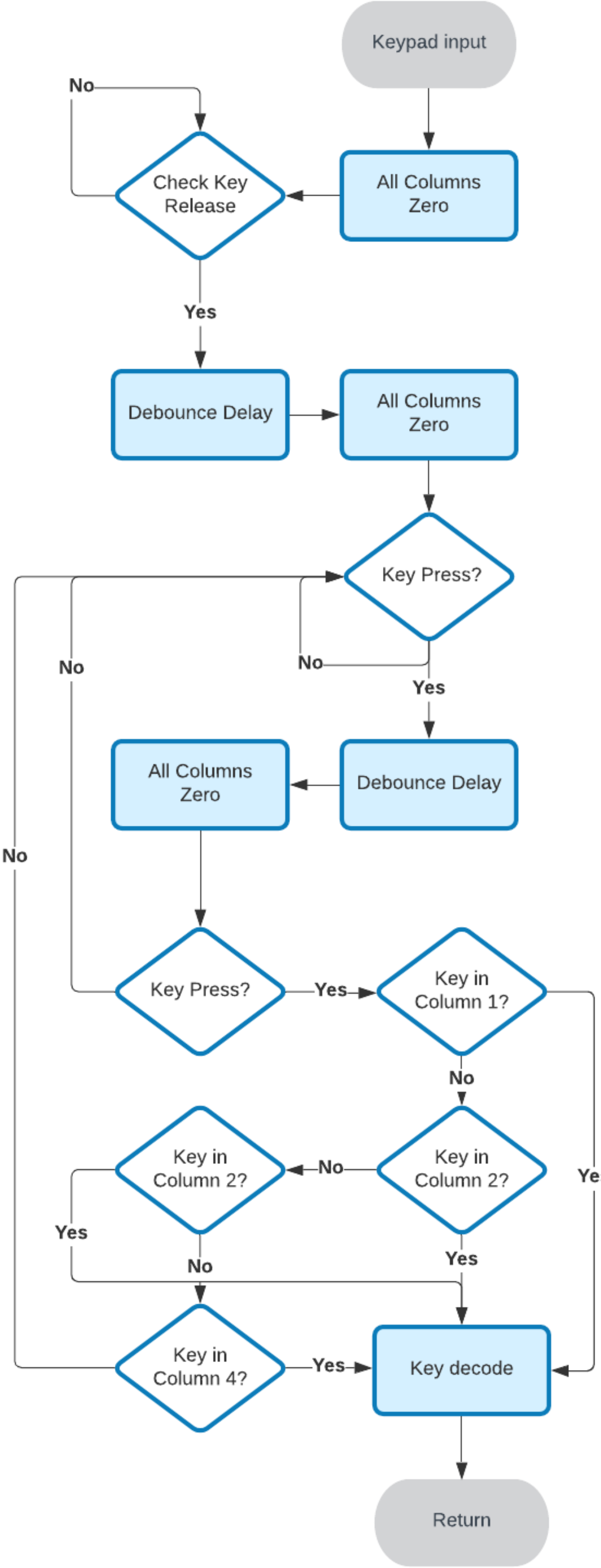
Complete design shown with proper labelling (attached as design.pdf)

FLOWCHART:

