

Lecture (05)

Practical applications on PIC16F8 using Assembly Language

Dr. Ahmed ElShafee

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

Agenda

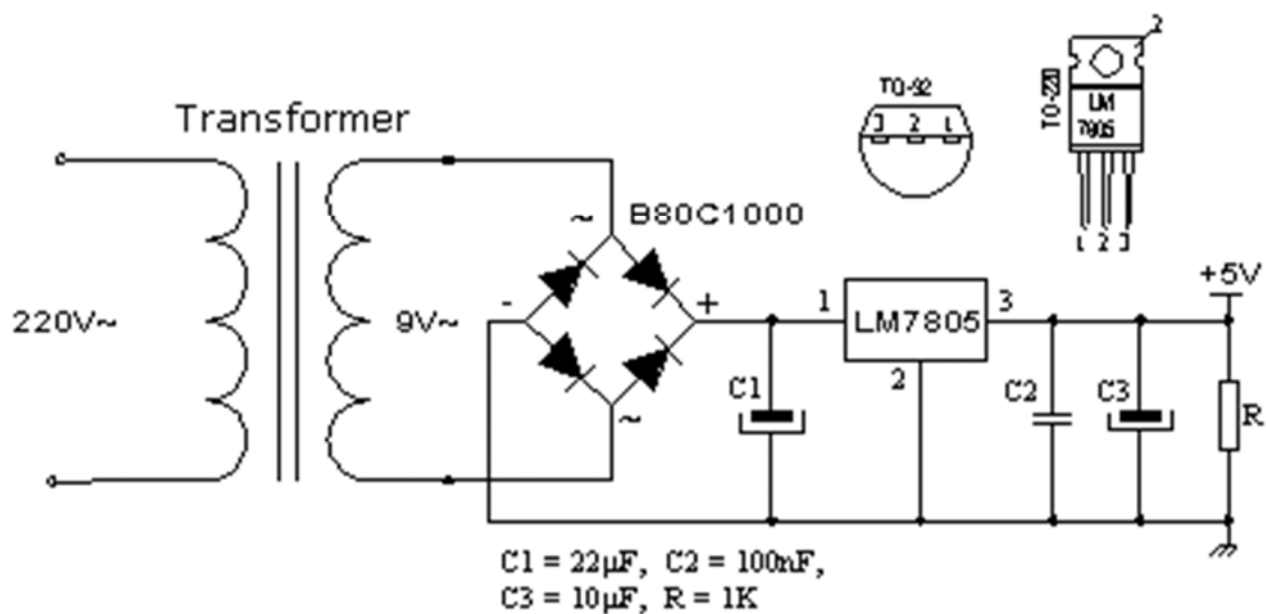
- *
- *
- *

Supplying the microcontroller

- For a proper function of any microcontroller, it is necessary to provide a stable source of supply
- supply voltage should move between 2.0V to 6.0V in all versions
- the source of supply is using the voltage stabilizer LM7805 which gives stable +5V on its output

٢

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

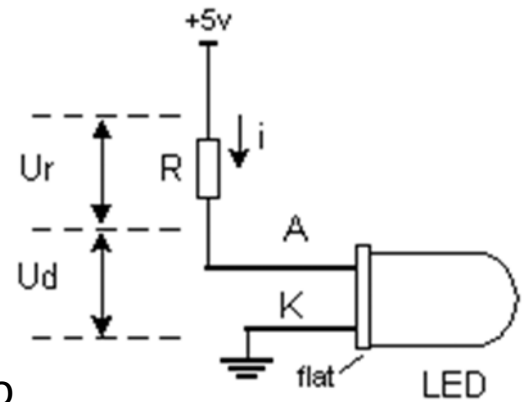


٤

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

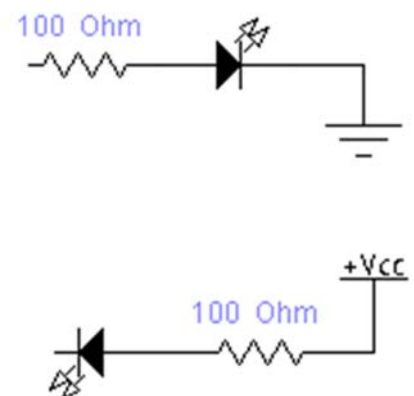
LED diodes

- Two types based on its diameter, which is usually 3 or 5 mm
- working current which is usually about 10mA
- current-limiting resistor must be the correct value so that the LED is not damaged or burn out (overheated).
- The voltage drop will range from 1.2v to 1.6v depending on the color of the LED.

