Unit - II

Interrupt & Timer:-

Timer:-

Pic 16C74 has 3 modules

Timer 0

Timer 1

Timer 2

Timer 0 & Timer 2 are 8 bit Timer

Timer 1 is a 16 bit Timer.

Timer 0:-

Timer 0 module is a simple 8 bit up counter. The clock source can be either internal clock or an external clock. when clock source is external, Timer 0 module can be programmed to increment on either the rising or Falling clock edge. Timer 0 has a programmable Pre scaler option. This Pre scaler can be assigned either to Timer 0 or the watch dog timer.

Timer 0 counter sets a Flag TOIF when it overflow & can cause a interrupt at that time. (ie) timer - 0 bit =

Option:- Register Configuration:-

Option register control the prescalar to Timer 0 clock source. The following option register configuration is for clock source to no watch dog Timer.

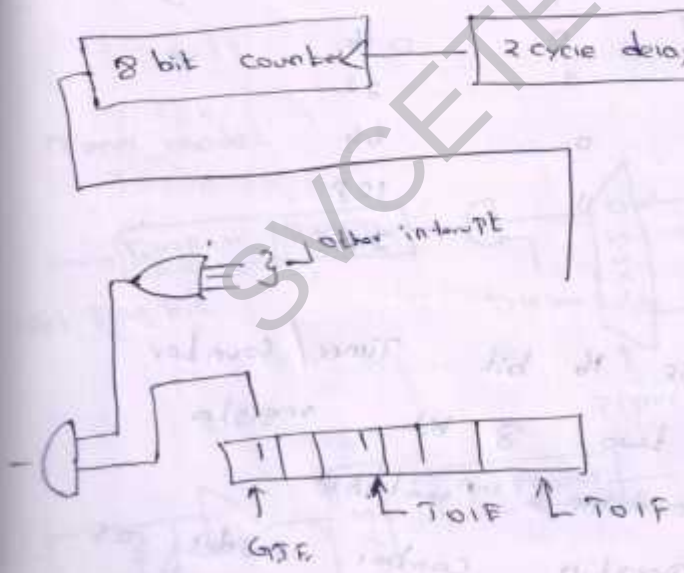
RBPU INTEDG 0 T0SE 1 PS2 PS1 PS0