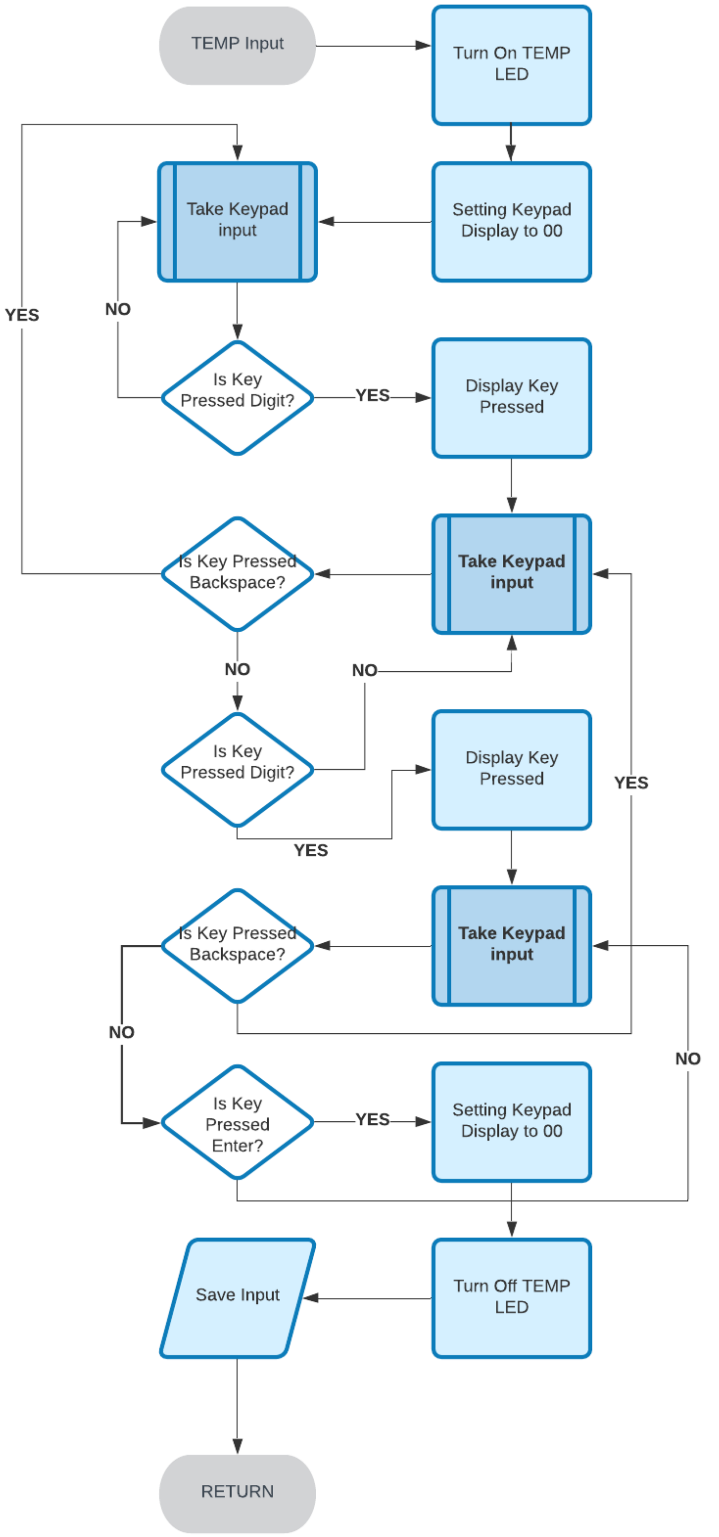
Main Program:



