PROBLEM STATEMENT

Problem:

A Mine Exploration Vehicle (MEV) is to be designed. The vehicle is six wheeled with an on board robotic arm to pick up samples and on board scientific instruments and cameras (vehicle fig is shown below).

The primary aim of the vehicle is to explore the terrain, measure atmospheric conditions in the mine and assess samples of rocks and soil. This can be controlled remotely from above the ground. All subsystems on the vehicle are controlled by the remote unit and are in contact with the remote unit through a RF modem.

Vehicular motion and Sample collection Subsystem

Each of the six wheels is to be powered by an individual motor so that the vehicle is capable of climbing over obstacles not exceeding a certain height. The vehicle can execute a 360°turn in place. The front and rear wheels are used for steering. An upright position is to be maintained at all times and hence the suspension of the vehicle cannot have a tilt > 45 °.The tilt is usually not allowed to exceed 30°during normal operation.The vehicle should be capable of digging up to small depths. Digging is to be accomplished by spinning one of the front wheels in place to grind into the soil. The same mechanism that is used for tilting the vehicle upwards is also used for tilting the vehicle downwards. The vehicle is designed to remain motionless while the digging wheel is spinning.DC motors are used for rotating the wheels and stepper motors are used for steering. The two front wheels are locked together by the same steering mechanism; the same is done in case of the rear wheels.A Robotic arm is used for sample collection. The robotic arm is capable of movement both in the horizontal plane and in the vertical plane.Communication of vehicle with surface takes place through RF modem (CC2420).

Design the necessary hardware and software for implementing the above-mentioned task.

ASSUMPTIONS

The following assumptions have been made in designing :

1.The clock frequency is assumed to be given as 2MHz.

2.Due to the non availability of RF modem(CC2420) in proteus,wired transmission via the keypad on the vehicle has been used to drive the Mine Exploration Vehicle(MEV).

3. Gyroscope is used to measure the tilt angle which needs to be less than 45° according to the problem.Since the gyroscope is not available in the proteus library, it is simulated via a potentiometer.The analog output of gyroscope is simulated using potentiometer.

DESIGN SPECIFICATIONS

Mine Exploration Vehicle is a six wheeled rover with a robotic arm designed to pick up samples and examine using on board scientific instruments and cameras.
The following are the design solution to the given problem:

- Each of the 6 wheels are powered by DC motors.
- 2 Stepper motors are used for the steering system. Stepper motor can be at-tached to the shaft connecting the two front wheels with the help of the rack and pinion steering mechanism, thus requiring one stepper motor each for the front wheels and rear wheels. This also locks the front wheels and the rear wheels as mentioned in the problem statement.
- This steering system enables the vehicle to climb over obstacles upto a certain height. An tilt of the vehicle is not allowed to exceed 45 degrees. This is en-sured using a gyroscope. As soon as the gyroscope records a tilt of 45 degrees of greater, the motor functions are stopped.
    - The digging of the test surface can also be done by moving only the front wheel. The vehicle is motionless during the digging process.
    - The sample of the soil can be examined using the robotic arm. The robotic has 2 DOF in the vertical and the horizontal plane.
    - The communication with the vehicle takes place through a RF modem which is used by the person on the surface operating the vehicle.
    - 360 degrees spin in place is done by rotating the left wheels forward and right wheels back-ward. This is using the differential drive system.

LIST OF CHIPS USED AND ADDRESS MAP

1) INTEL 8086

2) PPI 8255 (1)

Port A	00H
Port B	02H
Port C	04H
Control Register	06H

3) PPI 8255 (2)

Port A	08H
Port B	0AH
Port C	0CH
Control Register	0EH

4) EPROM (2732 x 2):

4 KB (Even Bank) + 4 KB (Odd Bank) from
EPROM1:0000h-01FFFh
EPROM2:0000h-01FFFh

5) SRAM (6116 x 2):

