Smart Garage System

Design Project

Problem Number 19

Group 49

In partial fulfilment of the requirements of CS/EEE/INSTR F241,
Microprocessors, Programming and Interfacing
19 April 2021

Group members:

Chinmay Bande	2019AAPS0202G
Jathin Narayan	2019A7PS1001G
Kunal Avalakki	2019A7PS1013G
Mihir Thalanki	2019A7PS1014G
Nimish Wadekar	2019A7PS1004G
Omkar Mahesh Garad	2019A7PS1010G

Contents

User Requirements and Technical Specifications

Design a smart garage system that allows remote handling of the garage door, and automatically senses and displays the number of cars inside the garage.

The technical specifications are as follows

- Door Size is 2.5m * 2.2m
- Distance between IRED and its corresponding sensor should not be more than 2m.
- A thin metal sheet of size 2m * 6m needs to be placed above the strain transducer so that the weight of the car is equally distributed on it.
- The 2 lateral IR LED and sensor are placed at sufficient distance (2m - 3m) from each other such that the light that gets reflected when a car approaches doesn't reflect onto the lateral sensor, and at some point, both sensors must have a line break simultaneously when a car passes through.
- Vehicle weight is between 200kg to 5000kg.

Assumptions and Justifications

Justifications

- 2 sensors are used to determine whether the car is entering or exiting the garage.
- Strain transducer is used to measure the weight of the car to determine whether it's a car or a valet walking.

Assumptions

- When the system boots up, the garage is empty.
- A maximum of 1 car/valet can enter/exit at a given point in time.
- Until a vehicle/valet has fully finished entering or exiting the garage, with the garage door in closed state, another vehicle/valet does not appear.
- The vehicle must halt for a few seconds while triggering both sensors for the strain transducer to measure its weight.

Components Used

- 8086
- 8284 with crystal
- 8255 - 2 nos.
- 8259
- 8253
- 2716 - 4nos. 2 ROM chips required for storing of IVT and the code, 2 chips for address $FFFFE_H$
- 6116 - 2nos. 2 RAM chips which are used for variable storage and stack.
- LS138 - 2nos.
- LS373 - 3nos.
- LS244
- LS245 - 2nos.
- A4988 - Microstepping Driver with Overstepping Current Protection to amplify current to drive the stepper motor in the garage door
- ADC 0803 - 8 bit A/D converter with analog input voltage ranging from 0 to V_{CC}(5V)
- QEE113 - Plastic Infrared Light Emitting Diode for detection of car entering/exiting
- TSSP94038 - IR Sensor module for detecting IR beams from IRED
- ST-TX01-ASK - ASK Transmitter module for transmitting radio frequencies in remote control
- ST-RX02-ASK - ASK Receiver module for receiving radio frequencies in remote control
- SLB700A/06VA1 - Strain transducer - Used as a weight sensor
- HT12D - Decodes the data received by the RF Receiver
- HT12E - Encodes the data from the button and sends it to the RF Transmitter
- JHD162A - LCD panel for displaying the number of cars in the garage
- 34H2A2850 - Stepper motor for garage door
- Buttons - For the valet's remote control
- Required gates

