```
;*******************************************************
;Program Name:  "magic.asm"
;Description:
;
;Author:        Jacob Watts & Jack Fernald
;Organization:  Weber State University ECE 3710
;Revision History
;Date[YYYYMMDD] Author          Description
;----           ------          -----------
;20230221       Jacob W.        initial commit
;*******************************************************
$include (c8051f020.inc)
;------------------------ TODO --------------------------

;--------------------- Registery ------------------------
;R0 - clr RAM but only in the beginning
;R1 - Counter for 10
;R2 -
;R3 -
;R4 -
;R5 -
;R6 -
;R7 -
;------------------------ DSEG --------------------------
        DSEG AT 30H
current_button: ds 1
last button:    ds 1
led position:   ds 1
rand_int:       ds 1

;------------------------ CSEG --------------------------
        CSEG
;------- Crystal Setup Code -----------------------------
        mov wdtcn,#0DEh ; disable watchdog
        mov wdtcn,#0ADh
        mov xbr2,#40h ; enable port output
        mov xbr0,#04h   ; enable uart 0
        mov oscxcn,#67H ; turn on external crystal
        mov tmod,#21H   ; wait 1ms using T1 mode 2
        mov th1,#256-167    ; 2MHz clock, 167 counts = 1ms
        setb    tr1
wait1:
        jnb tf1,wait1
        clr tr1     ; 1ms has elapsed, stop timer
        clr tf1
wait2:
        mov a,oscxcn    ; now wait for crystal to stabilize
        jnb acc.7,wait2
        mov oscicn,#8   ; engage! Now using 22.1184MHz

        mov scon0,#50H  ; 8-bit, variable baud, receive enable
        mov th1,#-6     ; 9600 baud
        setb    tr1     ; start baud clock

        ;clear all internal ram
        mov     r0,#255
clrall: mov     @r0,#0
        djnz    r0,clrall
;---------------PLACE CODE BELOW THIS LINE---------------

;---------------- Initialization Code -------------------
init:
        mov A, #0FFh
        mov P3, A
        mov P5, A
        RI1 bit 0C0h ;turn on flag for serial send
        mov R1, #10

;--------------------- Main Code ------------------------
main:
        call delay 10ms
        call check buttons
        mov A, current_button
        cjne A, last_button, cont
            jmp serial_check

cont:   cjne A, #001h, serial_check ;if button pressed jmp
            jmp tx sub
; ----- This is where we check for
serial_check:
        jnb ri, main ;jump to main if ri not set
        clr ri
        jmp tx_sub

;------------------------- tx ---------------------------
tx_sub:
        call update_disp
        call get_address

        setb TR1
fn:     clr A
        movc A, @A+DPTR
        jz main
        call sendcom
        inc DPTR
        sjmp fn
sendcom:
        mov sbuf0,a
here1:  jnb TI, here1
        clr TI
        ret

;---------------------- LED Display ---------------------
update_disp:
                mov led_position, rand_int
                mov A, led_position
update_p3_0:    cjne A, #0Ah, update_p3_1
                    mov A, #02h
                    mov P3, A
                    mov A, #0FFh
                    mov P5, A
                    ret

update p3 1:    cjne A, #01h, update_p5
                    mov A, #01h
                    mov P3, A
                    mov A, #0FFh
                    mov P5, A
                    ret

update p5:      mov dptr, #led table
                movc A, @A+dptr
                mov P5, A
                mov A, #0FFh
                mov P3, A

                ret
;-------------------- get string addr --------------------
get_address:
            mov A, rand_int
            rl A
            mov B, A
            mov dptr, #msg_idx
            movc A, @a+dptr ;msb of address
            xch a,b
            inc a
            movc a,@a+dptr
            mov dpl, a
            mov dph, b
            clr a
            clr c

            ret

;-------------------- message table ----------------------
msg_1: DB "It is certain", 0Dh, 0Ah, 0
msg_2: DB "You may rely on it", 0Dh, 0Ah, 0
msg_3: DB "Without a doubt", 0Dh, 0Ah, 0
msg_4: DB "Yes", 0Dh, 0Ah, 0
msg_5: DB "Most likely", 0Dh, 0Ah, 0
msg_6: DB "Reply hazy, try again", 0Dh, 0Ah, 0
msg_7: DB "Concentrate and ask again", 0Dh, 0Ah, 0
msg_8: DB "Don't count on it", 0Dh, 0Ah, 0
msg_9: DB "Very doubtful", 0Dh, 0Ah, 0
msg_10: DB "My reply is no", 0Dh, 0Ah, 0

;------------------- msg idx table ----------------------
msg_idx: DW msg_1, msg_2, msg_3, msg_4, msg_5, msg_6, msg_7, msg_8, msg_9, msg_10

;--------------------- LED table -------------------------
led_table: DB 0FFh, 0FFh, 0FEh, 0FDh, 0FBh, 0F7h, 0EFh, 0DFh, 0BFh, 07Fh

;------------------- Check Buttons ----------------------
check_buttons:  MOV A, P1
                MOV last_button, current_button ; Memory for next loop
                CPL A ; Complement A
                ANL A, #001h ; Mask out other bits from P1, we only care about the single button bit. Anding it with 1 will keep the state of the button
                ; e.g. button = 1 and with 1 = 1, button = 0 and with 1 = 0. T
                MOV current_button, A ; Move this newly found button state into current_button
                RET

;-------------------- 10ms Delay ------------------------
delay_10ms:
            djnz R1,here
            mov R1,#10

here:       mov TL0, #000h ;-9216 for 5ms
            mov TH0, #0DCh
            setb TR0 ;start timer
again:      jnb TF0, again
            clr TR0
            clr TF0

            mov A, R1
            mov rand_int, A
            clr A; for good measure
            clr C; for good measure
            ret

            END
```