2 KB (Even Bank) + 2KB (Odd Bank)
RAM:02000h-02FFEh
RAM:02001h-02FFFh

HARDWARE SPECIFICATIONS

Serial Number	Hardware	TYPE	Number Used
1.	Microprocessor	Intel 8086	1
2.	Octal Latch	74LS373	3
3.	Bi-Directional Buffer	74LS245	2
4.	Erasable Programmable Read Only Memory	2732(4k x8)	2
5.	Random Access Memory	6116(2K x8)	2
6.	Programmable Peripheral Interface	Intel 8255A	2
7.	Logic Gates	TTL ICs	Multiple
8.	Push/Pull Four Channel Motor Driver	L293D	4
9.	High voltage/High Current Darlington Transistor Array	ULN2003A	1
10.	DC Motor		8
11.	Unipolar Stepper Motor		2
12.	Pull Up Resistor		4
13.	Keypad Phone		1
14.	Analog to Digital Converter	ADC0804	1
15.	3*8 Decoder	74LS138	1

SYSTEM DESCRIPTION

- The microprocessor based system which is able to move,explore and pick up things on being ordered using an Intel 8086 microprocessor interfaced at 2MHz as a Central Processing Unit(CPU).
    - Two 4KB EPROM and two 2KB RAM are interfaced to the microprocessor.
- The system uses three 74LS373 latches for demultiplexing address-data lines, two 74LS245 bi directional buffers to buffer data lines and one 3*8 74LS138 Decoder for selecting either of the two PPI 8255A.

First 8255A PPI

- It is used in simple I/O mode(D7 bit of 8255A is equal to 1) for interfacing keypad.
    - Port A is used as an input port(MODE 00) but it is left unused.
- Port B is used as an input port(MODE 00) which takes in the input from the ADC0804
- Port C Lower is used as an input for the keys coming from keypad while Port C Upper is used as an output port.

Second 8255A PPI

- It is used in simple I/O mode(D7 bit of 8255A is equal to 1) for interfacing DC motors and Stepper Motors.
- In port A, PA0 – PA3 are used as output ports for interfacing the front wheels of vehicle while PA4 – PA7 are used as output ports for interfacing the middle two wheels of the vehicle.
- In port B, PB0 – PB3 are used as output ports for interfacing rear two wheels of the vehicle and PB4 – PB7 are used as output ports for interfacing the two stepper motors.
    - Port C lower is used as an output port for interfacing two DC motors used as Robotic Arm while Port C upper is unused.

Input

• Phone keypad input is used for manually entering various commands for the vehicle**.**

Output

• Depending on various inputs provided by the user via the keypad,the vehicle executes the user defined task with the help of various DC motors and Stepper motors.

Functioning:

Vehicle is controlled using the matrix keypad. When a key is pressed, the standard key-press detection algorithm (2 – key lockout followed by key de-bounce and finally detection) is used. Each of the keys corresponds to a function. The various functions of the vehicle can be summarized as follows:

Key	Function
1	Stop
2	Move Forward
3	Dig the soil (using front left wheel)
4	Move Left
5	360° rotation
6	Move Right
7	Move Arm Horizontally Right
8	Move reverse (back)
9	Move Arm Vertically Up
0	N/A
*	Move Arm Horizontally Left