PSA

T0CS 1 Timer 0 clock source
0 Timer 0

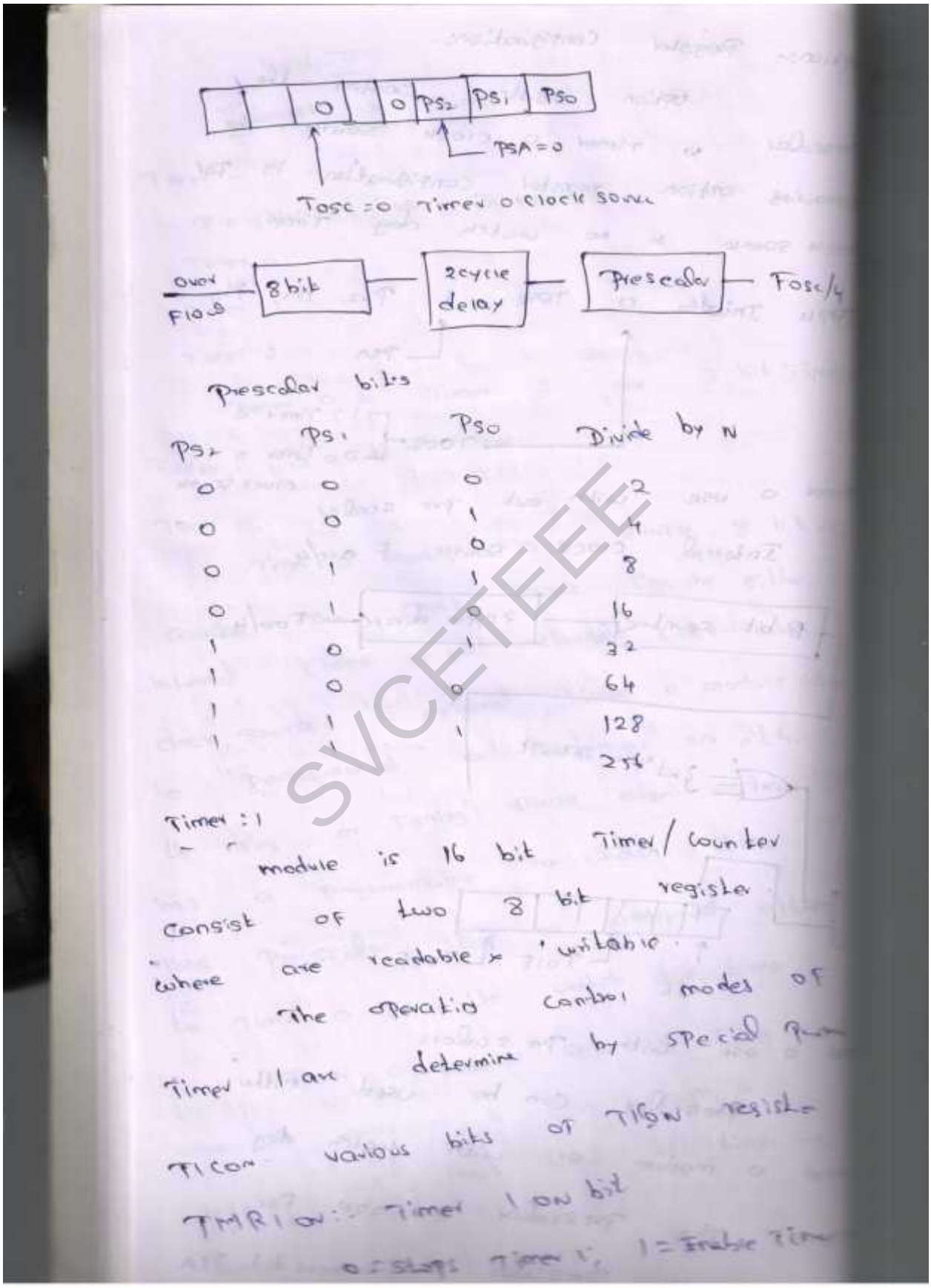
Timer 0 use with out Pre scalar

Internal clock source F osc/4.



Timer 0 use with Pre scalar:-

Prescalar can be used either Timer 0 module or with watch dog Timer. The prescalar available for assignment bit PSA



Prescaler bits

PS2	PS1	PS0	Divide by N
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

Timer : 1

module is 16 bit Timer / Counter

Consist of two 8 bit register

where are readable & writable

the operation control modes of

Timer 1 are determine by special

T1CON various bits of T1CON register

TMR1ON : Timer 1 ON bit

0 : stops Timer 1, 1 = Enable Tim

-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC

1 = Do not synchronize

0 : Synchronize

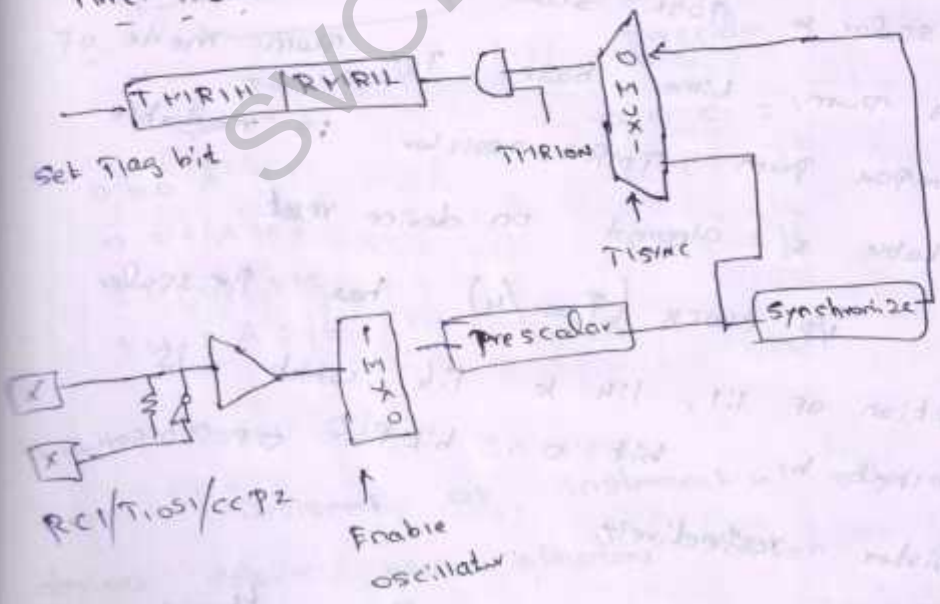
T1OSCEN :- oscillator enable control ckt

1 - oscillator is enable.