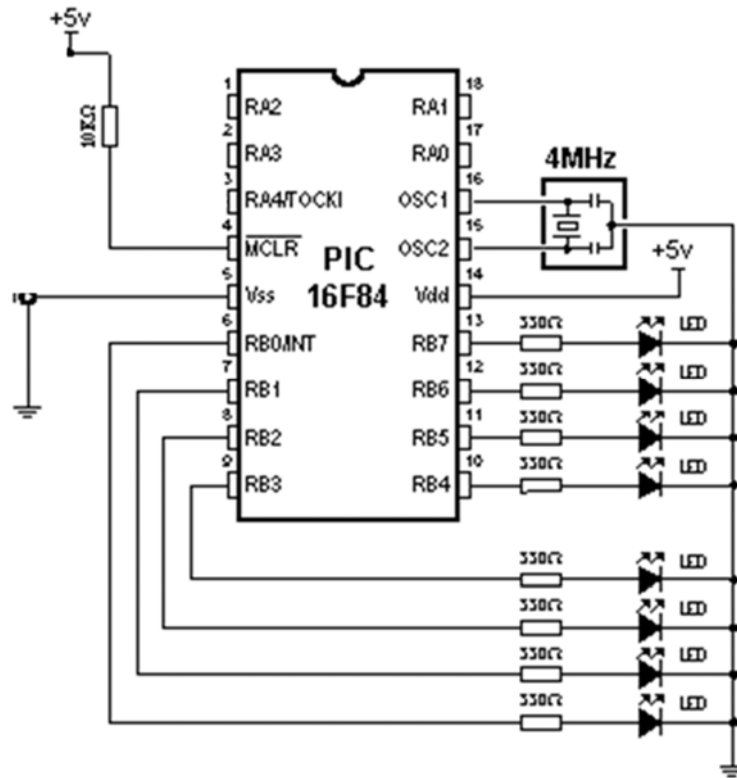


Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

- LEDs are connected to a microcontroller in two ways.
- One is to switch them on with logic zero, and other to switch them on with logic one.
- The first is called NEGATIVE logic and the other is called POSITIVE logic.



Led Flasher



v

```

.ORG 000
    GOTO start
.org 004
    GOTO start
start:
    movlw b'00000000'
    Bsf STATUS, RP0
    movwf TRISB
    Bcf STATUS, RP0

loop:
    movlw 0xff
    movwf PORTB
    call DELAY1S
    movlw 0x00
    movwf PORTB
    call DELAY1S
    goto loop
    
```

```

DELAY1S:
    MOVLW 0X05
    MOVWF 0X0E
DELAY1S_WAIT1:
    MOVLW 0XFF
    MOVWF 0X0D
DELAY1S_WAIT2:
    MOVLW 0XFF
    MOVWF 0X0C
DELAY1S_WAIT3:
    DECFSZ 0X0C,F
    GOTO DELAY1S_WAIT3
    DECFSZ 0X0D,F
    GOTO DELAY1S_WAIT2
    DECFSZ 0X0E,F
    GOTO DELAY1S_WAIT3
    RETURN
    
```

A

Push buttons

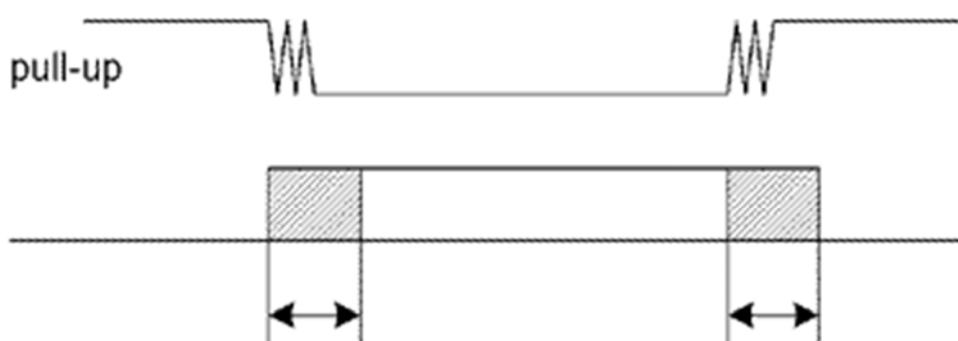
- Buttons that are used here are also called "dip-buttons".
- A pull-up resistor is needed to avoid producing short circuit between Vcc, and Gnd
- There are two possible connections as shown
 - Negative logic
 - Positive logic