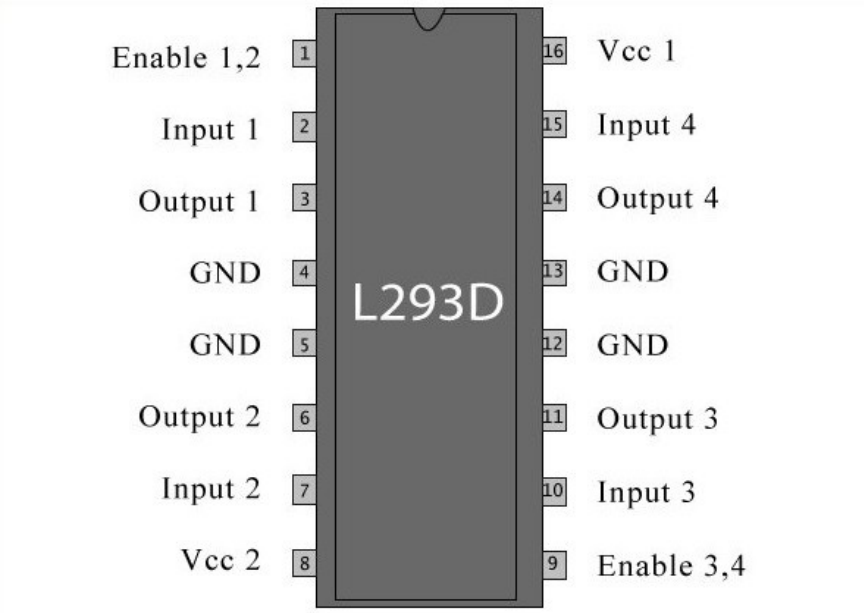
#	Move Arm Vertically Down

The vehicle can move in all directions and simultaneously move its hand in both vertical (Up and Down) and horizontal planes (Left and Right). It also checks its tilt angle and if it is greater than 45°, it is stopped.

HARDWARE DESCRIPTION

1. Push/Pull Four Channel Motor Driver(L293D):

As the MCUs PORT are not powerful enough to drive DC motors directly so we use L293D to drive them. It is a 16 PIN chip. The pin configuration is as follows:



This chip is designed to control 2 DC motors. There are 2 INPUT and 2 OUTPUT PINs for each motor. The following table shows the various uses:

Enable-1	Input1	Input2	Function
High	High	Low	Turn Clockwise(Reverse)
High	Low	High	Turn Anti Clockwise(Forward)
High	High	High	Stop
High	Low	Low	Stop
Low	X	X	Stop

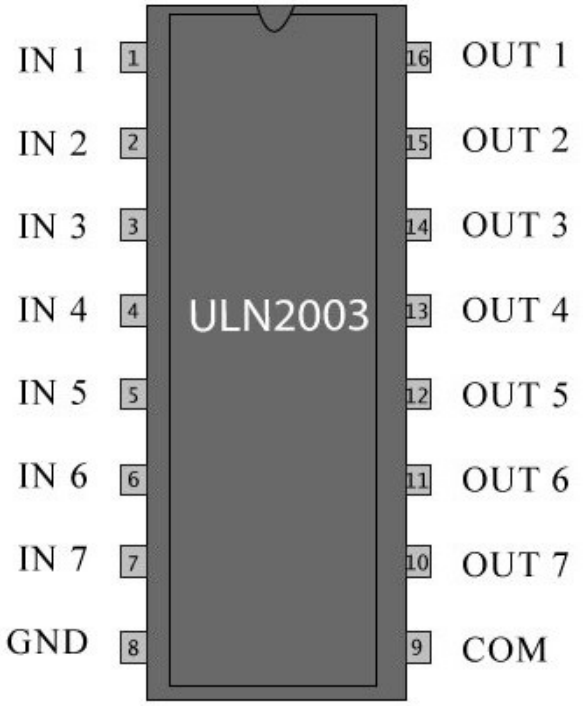
If Pin1 (E1) is low then the motor stops, irrespective of the states on Pin2 and Pin7, since both Input1 and Input2 become don't care. Hence it is essential to hold E1 high for the driver to function, or simply connect enable pins to positive 5 volts.

With Pin1 high, if Pin2 is set high and Pin7 is pulled low, then current flows from

Pin2 to Pin7 driving the motor in clockwise direction. If the states of Pin2 and Pin7 are flipped, then current flows from Pin7 to Pin2 driving the motor in anti- clockwise direction. Motors are connected to Pin11 and Pin14; Pin10 and Pin15 are input pins, and Pin9 (E2) enables the driver.