Address Map

Memory map

- ROM 1 - 00000_H to $00FFF_H$
- RAM 1 - 01000_H to $01FFF_H$
- ROM 2 - $FF000_H$ to $FFFFF_H$

I/O Map

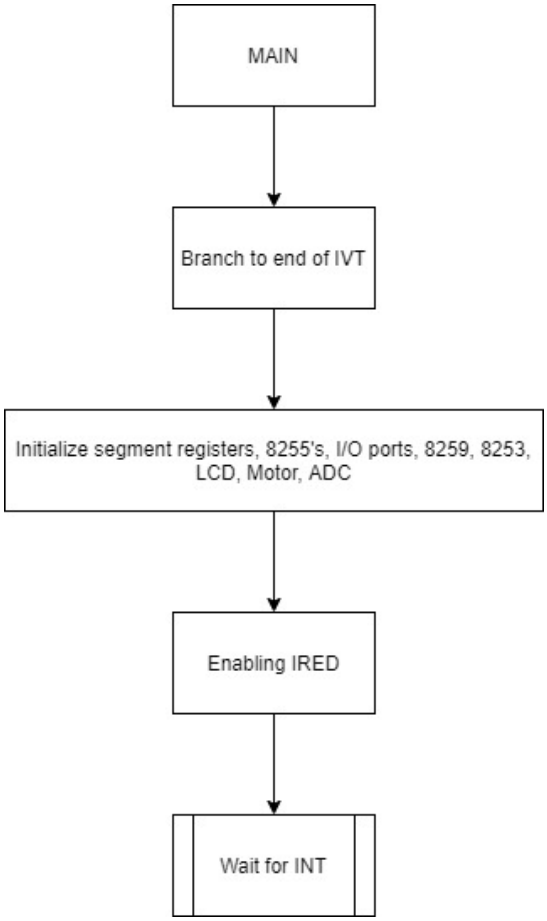
- 8255 - 00_H to 06_H
- 8255 - 08_H to $0E_H$
- 8259 - 10_H to 12_H
- 8253 - 18_H to $1E_H$

Design

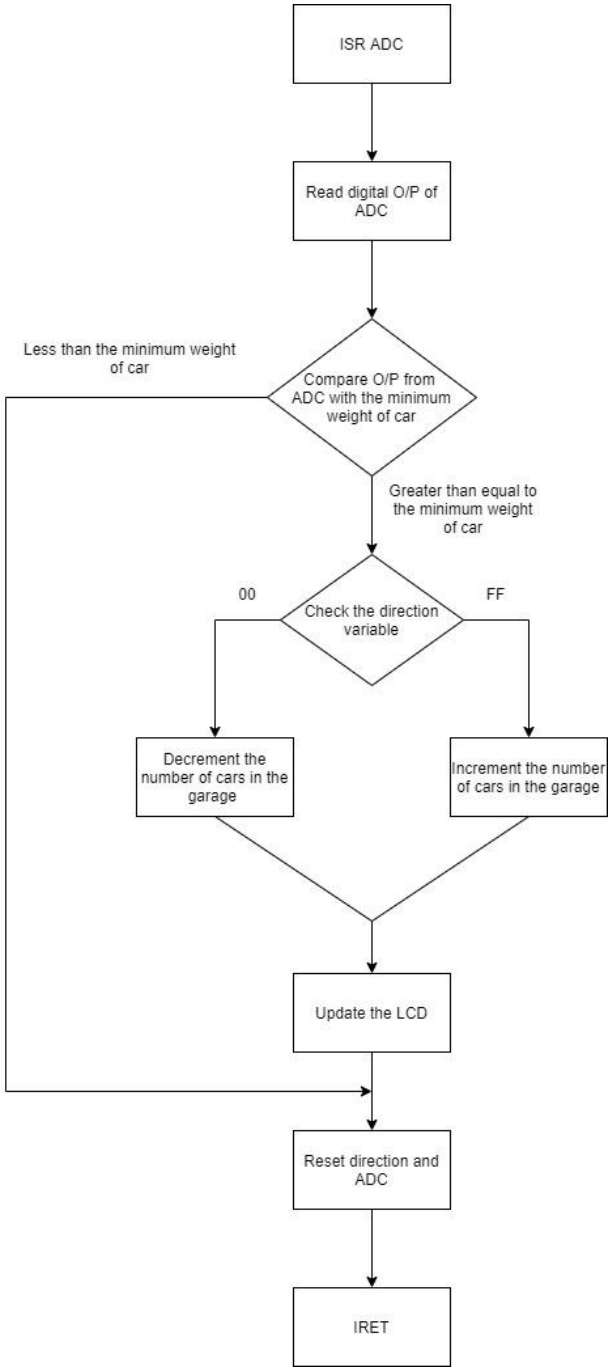
Complete design attached in separate pdf file.