9

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

- there is a short time period when vibration (oscillation) can occur as a result of unevenness of mechanical contacts, or as a result of the different speed in pushing a button (this depends on person who pushes the button).
- this phenomena is called SWITCH (CONTACT) DEBOUNCE.

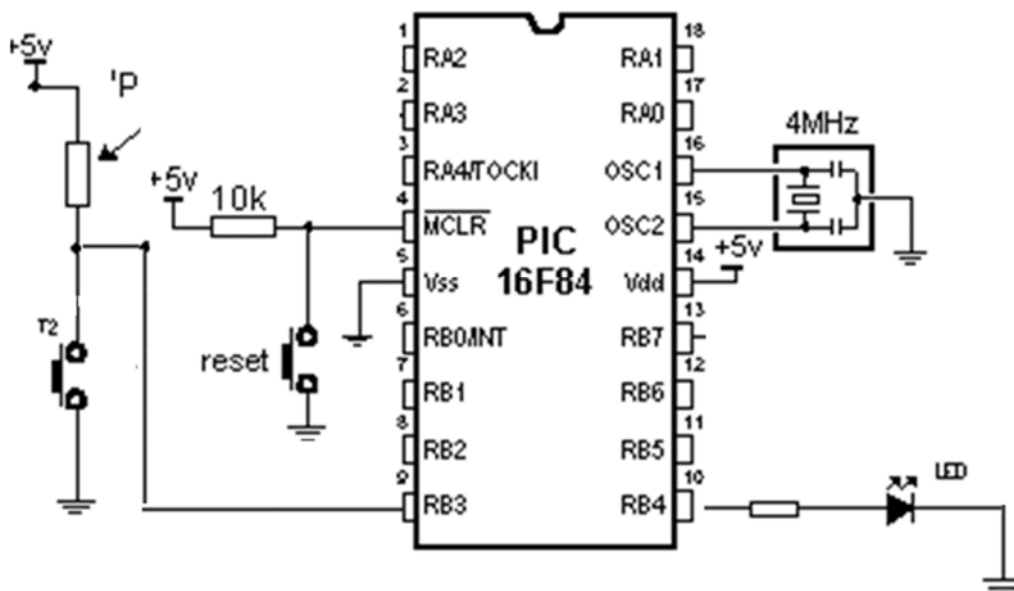


10

Button is pressed t_d t_d Button is released

- If this is overlooked when program is written, an error can occur, or the program can produce more than one output pulse for a single button push.
- to avoid this, we can introduce a small delay when we detect the closing of a contact.
- This will ensure that the push of a button is interpreted as a single pulse.
- The problem can be partially solved by adding a capacitor across the button, but a well-designed program is a much-better answer.
- The program can be adjusted until false detection is completely eliminated. Image below shows what actually happens when button is pushed.