2. High voltage/High Current Darlington Transistor Array(ULN2003A)

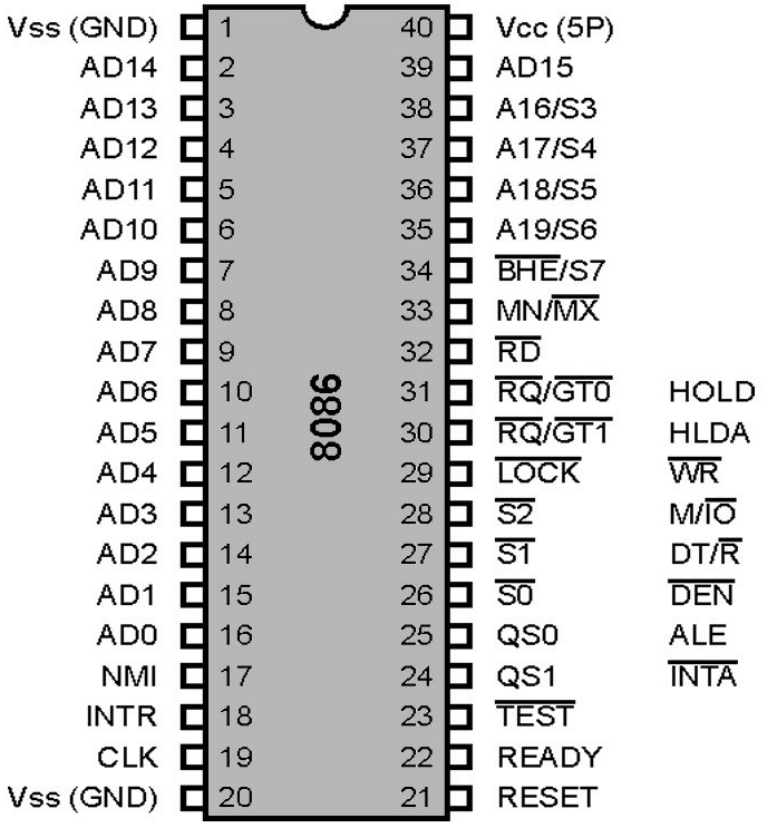
It consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode flyback diodes for switching inductive loads. It is used to drive the stepper motors in the circuit design that has been implemented here. The ULN2003A has a 2.7kΩ series base resistor for each Darlington pair for operation directly with TTL or5-V CMOS devices. Its pin out is shown below:



B are input pins and C are output pins. The inputs and outputs are provided opposite to each other in the pin layout. Each driver also contains a suppression diode to Dissipate voltage spikes while driving inductive loads. Pins 13,14,15,16 act as outputs here and go to the stepper motors in the design.

3. Microprocessor(8086)

8086 is a 16bit processor. It's ALU, internal registers works with 16bit binary word. All internal registers, as well as internal and external data buses, are 16bits wide, firmly establishing the "16-bit microprocessor" identity of the 8086. A 20-bit external address bus gave a 1 MB physical address space (2^{20} =1,048,576).
It is capable of working only in the real mode of operation.