Flow Chart

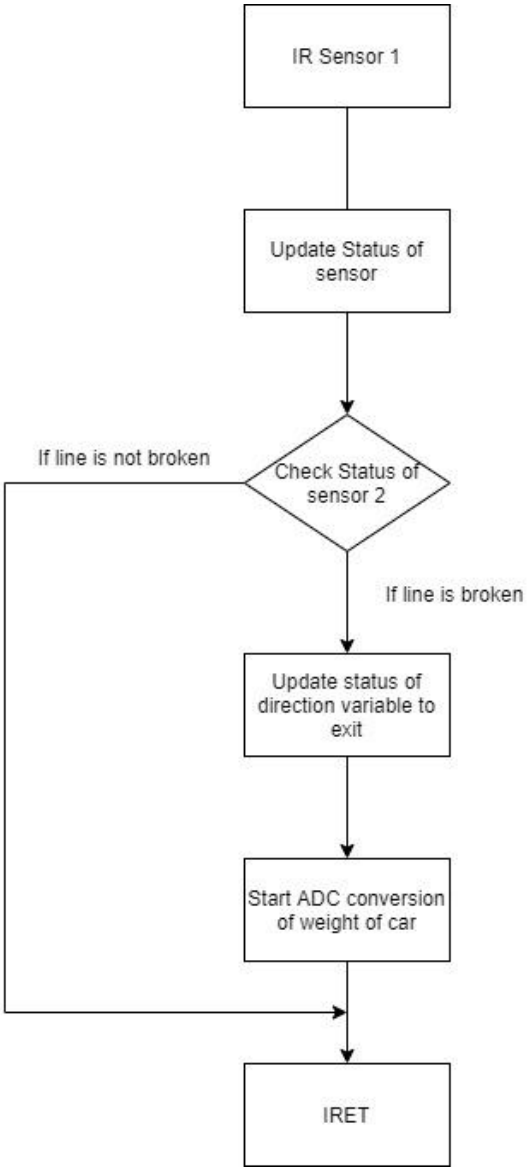
Main Program

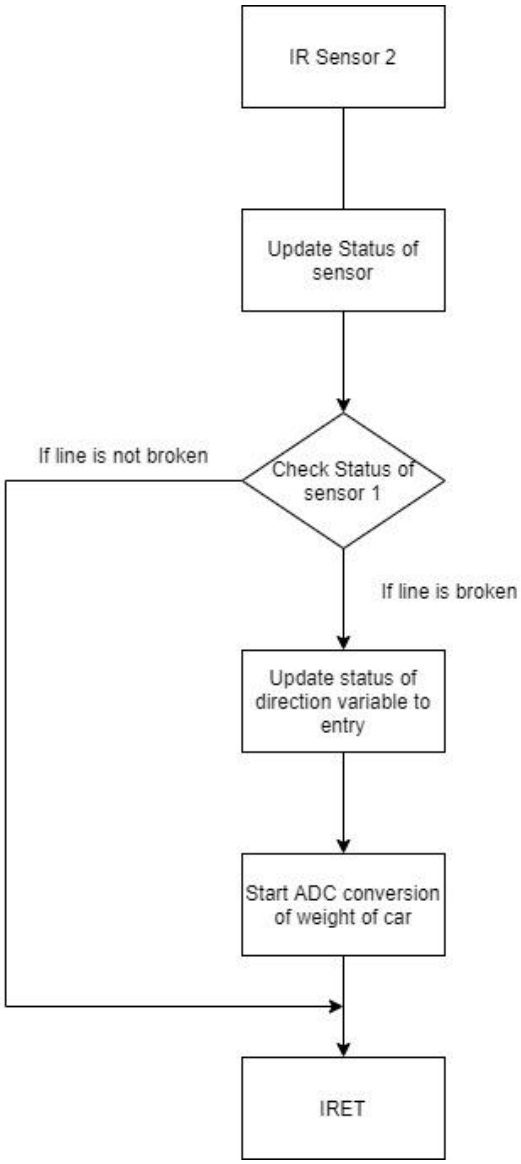


ADC ISR

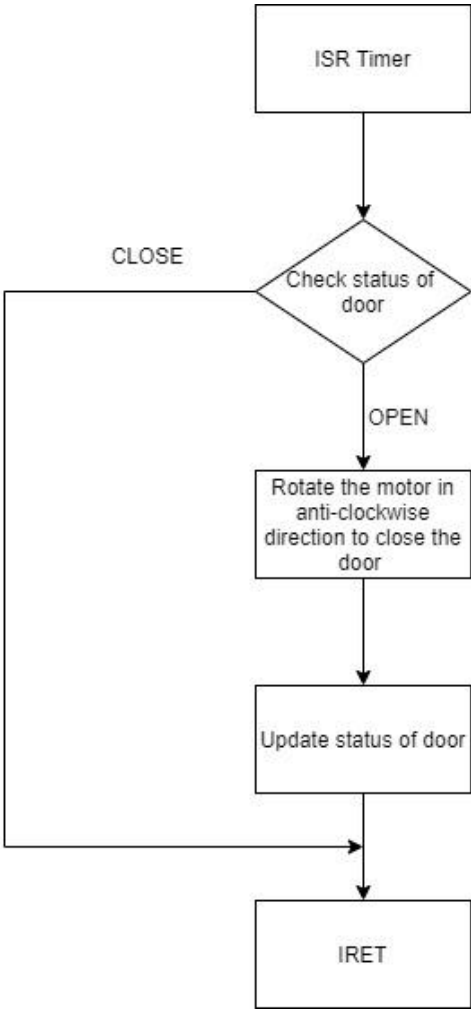

ISR ADC
Read digital O/P of ADC
Less than the minimum weight of car
Compare O/P from ADC with the minimum weight of car
Greater than equal to the minimum weight of car
00
Check the direction variable
FF
Decrement the number of cars in the garage
Increment the number of cars in the garage
Update the LCD
Reset direction and ADC
IRET

