

Good Vs Bad Coding Styles

EE337

Microcontroller Pt51

Presented by : Zeal Sheth

Assembly Language Code: Bad Coding Style

- Main program is Wall of Code
- Non Modular Code
- Poor Indentation
- Repetition of bunch of code lines
- Overwriting of register values
- No Comments

Assembly Language Code: Bad Coding Style

```
;-----  
;TITLE: BAD CODING STYLES TO  
BLINK LEDs  
;AUTHOR : ZEAL SHETH (WEL)  
;-----  
LED EQU P0
```

```
ORG 00H  
LJMP MAIN  
;-----
```

```
ORG 100H  
MAIN:
```

```
MOV LED,#00H  
BACK:MOV A,#55H  
MOV LED,A  
;-----
```

Main Program is
hugh wall of code .
Non modular.
Difficult to debug

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
MOV A,#0AAH  
MOV LED,A
```

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
SJMP BACK
```

```
FIN: SJMP FIN ;ALL DONE.  
;-----
```

Assembly Language Code: Bad Coding Style

```
;-----  
;TITLE: BAD CODING STYLES TO  
  BLINK LEDs  
;AUTHOR : ZEAL SHETH (WEL)  
;-----  
LED EQU P0
```

```
ORG 00H  
LJMP MAIN
```

```
;-----  
  
ORG 100H  
MAIN:  
  
MOV LED,#00H  
BACK:MOV A,#55H  
MOV LED,A  
;-----
```



No
Indentation
:Each new
label
corresponds
to loop

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
MOV A,#0AAH  
MOV LED,A
```

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
SJMP BACK
```

```
FIN:      SJMP      FIN ;ALL DONE.  
;-----
```

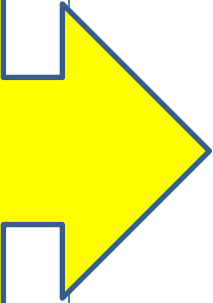
Assembly Language Code: Bad Coding Style

```
;-----  
;TITLE: BAD CODING STYLES TO  
  BLINK LEDs  
;AUTHOR : ZEAL SHETH (WEL)  
;-----  
LED EQU P0
```

```
ORG 00H  
LJMP MAIN
```

```
;-----  
  
ORG 100H  
MAIN:  
  
MOV LED,#00H  
BACK:MOV A,#55H  
MOV LED,A  
;-----
```

Repetition of
bunch of
code lines:
Can be
converted to
subroutine



```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
MOV A,#0AAH  
MOV LED,A
```

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
SJMP BACK
```

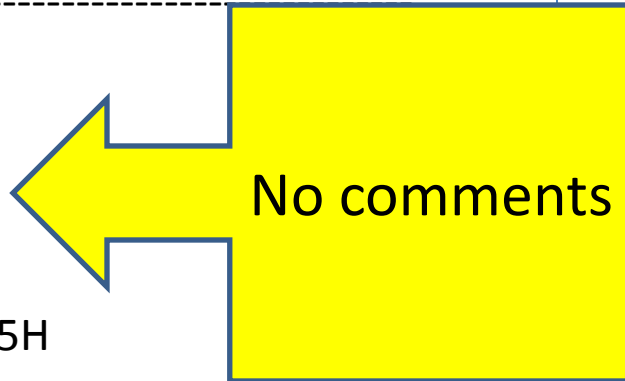
```
FIN:      SJMP      FIN ;ALL DONE.  
;-----
```

Assembly Language Code: Bad Coding Style

```
;-----  
;TITLE: BAD CODING STYLES TO  
  BLINK LEDs  
;AUTHOR : ZEAL SHETH (WEL)  
;-----  
LED EQU P0
```

```
ORG 00H  
LJMP MAIN
```

```
;-----  
  
ORG 100H  
MAIN:  
  
MOV LED,#00H  
BACK:MOV A,#55H  
MOV LED,A  
;-----
```



```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
MOV A,#0AAH  
MOV LED,A
```

```
MOV R1, #0FFH  
MOV R2, #0FFH  
DELAY1:  
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
SJMP BACK
```

```
FIN:      SJMP      FIN ;ALL DONE.  
;-----
```

Assembly Language Code: Good Coding Style

- Main program is set of function (subroutine) calls
- Modular Code : Easy Debugging
- Subroutine : For Repeated bunch of code lines
- Proper Indentation
- Stack to prevent overwriting of register values
- Appropriate Comments

Assembly Language Code: Good Coding Style

```
;-----  
;TITLE: GOOD CODING STYLES TO BLINK LED  
;AUTHOR : ZEAL SHETH (WEL)  
;-----
```

```
LED EQU P0  
ORG 00H  
LJMP MAIN
```

```
;-----  
ORG 50H
```

DELAY:

```
PUSH PSW  
PUSH AR1; STORE R1 (BANK 0)  
; ON THE STACK  
PUSH AR2
```

```
MOV R1, #0FFH  
MOV R2, #0FFH
```

Modular
Code : Easy
Debug

DELAY1:

```
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
POP AR2 ; POP MUST BE IN  
;REVERSE ORDER OF PUSH  
POP AR1  
POP PSW  
RET
```

```
;-----  
ORG 100H
```

MAIN:

```
MOV LED,#00H ;MAKE P0 AS  
;OUTPUT PORT
```

```
MOV A,#55H
```

BACK:

```
MOV LED,A  
LCALL DELAY  
CPL A  
SJMP BACK
```

FIN: SJMP FIN ;ALL DONE.