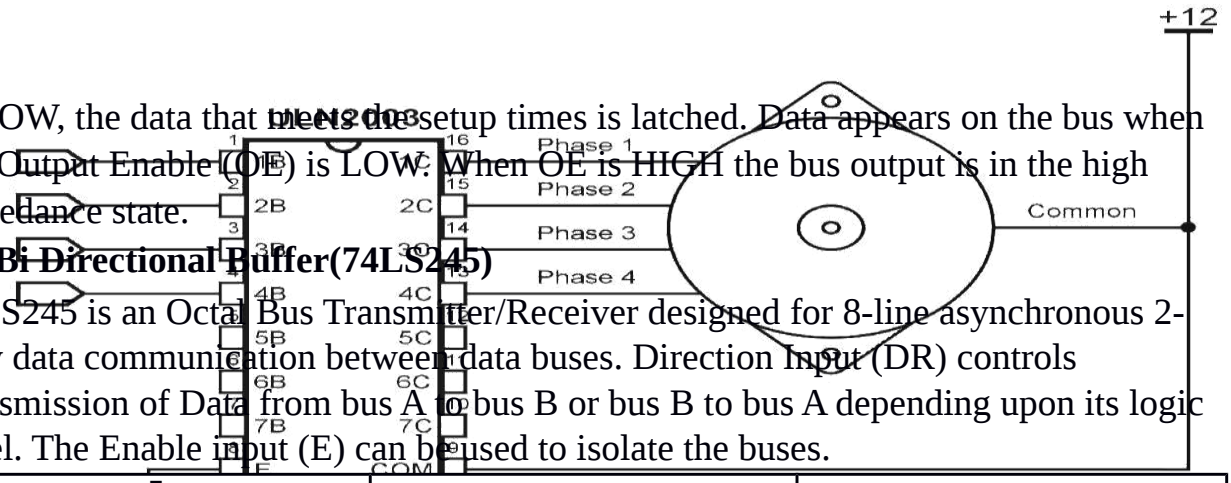


The data bus was multiplexed with the address bus in order to fit a standard 40-pin dual in-line package.

4. Latch(74LS373)

74LS374 is a high-speed, low-power Octal D-type Flip-Flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. The output of a latch is set to the logical state of its data input line a clock pulse arrives - and retains that state when the clock pulse has elapsed. A buffered Clock (CP) and Output Enable (OE) is common to all flip-flops. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE

is LOW, the data that meets the setup times is latched. Data appears on the bus when the Output Enable (OE) is LOW. When OE is HIGH the bus output is in the high impedance state.



5. Bi Directional Buffer(74LS245)

74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input (E) can be used to isolate the buses.

Ē	**DIR**	**Output**
L	L	BUS B to BUS A
L	H	BUS A to BUS B
H	X	Isolation

6.3*8 Decoder(LS138)

LS138 is a high speed 1-of-8 Decoder /Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The decoder accepts three binary weighted inputs (A0, A1, A2) and when enabled provides eight mutually exclusive active LOW Outputs (O0–O7). The LS138 features three Enable inputs, two active LOW (E1, E2) and one active HIGH (E3). All outputs will be HIGH unless E1 and E2 are LOW and E3 is HIGH. **Decoder selects one of the two 8255**

PIC. Address lines of the central address bus A3, A4, A5 goes as input into the decoder. Output Y0 is connected to the chip select (CS) of first 8255 and output Y1 is connected to the chip select of second 8255.

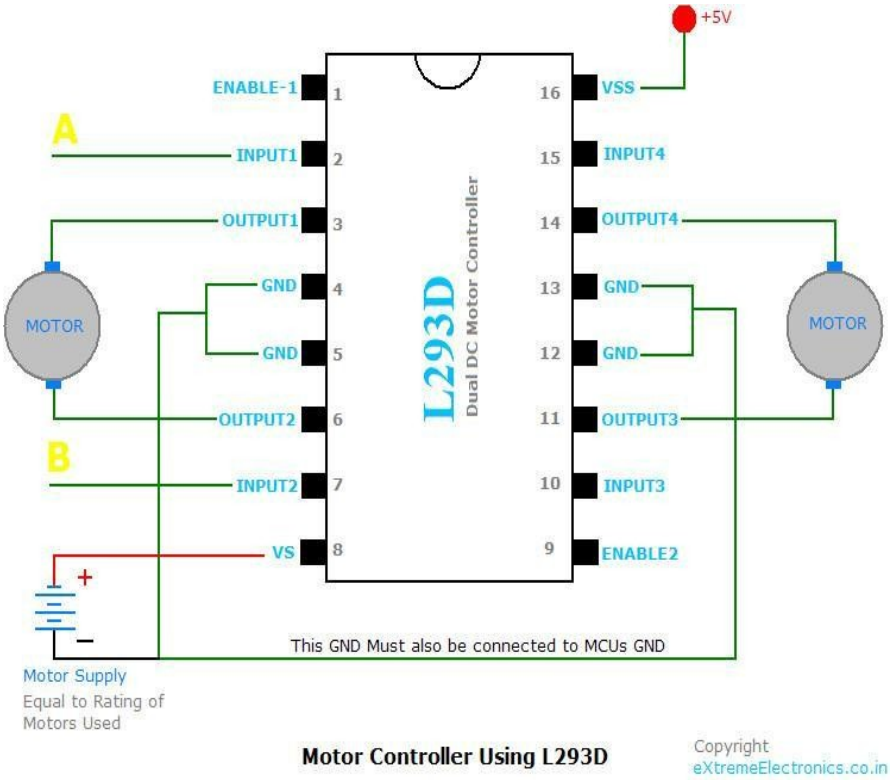
7. 12 Volt Unipolar Stepper Motor

Stepper motors are electromechanical devices that convert a pattern of inputs and the rate-of-change of those inputs into precise rotational motion. The rotational angle and direction for each change (step) is determined by the construction of the motor as well as the step pattern input.