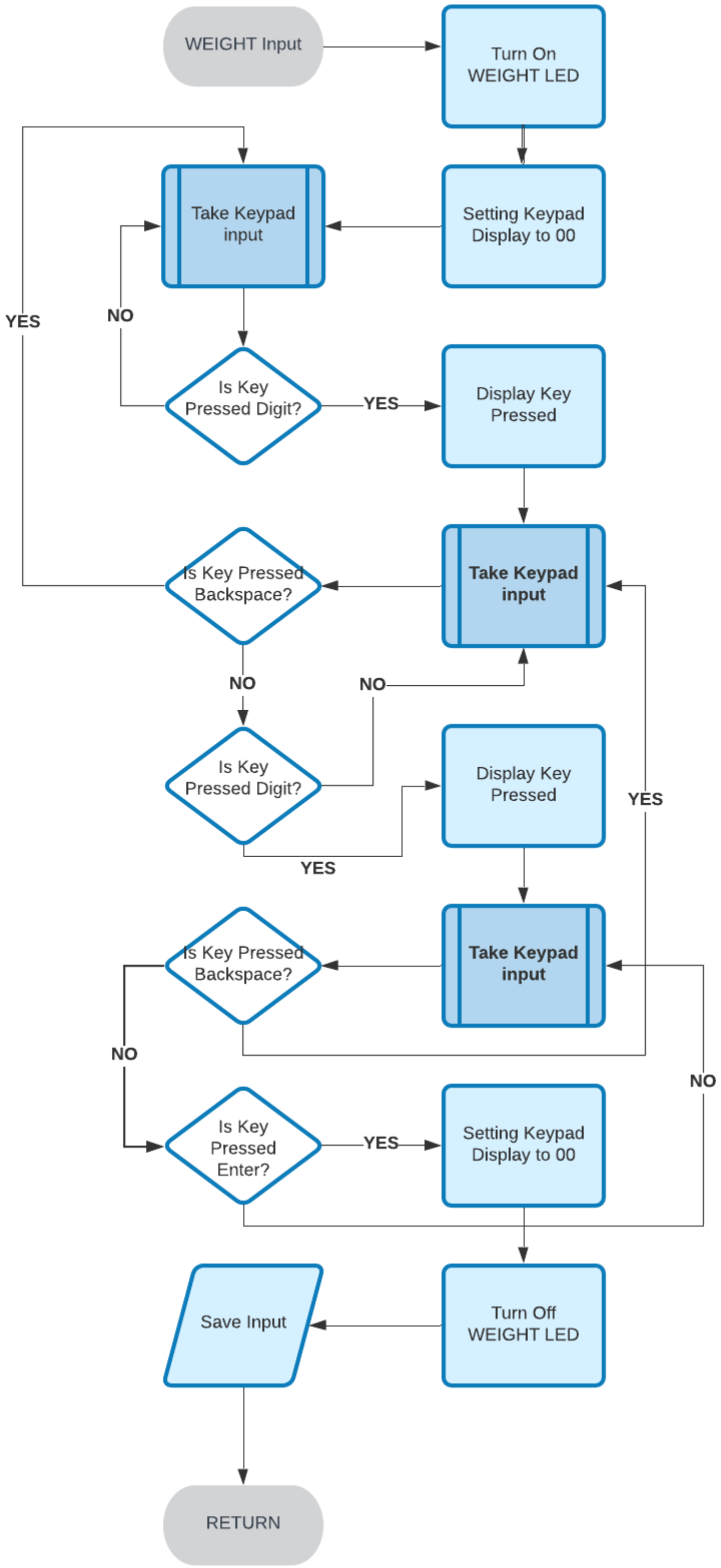
Flowchart of Keyboard Input



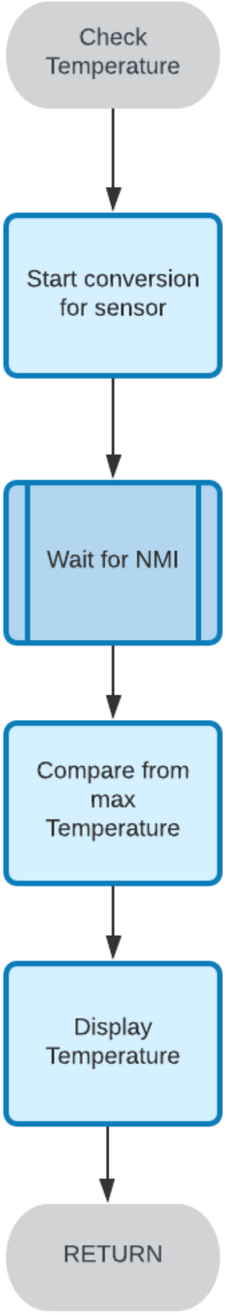
Flowchart of Temperature Input



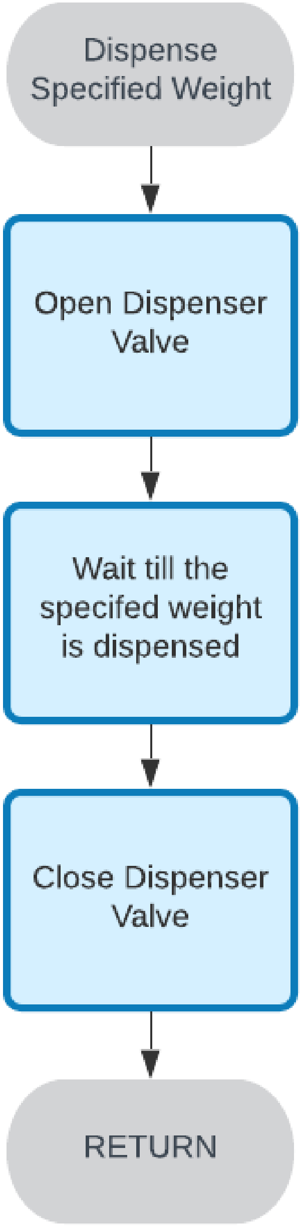
Flowchart of Weight Input



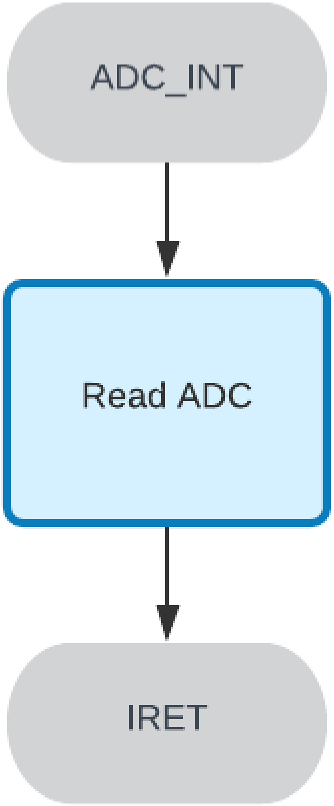
Flowchart of Temperature Check



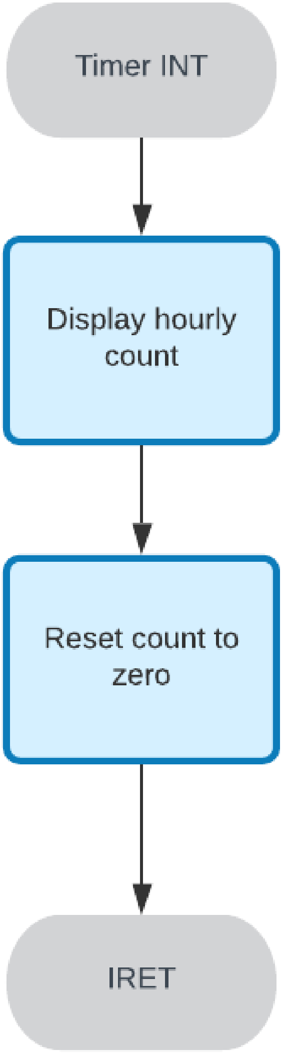
Flowchart of Dispense Specified Weight


Dispense Specified Weight
Open Dispenser Valve
Wait till the specifed weight is dispensed
Close Dispenser Valve
RETURN

Flowchart of ADC ISR



Flowchart of Timer ISR



Variations in Proteus Implementation with Justification:

1. 2732 is used as 2716 – not available in Proteus.
2. Using 8253 – as 8254 not available in Proteus.
3. Setting clock for counter 0 at 2 MHz for 8253, instead of 5MHz in 8254, which gives output signal of 1 MHz by passing count value of 2d in Mode 3.
4. 8284 System Clock generator is not used in Proteus as clock frequency can be set using the microprocessor properties window.
5. 8259 doesn’t work in Proteus 7. So, Proteus 8 is being used for simulation.

Firmware:

Implemented using emu8086 attached (attached as main.asm and FPM.bin)

List of Attachments:

1. Complete Hardware Real World Design – design.pdf
2. Manuals
    a. 8255
    b. 8254
    c. 8259
    d. 8086
    e. 8284
    f. 6116 RAM
    g. 2716 ROM
    h. 74LS138
    i. 74LS373
    j. 74LS244
    k. 74LS245
    l. L293D
    m. ADC0808
    n. 7447
    o. LM35
    p. 7 Segment Display
    q. Bipolar Stepper Motor
3. Proteus File – MUP.dsn
4. EMU8086 ASM File – main.asm
5. Binary File after assembly – FPM.bin