```
;-----
```


Assembly Language Code: Good Coding Style

```
;-----  
;TITLE: GOOD CODING STYLES TO BLINK LED  
;AUTHOR : ZEAL SHETH (WEL)  
;-----
```

```
LED EQU P0  
ORG 00H  
LJMP MAIN
```

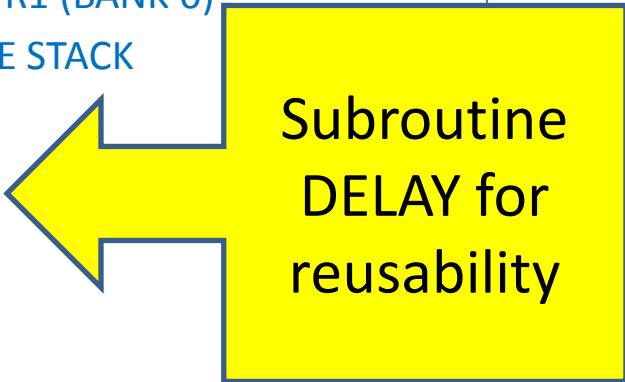
```
;-----  
ORG 50H
```

DELAY:

```
PUSH PSW  
PUSH AR1; STORE R1 (BANK 0)  
; ON THE STACK
```

```
PUSH AR2
```

```
MOV R1, #0FFH  
MOV R2, #0FFH
```



Subroutine
DELAY for
reusability

DELAY1:

```
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
POP AR2 ; POP MUST BE IN  
;REVERSE ORDER OF PUSH  
POP AR1  
POP PSW  
RET
```

```
;-----  
ORG 100H
```

MAIN:

```
MOV LED,#00H ;MAKE P0 AS  
;OUTPUT PORT
```

```
MOV A,#55H
```

BACK:

```
MOV LED,A  
LCALL DELAY  
CPL A  
SJMP BACK
```

```
FIN: SJMP FIN ;ALL DONE.
```

```
;-----
```

Assembly Language Code: Good Coding Style

```
;-----  
;TITLE: GOOD CODING STYLES TO BLINK LED  
;AUTHOR : ZEAL SHETH (WEL)  
;-----
```

```
LED EQU P0  
ORG 00H  
LJMP MAIN
```

```
;-----  
ORG 50H
```

DELAY:

```
PUSH PSW  
PUSH AR1; STORE R1 (BANK 0)  
; ON THE STACK  
PUSH AR2
```

```
MOV R1, #0FFH  
MOV R2, #0FFH
```

Proper
Indentation



DELAY1:

```
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
POP AR2 ; POP MUST BE IN  
; REVERSE ORDER OF PUSH  
POP AR1  
POP PSW  
RET
```

```
;-----  
ORG 100H
```

MAIN:

```
MOV LED, #00H ; MAKE P0 AS  
; OUTPUT PORT
```

```
MOV A, #55H
```

BACK:

```
MOV LED, A  
LCALL DELAY  
CPL A  
SJMP BACK
```

FIN: SJMP FIN ; ALL DONE.

```
;-----
```

Assembly Language Code: Good Coding Style

```
;-----  
;TITLE: GOOD CODING STYLES TO BLINK LED  
;AUTHOR : ZEAL SHETH (WEL)  
;-----
```

```
LED EQU P0  
ORG 00H  
LJMP MAIN
```

```
;-----  
ORG 50H
```

DELAY:

```
PUSH PSW  
PUSH AR1; STORE R1 (BANK 0)  
; ON THE STACK  
PUSH AR2
```

```
MOV R1, #0FFH  
MOV R2, #0FFH
```

Push and Pop
to store
register
values so that
they are not
overwritten

DELAY1:

```
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
POP AR2 ; POP MUST BE IN  
; REVERSE ORDER OF PUSH  
POP AR1  
POP PSW  
RET
```

```
;-----  
ORG 100H
```

MAIN:

```
MOV LED, #00H ; MAKE P0 AS  
; OUTPUT PORT
```

```
MOV A, #55H
```

BACK:

```
MOV LED, A  
LCALL DELAY  
CPL A  
SJMP BACK
```

```
FIN: SJMP FIN ; ALL DONE.
```

```
;-----
```

Assembly Language Code: Good Coding Style

```
;-----  
;TITLE: GOOD CODING STYLES TO BLINK LED  
;AUTHOR : ZEAL SHETH (WEL)  
;-----
```

```
LED EQU P0  
ORG 00H  
LJMP MAIN
```

```
;-----  
ORG 50H
```

DELAY:

```
PUSH PSW  
PUSH AR1; STORE R1 (BANK 0)  
; ON THE STACK
```

```
PUSH AR2
```

```
MOV R1, #0FFH  
MOV R2, #0FFH
```



Appropriate
comments

DELAY1:

```
NOP  
DJNZ R1, DELAY1  
DJNZ R2, DELAY1
```

```
POP AR2 ; POP MUST BE IN  
; REVERSE ORDER OF PUSH  
POP AR1  
POP PSW  
RET
```

```
;-----  
ORG 100H
```

MAIN:

```
MOV LED, #00H ; MAKE P0 AS  
; OUTPUT PORT
```

```
MOV A, #55H
```

BACK:

```
MOV LED, A  
LCALL DELAY  
CPL A  
SJMP BACK
```

```
FIN: SJMP FIN ; ALL DONE.
```

```
;-----
```

Thank You