It is interfaced using ULN2003A.

8. DC Motor

DC motor has a stationary set of magnets in the stator and an armature with a series of two or more windings of wire wrapped in insulated stack slots around iron pole pieces (called stack teeth) with the ends of the wires terminating on a commutator. **It is interfaced using L293D** as follows:



9. RF Modem (CC2420)

The CC2420 is a true single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver designed for low-power and low-voltage wireless applications. CC2420 includes a digital direct sequence spread spectrum baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps. RF modem is shown as a ZAT DART

which is a dual asynchronous receiver transmitter.

10. POT HG

It is a high granularity interactive potentiometer which is **used to simulate gyroscope used for measuring the tilt angle.**

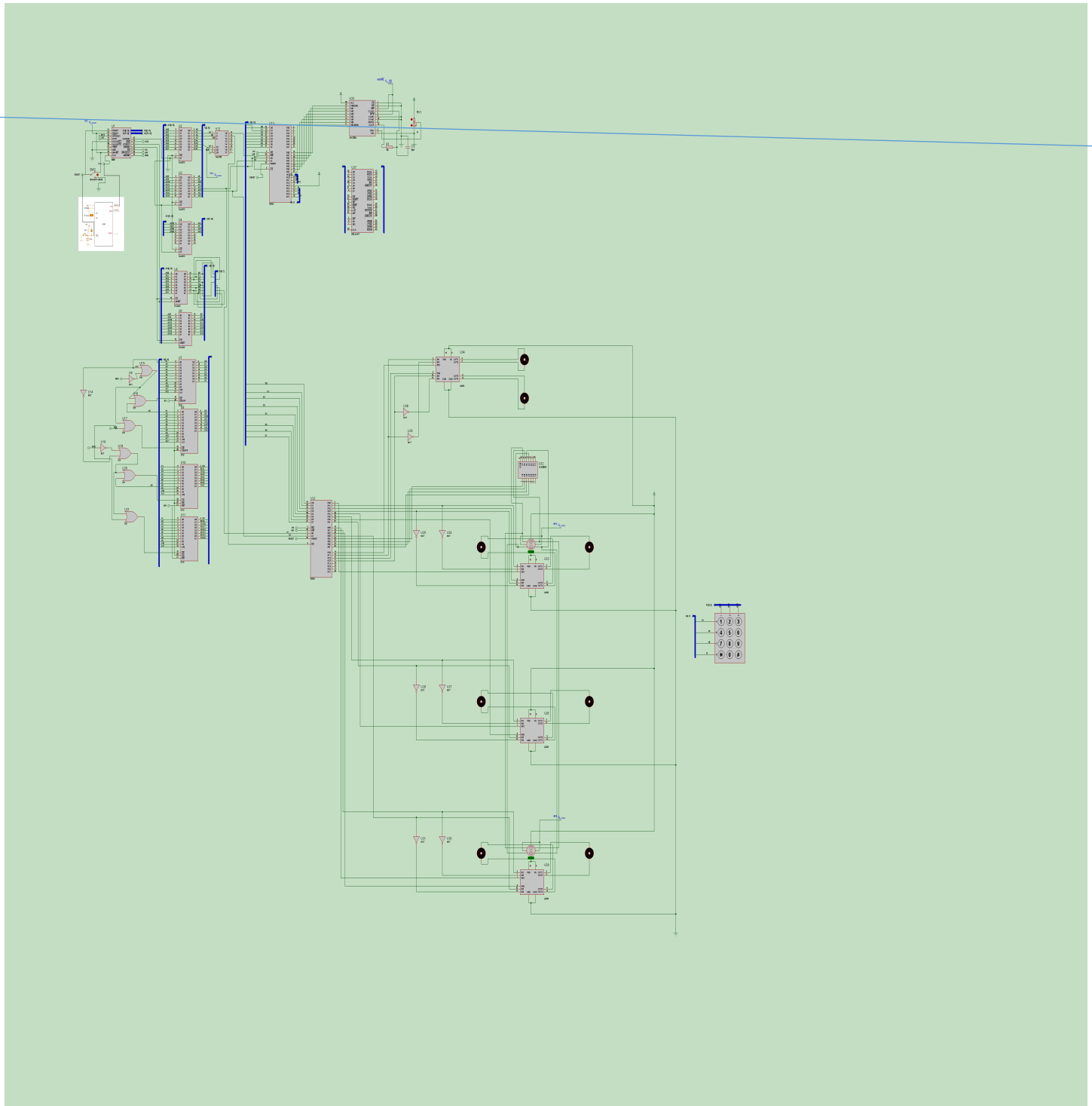
11. ADC0804 Analog to Digital Convertor

ADC0804 is a very commonly used 8-bit analog to digital convertor. It is a single channel IC, *i.e.*, it can take only one analog signal as input. The digital outputs vary from 0 to a maximum of 255. The step size can be adjusted by setting the reference voltage at pin9. When this pin is not connected, the default reference voltage is the operating voltage, *i.e.*, Vcc. The step size at 5V is 19.53mV (5V/255), *i.e.*, for every 19.53mV rise in the analog input, the output varies by 1 unit. To set a particular voltage level as the reference value, this pin is connected to half the voltage.

FLOWCHART OF THE SOFTWARE

Jkjlvkhjkhjk

Hardware Circuit Diagram:



Assembly Level Program:

```
#make_bin#

#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#

#PRTA1=00h#
#PRTB1=02h#
#PRTC1=04h#
#CREG1=06h#

#PRTA2=08h#
#PRTB2=0Ah#
#PRTC2=0Ch#
#CREG2=0Eh#

#CS=0000h#
#IP=0000h#

#DS=0000h#
#ES=0000h#

#SS=0000h#
#SP=FFFEh#

#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#

; add your code here
         jmp     st1
         db     509 dup(0)
         pos    db 33h
;IVT entry for 80H

         dw     t_isr
         dw     0000
         db     508 dup(0)

;main program

st1:      cli
; intialize ds, es,ss to start of RAM
          mov       ax,0200h
          mov       ds,ax
          mov       es,ax
          mov       ss,ax
          mov       sp,0FFFEH
```

```
;initialise
8255.....................................................................
...
    MOV AL,10001010B ; port A_input port c_upper_input portb_input port c
lower_output
    OUT CREG1,AL
    mov     al,80h  ;initialising the other 8255
    out     CREG2,al

;code for checking key
release.....................................................

;check for key release
    ;mov al,00h
    ;out 00h,al


X0: MOV AL,PRTA1
    OUT PRTC1,AL
X1: IN AL, PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JNZ X1

call delay2ms

;check for key press
    MOV AL,PRTA1
    OUT PRTC1 ,AL
X2: IN AL, PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JZ X2

call delay2ms

;check for key press
    MOV AL,PRTA1
    OUT PRTC1 ,AL
    IN AL, PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JZ X2

;check for key press in column1
    MOV AL, CREG2
    MOV BL,AL
    OUT PRTC1,AL
    IN AL,PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JNZ X3

;check for key press in column2
    MOV AL, 0DH
    MOV BL,AL
```

```
    OUT PRTC1 ,AL
    IN AL,PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JNZ X3

;check for key press in column3
    MOV AL, 0BH
    MOV BL,AL
    OUT PRTC1,AL
    IN AL,PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JNZ X3

;check for key press in column4
    MOV AL, 07H
    MOV BL,AL
    OUT PRTC1,AL
    IN AL,PRTC1
    AND AL,0F0H
    CMP AL,0F0H
    JZ X2

;decode key
X3: OR AL,BL ;concatinating the rows and columns, thus al register now
contains the hex code for the key pressed

y1: cmp AL,0EEh ;key 0
    jnz y2
    call stop

y2: cmp AL,0EDh
    jnz y3
    call forward ;key 1

y3: cmp AL,0EBh
    jnz y4
    call dig ;key 2

y4: cmp AL,0DEh
    jnz y5
    call left ;key 4

y5: cmp AL,0DDh
    jnz y6
    call spin ;key 5

y6: cmp AL,0DBh
    jnz y7
    call right ;key 6

y7: cmp AL,0BEh
    jnz y8
    call arm_horizontal_right ;key 8
```

```
y8: cmp AL,0BDh
    jnz y9
    call back ;key 9

y9: cmp AL,0BBh
    jnz y10
    call arm_vertical_up ;key 10