APPENDIX:

8086: https://drive.google.com/file/d/1ITI7JJpn9uz9t0HeuOvJOQbLvK_ht-vN/view?usp=sharing

8255:https://drive.google.com/file/d/1IFFKG92R485kafEq0_vrQNwZCHbPXdQa/view?usp=sharing

8254: https://drive.google.com/file/d/1IuAJKgKdh5irJG4CGukq8tj0jzoHpe7_/view?usp=sharing

8259:https://drive.google.com/file/d/1ZdeLjPOd8yADPtb9ut8iJpW2_CwdSQkG/view?usp=sharing

8284: https://drive.google.com/file/d/1sBiA8yCE_p0lq2TChc_EIiRTSAIMAyYz/view?usp=sharing

6116 RAM: https://drive.google.com/file/d/1PMA0PDqv5oJOnCyLRRlXjBKxnTcRGUJ-/view?usp=sharing

2716 ROM:

https://drive.google.com/file/d/1cp2GdRbxT4R69ZO9x4fvm5ctZe5uOsN1/view?usp=sharing

74LS138:https://drive.google.com/file/d/1_yvKu2QK_3RbkAr3HmInbMLq5yptb321/view?usp=sharing

74LS373: https://drive.google.com/file/d/1cp2GdRbxT4R69ZO9x4fvm5ctZe5uOsN1/view?usp=sharing

74LS244: https://drive.google.com/file/d/1ev9vn8YvgcFiSfvd_YiwubKsWoUEnEbz/view?usp=sharing

74LS245: https://drive.google.com/file/d/1cp2GdRbxT4R69ZO9x4fvm5ctZe5uOsN1/view?usp=sharing

L293D: https://drive.google.com/file/d/1UAPNHXmZZoWqwILBm1nqA9_WbyNZOhuk/view?usp=sharing

ADC0808: https://drive.google.com/file/d/16kMSkXoX4QIhlizxU9S_M8L_scquBqYE/view?usp=sharing

7447: https://drive.google.com/file/d/15FGl0Hp9rYo26nKYj_bObE6_TYvj9yH/view?usp=sharing

LM35: https://drive.google.com/file/d/1Y09O2tuBP2Je4YRqhc73yVEHOVxiZ9ST/view?usp=sharing

7 Segment Display:

https://drive.google.com/file/d/1VEfR7bqaXwPtB33xPVGCqfEiG5pnWp0/view?usp=sharing

Bipolar Stepper Motor: https://drive.google.com/file/d/1Whg5PwMMYOm8Tbod7uXPEHzK62VhlpU/view?usp=sharing