0 - oscillator is shut OFF.

Select bits		Prescalar value
T1CK PS1	T1CK PS0	
1	1	1:8
1	0	1:4
0	1	1:2
0	0	1:1

Timer mode:



Timer mode:-

As a Timer mode Timer 1 increment

Counter Mode:-

As a counter in counter mode, external clock i/p from pin T1CKI is select

Reading & writing Timer 1:-

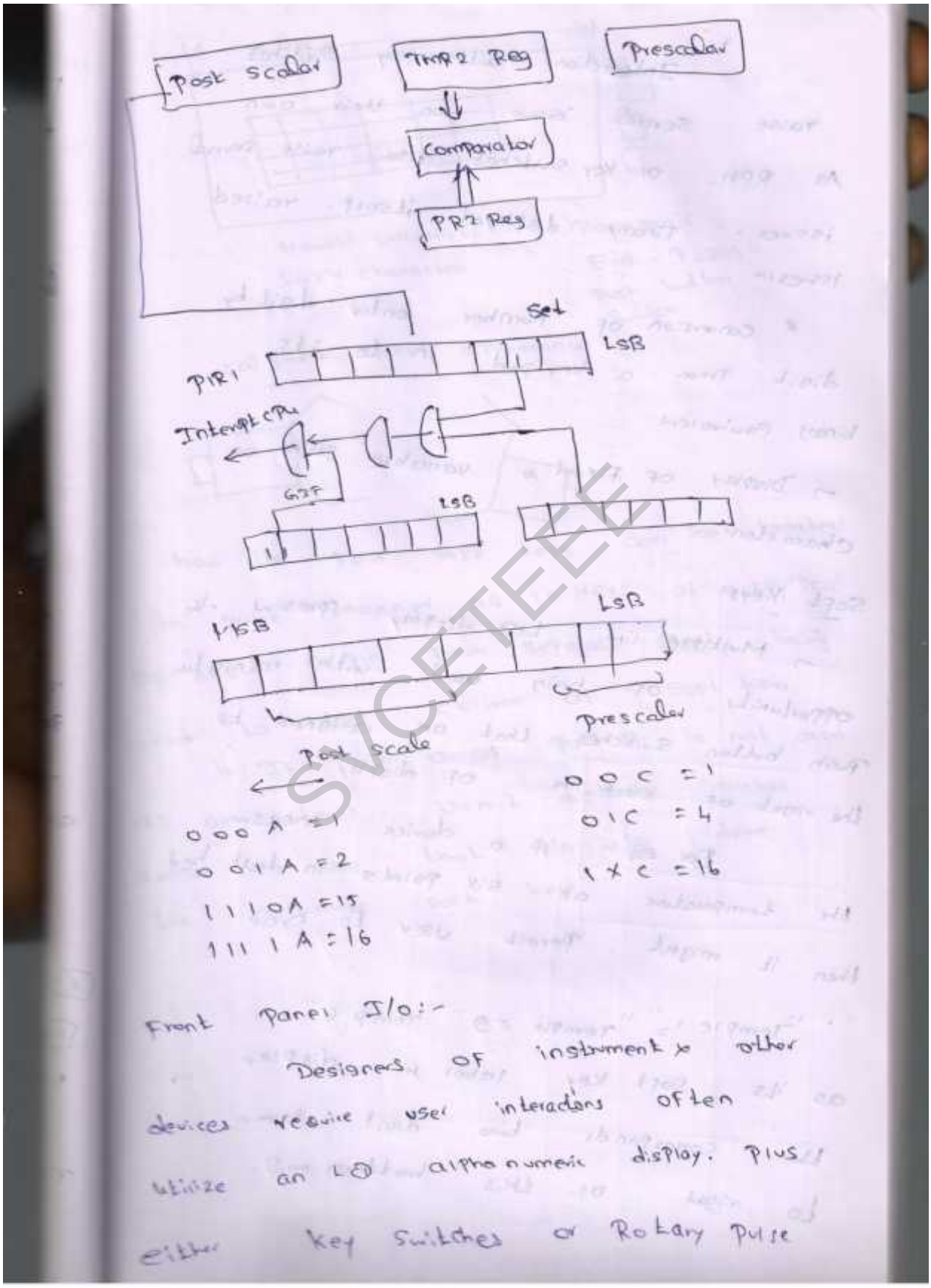
Reading TMR1H & TMR1L from Timer 1, when it is running from external clock source have to be done with care. Reading TMR1H or TMR1L for independent 8 bit value does not any problem. when 16 bit value of Timer.