y10: cmp AL,07Eh
    jnz y11
    call arm_horizontal_left ;key 12

y11: cmp AL,07Dh
    jnz y12
    call delay2ms ;key 13       ;empty key, can be used later

y12: cmp AL,07Bh
    jnz y13
    call arm_vertical_down ;key 14

y13: JMP X0

;subroutines from here on

delay2ms proc near
    PUSH CX
    mov cx,00ffh
xx2: nop
    nop
    loop xx2
    POP CX
    ret
delay2ms endp

;STOP
stop proc near
           mov al,PRTA1
           out PRTA2,al
           out PRTB2,al
           out PRTC2,al
            ret
stop endp

;FORWARD
forward proc near
          mov       al,0ffh
          out       PRTA2,al
          mov       al,0fh
          out       PRTB2,al
          ret
forward endp

;BACKWARD
back proc near
    mov al,55h
    out PRTA2,al
```

```
    mov al,05h
    out PRTB2,al
    ret
back endp

;DIG
dig proc near
    mov  al,PRTC2
    out  PRTA2,al
    ret
dig endp

;360
spin proc near
   mov  al,0ddh
   out  PRTA2,al
   mov  al,0dh
   out  PRTB2,al
   ret
spin endp

;RIGHT
right proc near

   mov al,PRTC2
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,09h
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,03h
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,CREG1
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   ret
right endp

;LEFT
left proc near

   mov al,CREG1
```

```
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,03h
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,09h
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

   mov al,PRTC2
   rol al,4
   and al,0f0h
   out PRTB2,al
   call delayMor

       ret
left endp

;ARM UP
arm_vertical_up proc near
    mov al,03h
    out PRTC2,al
    ret
arm_vertical_up endp

;ARM DOWN
arm_vertical_down proc near
    mov al,01h
    out PRTC2,al
    ret
arm_vertical_down endp

;ARM LEFT
arm_horizontal_left proc near
    mov al,PRTC1
    out PRTC2,al
    ret
arm_horizontal_left endp

;ARM RIGHT
arm_horizontal_right proc near
    mov al,PRTC2
    out PRTC2,al
    ret
arm_horizontal_right endp

delayMor proc near
```

```
    MOV BX,1000
    xdel: DEC BX
    CMP BX,0000
    JNZ xdel
    RET
delayMor endp

t_isr:
iret
```

Jkvjhvkhjkhjk

References:

- Microprocessor Programming & Interfacing, BITS Pilani CS/ECE/EEE/INSTR F241 course.
- http://www.ccse.kfupm.edu.sa/~hazem/coe305/74373.pdf
- https://www.ece.cmu.edu/~ece348/labs/docs/SN74LS245.pdf
- http://www.futurlec.com/Memory/2732.shtml
- http://en.wikipedia.org/wiki/Intel_8255
- http://users.ece.utexas.edu/~valvano/Datasheets/L293d.pdf
- http://www.ti.com/lit/ds/symlink/uln2003a.pdf
- http://esd.cs.ucr.edu/labs/general/ADC0801.pdf
- http://ecee.colorado.edu/~mcclurel/sn74ls138rev5.pdf
- http://www.datasheetarchive.com/6116%20memory-datasheet.html