IR Sensor 1 and 2 ISRs

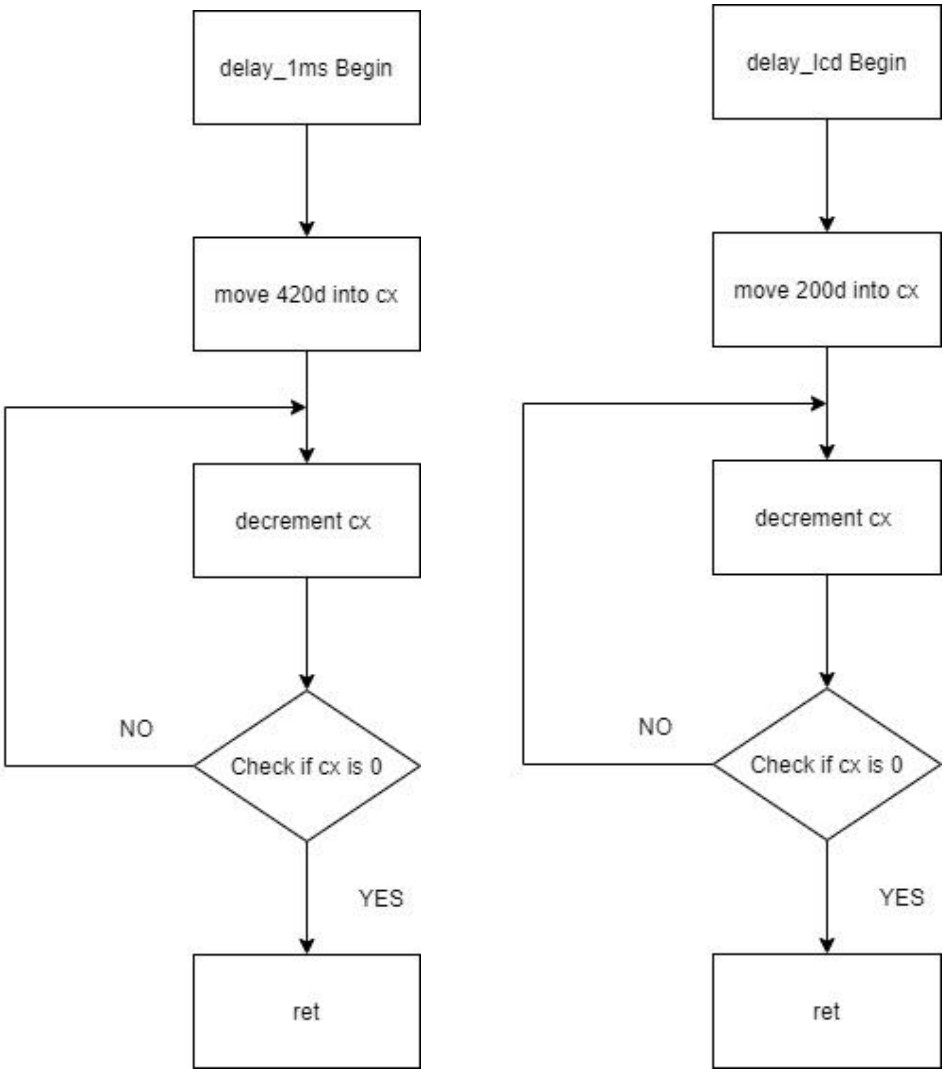


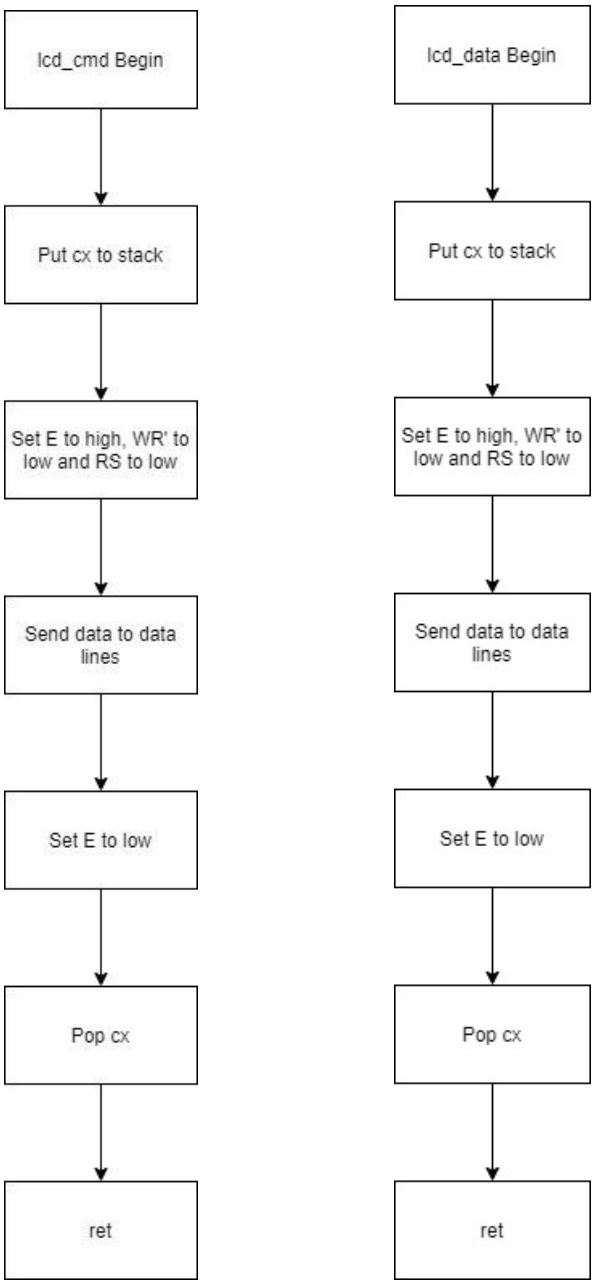


Timer ISR


ISR Timer
CLOSE
Check status of door
OPEN
Rotate the motor in anti-clockwise direction to close the door
Update status of door
IRET