Timer 2

Timer 2 is an 8 bit Timer with Prescaler & post scaler. It can be used as PWM time base for PWM mode of Compare PWM TMR2 register is readable writable & cleared on device reset.

i/p clock (Fosc/4) has prescaler option of 1:1, 1:4 & 1:6 which is select by bit 0, bit 1, of T2 con register respectively.

The o/p of TMR2 through A bit post scaler (1:1, 1:2 to 1:16)



Front Panel I/O:-

Designers of instrument & other devices require user interactions often utilize an LCD alpha numeric display. Plus either key switches or Rotary Pulse

Interaction with key switches

raise several issue on their own As RPG, or key switches, also raise several issues. Finally displayed itself raised issues:-

* conversion of number enter digit by digit from a key pad in to its binary equivalent

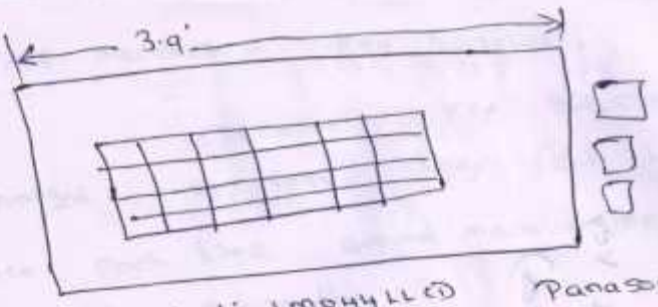
→ Display of Fixed & variable string of character.

Soft Keys:

→ Multiple like LCD display present the opportunity of being used with miniature Push button switches that are aligned to the right of each line of display.

For eg. if a device measure the temperature at six points in test bed then it might permit user to cycle

"Temple 1" = "Temple 2" "Temp 6" as its soft key label & display the corresponds two digit temp to right of this bottom row.



Hitachi LM044LL (1)
20x4 character

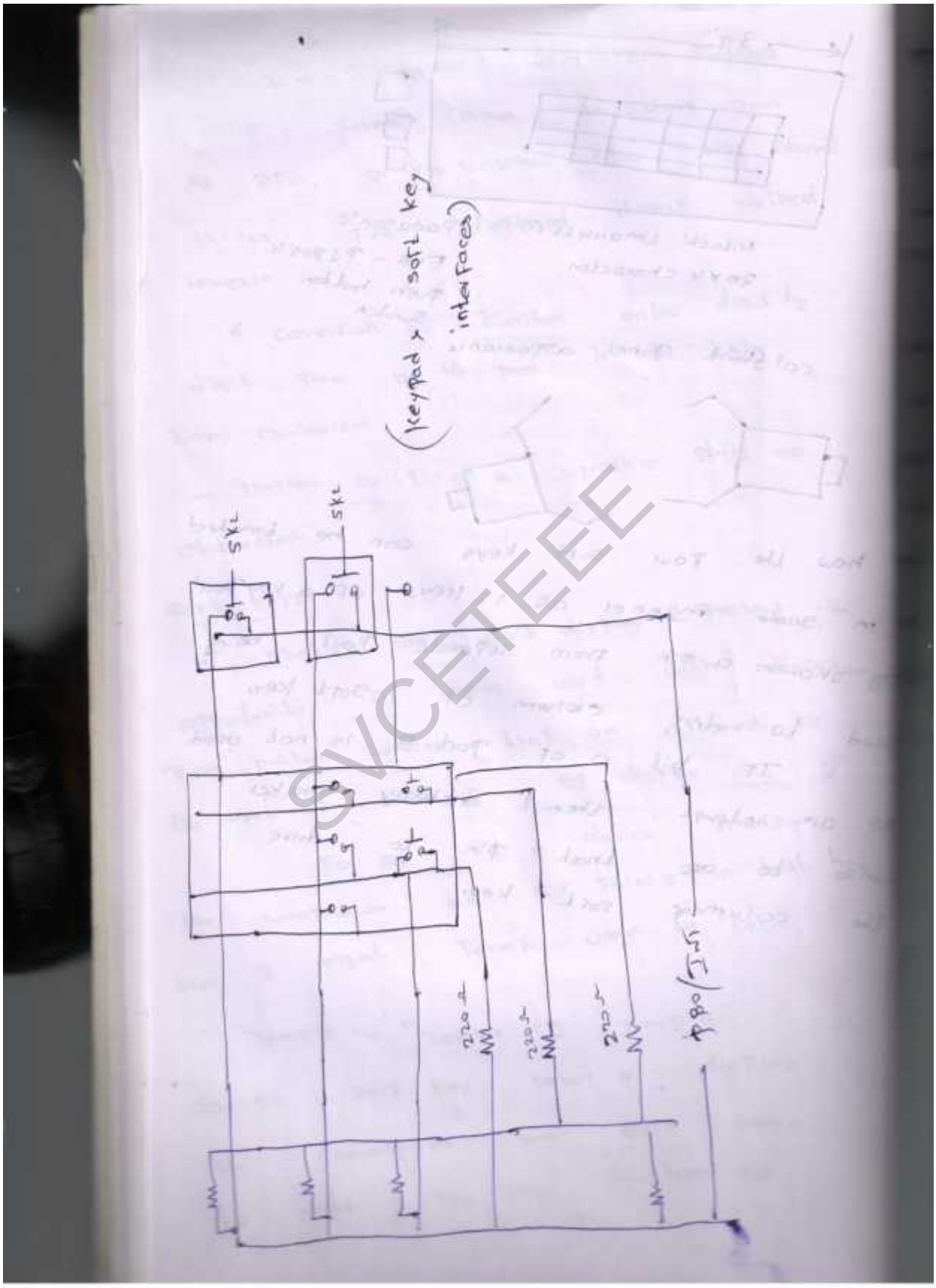
Panasonic
EVQ - PJ805K
Push button
Switch