PressControlledLed



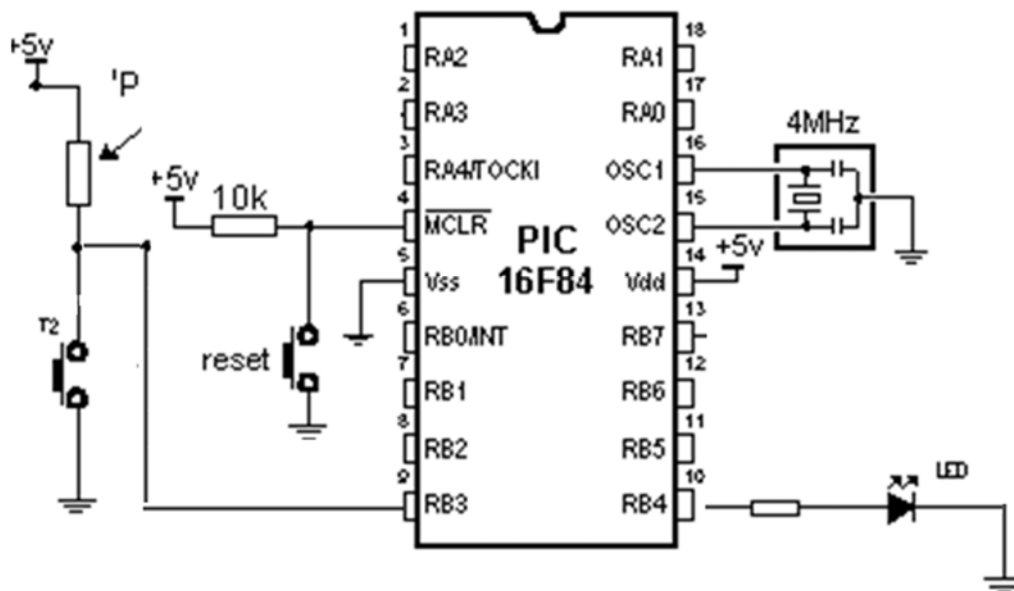
```

.ORG 000
    GOTO start
.org 004
    GOTO start
start:
    movlw b'00001111'
    Bsf STATUS, RP0
    movwf TRISB
    Bcf STATUS, RP0

loop:
    btfsc PORTB,3
    bcf PORTB,4
    btfss PORTB,3
    bsf PORTB,4
    goto loop

```

PressControlledLedFlasher



```

.ORG 000
    GOTO start
.org 004
    GOTO start
start:
    movlw b'00001111'
    Bsf STATUS, RP0
    movwf TRISB
    Bcf STATUS, RP0

loop:
    btfss PORTB,3
    goto RecheckPress
    BCF PORTB,4
    goto loop

```

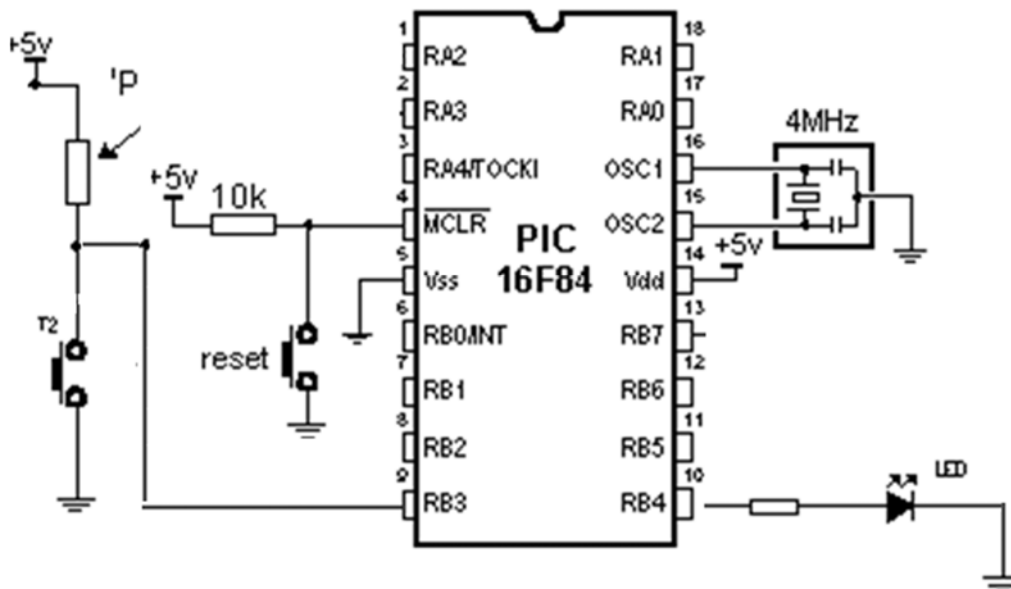
```

FLASH:
    bsf portb,4
    call DELAY20MS
    btfsc PORTB,3
    GOTO LOOP
    BCF PORTB,4
    CALL DELAY20MS
    GOTO FLASH
    btfsc PORTB,3
    GOTO LOOP

RecheckPress:
    call DELAY20MS
    btfss PORTB,3
    goto FLASH