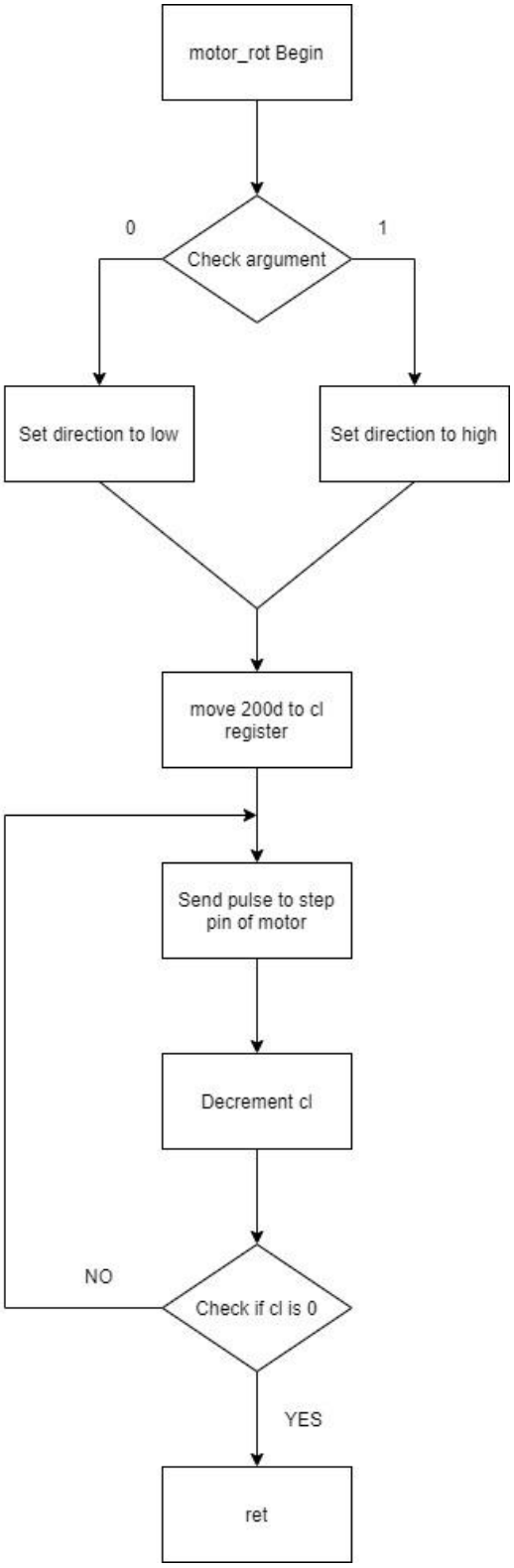
Delay 1ms and Delay for LCD procedures



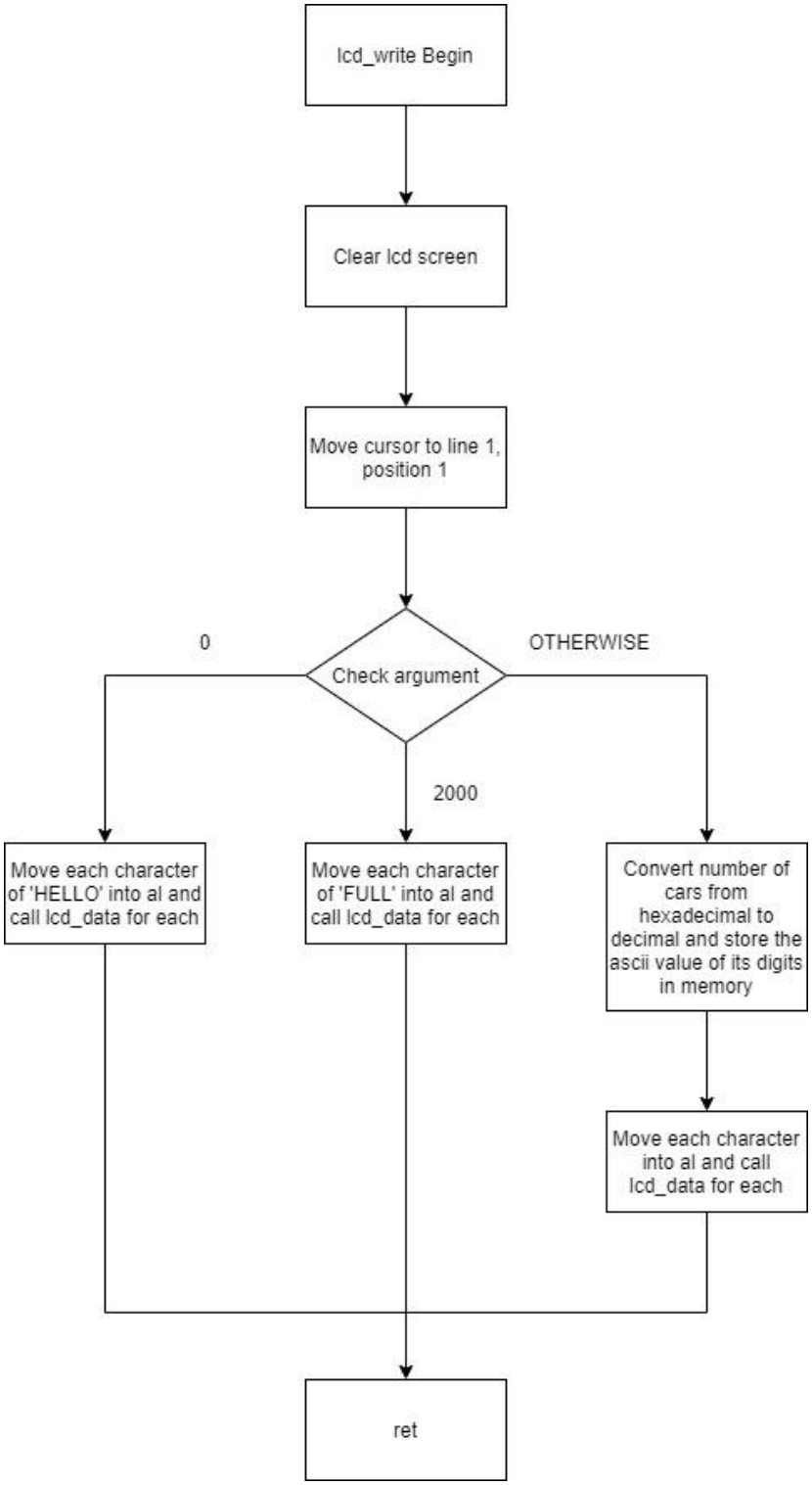


Procedures to send command and data respectively to LCD

Procedure to rotate motor



Procedure to write a number (or string if empty or full) to LCD



Variations in Proteus Implementation

- 8284: No 8284 in proteus, used internal clock available in Proteus.
- ROM: 2732 instead of 2716.
- Memory decoding: Lines A13, A14, A15 instead of A12, A13, A14.
- ROM2 not used because proteus allows changing the starting location.
- Motor driver: L297 instead of A4998.
    - L297 has no sleep mode, always active.
- Strain Transducer: Replaced with a simple DC source.
- IR Sensors replaced with SPDT switches.
- Remote button (along with RF transmitter, encoder, decoder and receiver) replaced with a single SPDT switch.
- Replaced 8259 with a logic circuit making use of NMI to handle interrupts due to faulty 8259 chip in Proteus.
- Reordered 8255 ports due to a bug in Proteus that prevented I/O mode from working after using BSR mode on that chip.
- Replaced 8253 with a delay subroutine to accommodate lesser interrupts into the Proteus design.

Firmware

Implemented using Emu8086, attached.

List of Attachments

1. Complete hardware design - Design.pdf
2. Manuals:
    a. A4998 Stepper Motor Driver Module
    b. ADC0803
    c. HT12D RF Decoder
    d. HT12E RF Encoder
    e. JHD162A LCD
    f. NEMA34 Motor
    g. QEE113 IRED
    h. SLB700A Strain Transducer
    i. ST-RX02-ASK RF Receiver Module
    j. ST-TX01-ASK RF Transmitter Module
    k. TSSP940 IR Sensor
3. Proteus design file - Design.dsn
4. Emu8086 ASM file (for hardware design) - Code.asm
5. Emu8086 ASM file (for Proteus Simulation) - Code_sim.asm
6. Binary file after assembly - Code.bin
7. Binary simulation file after assembly - Code_sim.bin