(a) Front Panel appearance

how the four soft keys can be treated in same manner as 12 keys of a keypad

Fig shows a pin from seperate Port being used to drive column of soft keys

If bit 0 of Port B is not used as an input then it actualy makes sense to use that pin to drive the column of soft keys



STATE machines & Key switches:-

Because key switches are not charged very fast, they can be checked once each time around main line look in a key switch subroutine.

Recall that loop time at 10ms was select because the minimum key bounce time of most mechanical key switch is less than 10 ms. Consequently if key switch detect that key is newly pressed it can be assured that next time it is called 10ms later, any erratic bouncing of the key contact will have settle out.

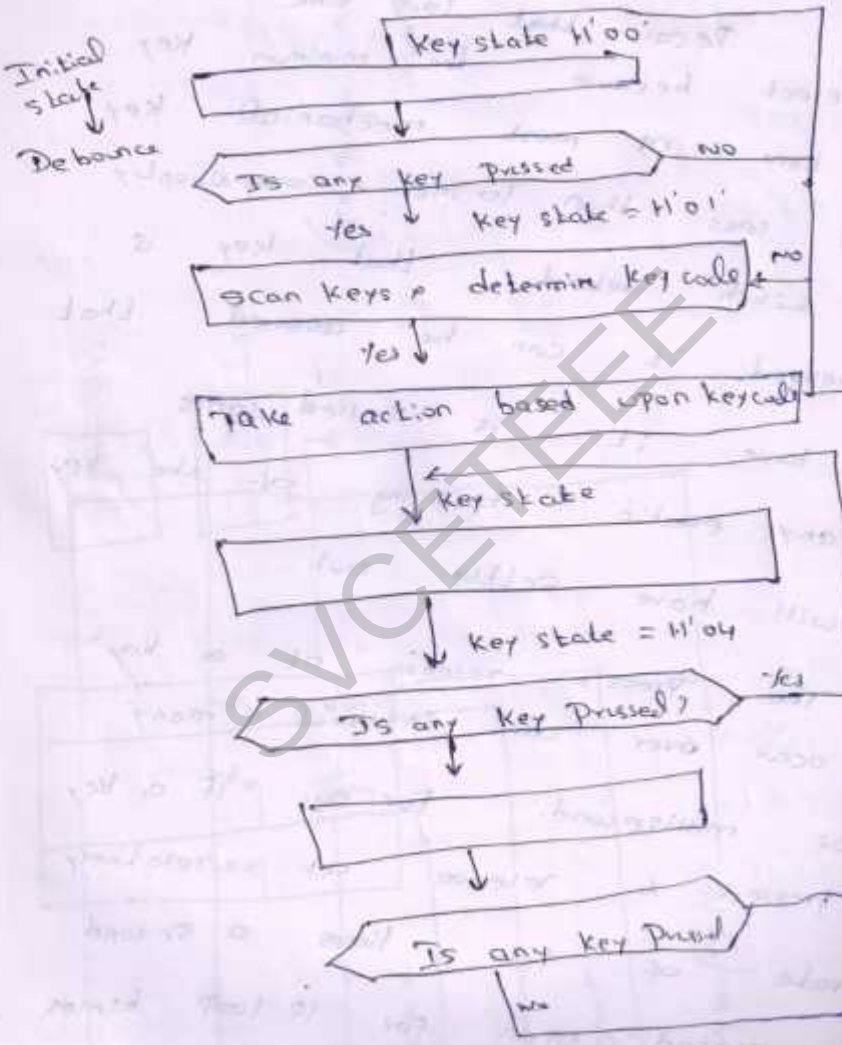
The press & release of a key switch occur over an interval many tens of millisecond. For eg: if a key Switch pressed & release at relatively Fast rate of four times a second. may be closed open for 12 loop times, closed for 12 loop times etc.