```

PressControlledLedStatusToggle




```

.ORG 000
    GOTO start
.org 004
    GOTO start
start:
    movlw b'00001111'
    Bsf STATUS, RP0
    movwf TRISB
    Bcf STATUS, RP0
    movlw 0x00
    movwf 0x0f
    bcf portb,4

loop:
    btfss PORTB,3
    goto RecheckPress
resume_loop:
    call DELAY1S
    goto loop

```

```

RecheckPress:
    call DELAY20MS
    btfsc PORTB,3
    goto resume_loop
    btfss 0x0f,0
    goto on
    goto off

on:
    bsf 0x0f,0
    bsf portb,4
    goto resume_loop

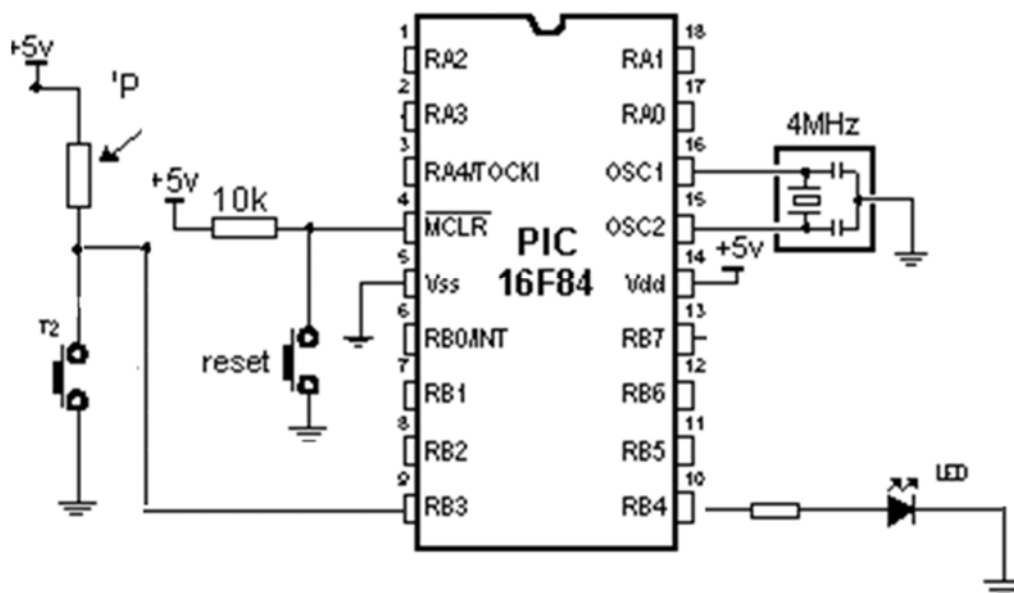
off:
    bcf 0x0f,0
    bcf portb,4
    goto resume_loop

```

17

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

PressControlledLedFlasherToggle



18

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

```

.ORG 000
    GOTO start
.org 004
    GOTO start
start:
    movlw b'00001111'
    Bsf STATUS, RP0
    movwf TRISB
    Bcf STATUS, RP0
    bcf portb,4
    movlw 0x00 ;led status
    movwf 0x0f

loop:
    btfss PORTB,3
    goto RecheckPress
resume_loop:
    btfsc 0x0f,0
    goto flash
    bcf portb,4
    goto loop

```

١٩

Dr. Ahmed ElShafee, ACU Spring 2013, Practical App. CS II

```

RecheckPress:
    call DELAY20MS
    btfsc PORTB,3
    goto resume_loop
    btfsc 0x0f,0
    goto off_mode
    btfss 0x0f,0
    goto flash_mode

off_mode:
    bcf 0x0f,0
    bcf portb,4
    call DELAY20MS
    call DELAY20MS
    goto resume_loop

flash_mode:
    bsf 0x0f,0
    bsf portb,4
    call DELAY20MS
    call DELAY20MS
    goto resume_loop

```

```

FLASH:
    bsf portb,4
    call DELAY20MS
    btfss PORTB,3
    goto RecheckPress
    BCF PORTB,4
    CALL DELAY20MS
    btfss PORTB,3
    goto RecheckPress
    GOTO FLASH