The key switch subroutine will use a state variable called state to keep track.

Denounce the key switch

Determine which key is pressed

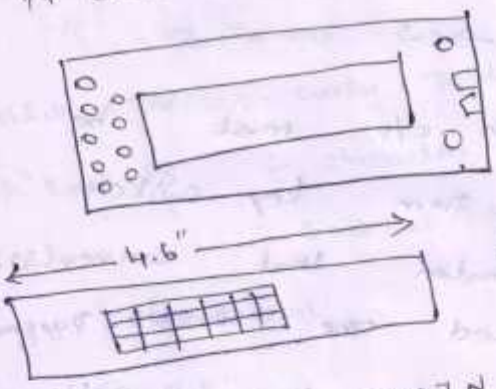
Take appropriate action once for that press of key.



Key switch subroutine algorithm.

Display Plus RPG use:-

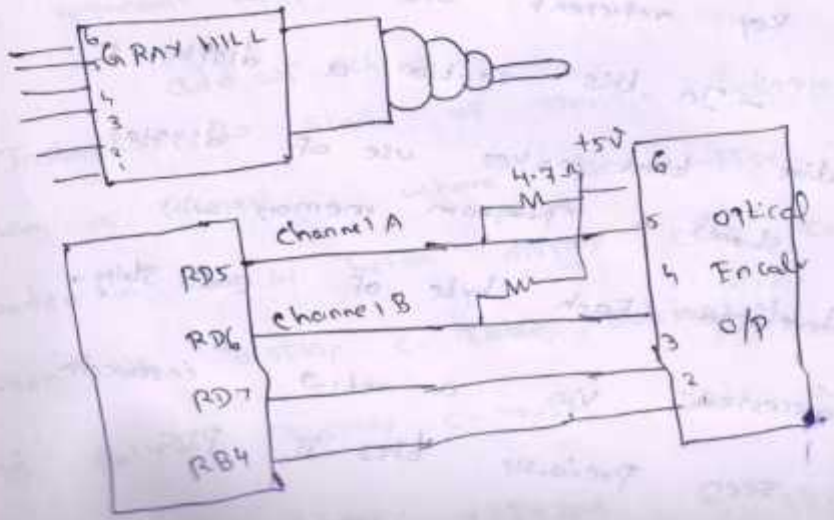
Another popular display - Plus - I/P device

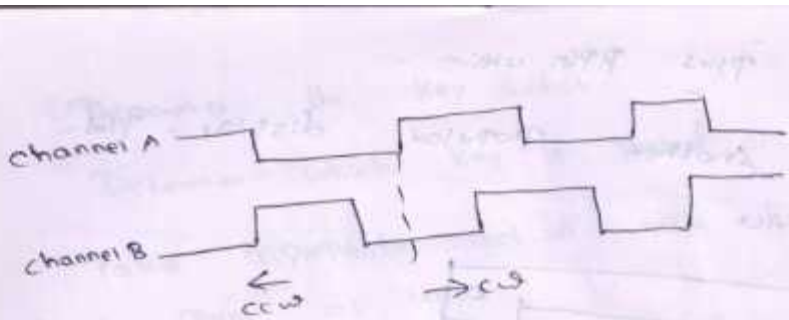


Oplrex DMC - 24227 N

Gray hill rotary pulse generator

shown here. Fig differ from packard RPG. It feature 24 detented position per evolution so it clicks from action push button switch. when used with an display switch RPG's knob alpha numeric





Two RPG o/p must be treated differently from key switch since the information that convey is They must be treated the selected parameter can be changed in large steps when the RPG for this application the incrementing of RPG- interval must be stopped when it reaches HFF

Display of Constant strings:

Constant strings arise in several ways The labels associate with soft key represent one application.