```

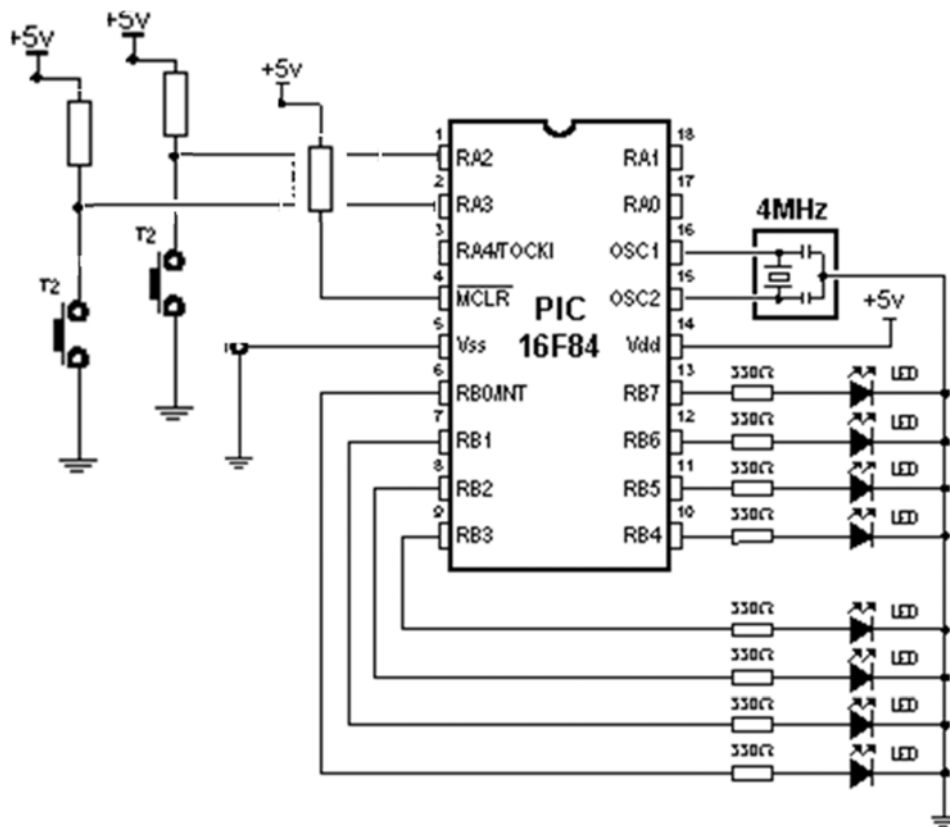
```

DELAY20MS:
    MOVLW 0XFF
    MOVWF 0X0D
DELAYMSS_WAIT1:
    MOVLW 0XFF
    MOVWF 0X0C
DELAYMSS_WAIT2:
    DECFSZ 0X0C,F
    GOTO DELAYMSS_WAIT2
    DECFSZ 0X0D,F
    GOTO DELAYMSS_WAIT1

    RETURN

```

2PressControlledLedsLevel



٢١

```

.ORG 000
    GOTO start
.org 004
    GOTO start
;portb --> 0x19
start:
    movlw b'00000000'
    Bsf STATUS, RP0
    movwf TRISB
    movlw b'00011111'
    movwf TRISA
    Bcf STATUS, RP0
    movlw 0x00
    movwf portb
;data
    call LOADDATA
    movlw 0x10
    
```

```

movwf FSR
    movf INDF,w
    movwf portb
    movwf 0x19

loop:
    btfss PORTA,2
    goto up
    btfss PORTA,3
    goto down
    goto loop

down:
    call DELAY20MS
    btfsc porta,3
    goto loop
    btfss 0x19,0
    goto loop
    decf FSR,f
    
```

٢٢

```
movf INDF,0
    movwf portb
    movwf 0x19
    goto down
up:
    call DELAY20MS
    btfsc porta,2
    goto loop
    btfsc 0x19,7
    goto loop
    incf FSR,f
    movf INDF,0
    movwf portb
    movwf 0x19
    goto up
```

LOADDATA:

```
movlw 0x00
movwf 0x10
movlw 0x01
movwf 0x11
movlw 0x03
movwf 0x12
movlw 0x07
movwf 0x13
movlw 0x1f
movwf 0x14
movlw 0x1f
movwf 0x15
movlw 0x3f
movwf 0x16
movlw 0x7f
movwf 0x17
movlw 0xff
movwf 0x18
return
```

Thanks,
See you next Week, isA