- In this section a display subroutine that makes use of display strings stored in program memory will be developed Each byte of each string is accessed via a retlw instruction As seen previously this is PIC

The Source code Form of a display string stored in Program memory can be illustrated following eg used to display string.

Row 4 coll

relw H'D4' :- cursor Positioning code

dt "Row4 cdl :- characters to be display

relwo : End of string

dt 'Row 4 coll'

relw A'R'

relw A'O'

relw A'W'

relw A'C'

relw A'O'

relw A'1'

relw A'1'

uses the initial underline character to serve as a reminder that entities such as represent labels for constant display strg

add wf pcl, F

Instruction to start of specific display string of interest. where this offset is loaded in to w before display c is called. when the Display c. Table subroutine is called Display c-routine will add to address represent by cDg.

```
Display c Table Subroutine
;
This subroutine is called with c pointer
contain offset CDS to desired byte
; Locate this table subroutine
; space
Display c Table
movf    cPointer w ;  copy pointer to w
addwf   PCL,F         change PC with PCLATH =
                                    H'00'
retlw   H'80' ; cursor position code
Row1 col2
        retlw   H'80'
        dt      "Row1 col2"
        retlw   0
- Row2 col7
        retlw   H'c6'
        dt      "Row2 col7"
        retlw   0
- Row4 col11
        retlw   H'D4'
        dt      "Row4 col1"
        retlw   0
```

Display C Subroutine

:- this subroutine is called
w contain OFF set from cDS to
:. beginning of desired constant
display string. It use one byte
:- c Pointer variable to hold the
Pointer to byte
:- Assume to circuit.

C-Pointer :- save Pointer
bcf PORT C :- Drive LCD.
incf cPointer, F : Point to next byte.
iorlw 0 :- set z if end of string
goto display c_dom
bdf ss PIR1, SSPTF
goto Display c_2.
bcf PORTD, 7 :- Drive LCD's E Pin Low
call T120 :- Give LCD 120 µs
bcf PORT C, 4 :- Drive LCD R's Pin
goto Display c_1
Display c_dom
return

Display of variable string:-

when entering setup Parameter with either a key pad on a Reg & when displaying results of an instrument measurement. it is necessary to write a string of Ascii code character

→ Cursor - Position code

→ Ascii string characters displayed

→ End of string.

VSTstring : length equiz : Max No of character VSTZ

(a) Fanale section of code:-

Clock Bank 0 RAM

:

VSTRING : VSTRING length

marker

endc

if marker > 'if 80'

error RAM use exceed Bank 0"

end if

(b) Assign a variable string.

Assign a variable string to RAM using version 2.00 or higher of microchip

VSTRING	H'F'4	Cursor - Positioning - Code
VSTRING +1	H'33'	
VSTRING +2	H'31'	Four digit number
VSTRING +3	H'32'	
VSTRING +4	H'36'	
VSTRING +5	H'00'	End of string designator

PIC microcontroller interrupt:

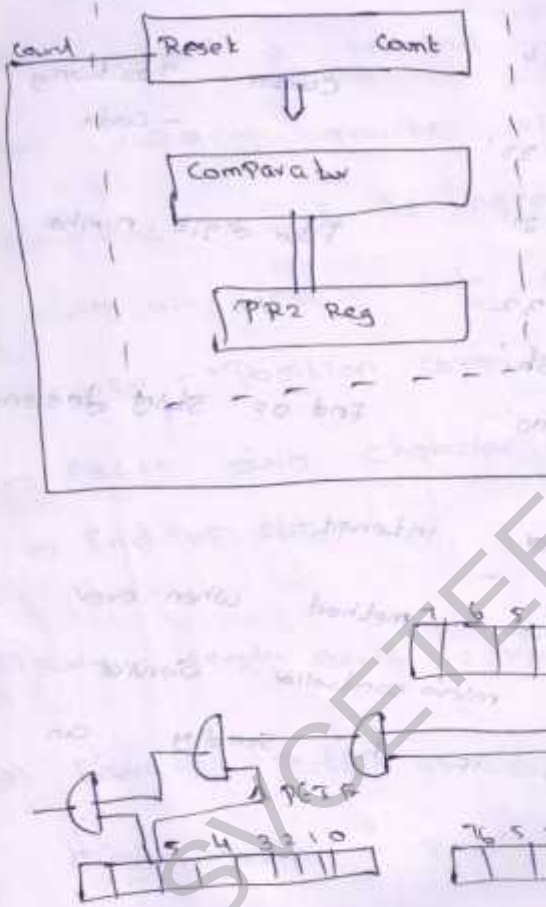
In interrupt method whenever any device need microcontroller service the device notifies it by sending an interrupt signal.

Merits interrupt:

- Advantage of interrupt is microcontroller serve many device at same time of course.
- The interrupt method the microcontroller time by polling devices.

A waste of microcontroller time that could have been used to perform some useful task.

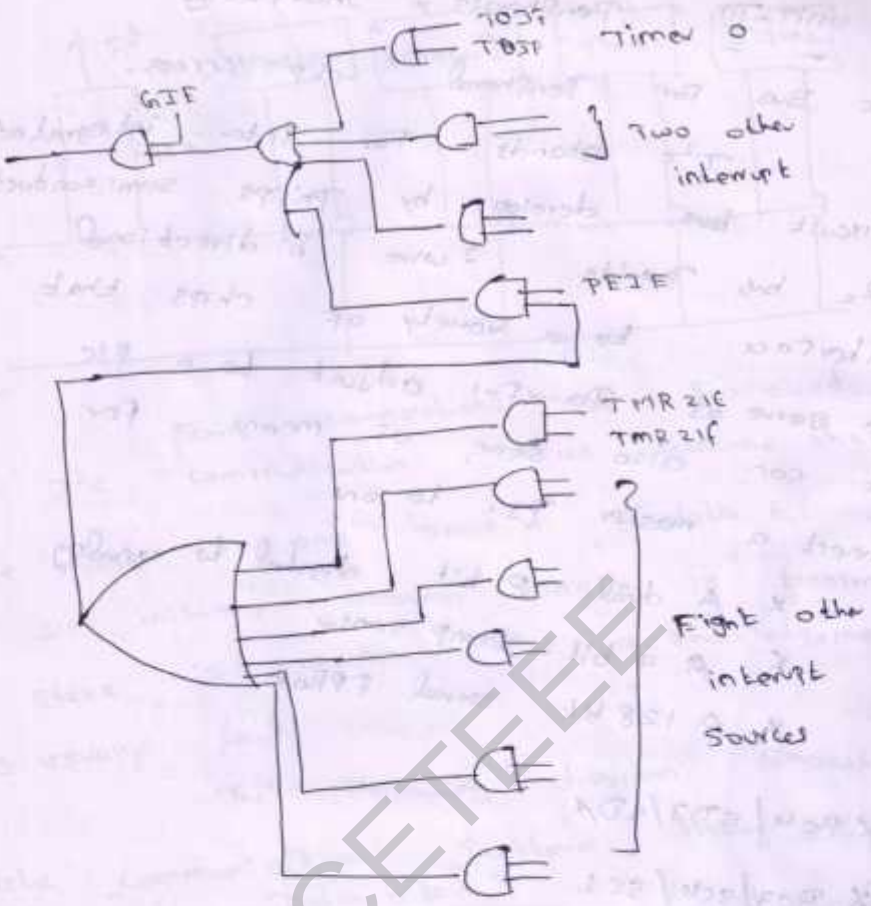
Interupt logic



bsf INTCON, GIE Global Interupt

INTCON, GIE Global interupt

GIE automatically cleared after an interupt occur Suspend Further interupt for the duration of interupt



Int service : This interrupt service routine

mov wf w_Temp : set aside w & status

: address will not used in mainline

: copy to RAm

: Execute

btfs PIR1, TMR2IF : Check for Timer 2

call Timer 2 : If ready

: btfsc

: call

swapf STATUS_Temp : - Restor STATUS bits

movf status : with out affect zbit