

Praktisk introduktion til PIC-kredse

23. februar 2006

Mads Pedersen, OZ6HR
mads@oz6hr.dk

Plan



- Introduktion til PIC-kredse
- Familier
- Udviklingsværktøjer
- Programmeringssprog
- Programmer
- Projekter

Introduktion



- Til radioamatørformål anvendes microcontrollere ofte
- Vi har tidligere haft et teoretisk foredrag
- Manglet praktisk indgangsvinkel
- AVR introduceret og brugt, men PIC bruges stadig mange steder

PIC-familien

- Microchip
 - www.microchip.com
- PIC10, PIC12, PIC14, **PIC16**, PIC17, PIC18, PIC24
- 6, 8, 14, 18, 20, 28, 40, 44, 64, 80, 100 pins
- www.microchip.com/stellent/idcplg?IdcService=S_S_GET_PAGE&nodeId=74
- F.eks. Microchip PIC16-familien
 - www.microchip.com/ParamChartSearch/chart.aspx?branchID=1002&mid=10&lang=en&pagelD=74

PIC16C84, PIC16F84

- Features

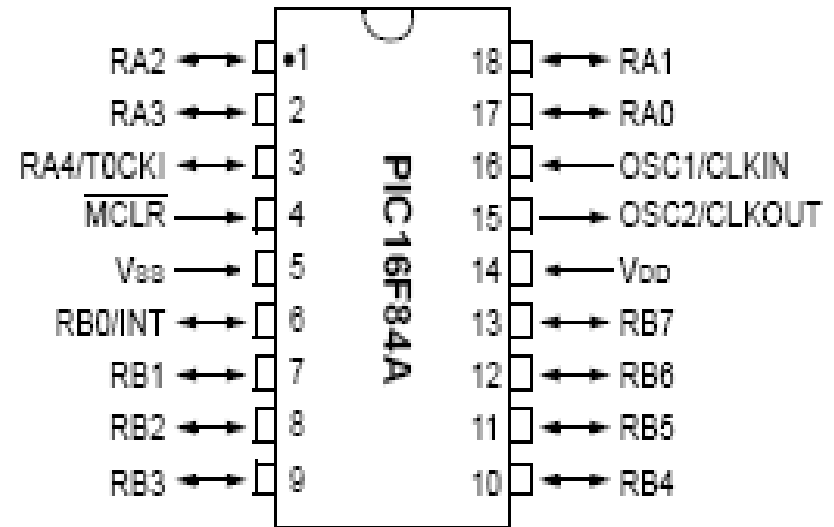
- 18 pins
- 13 I/O pins
- Program memory: 1024 bytes
- RAM: 68 bytes

- PIC16C84

- Meget brugt
- Produceres dog ikke længere

- PIC16F84

- Minder om PIC16C84
- Produceres stadig, men er på vej ud
- Forholdsvis dyr



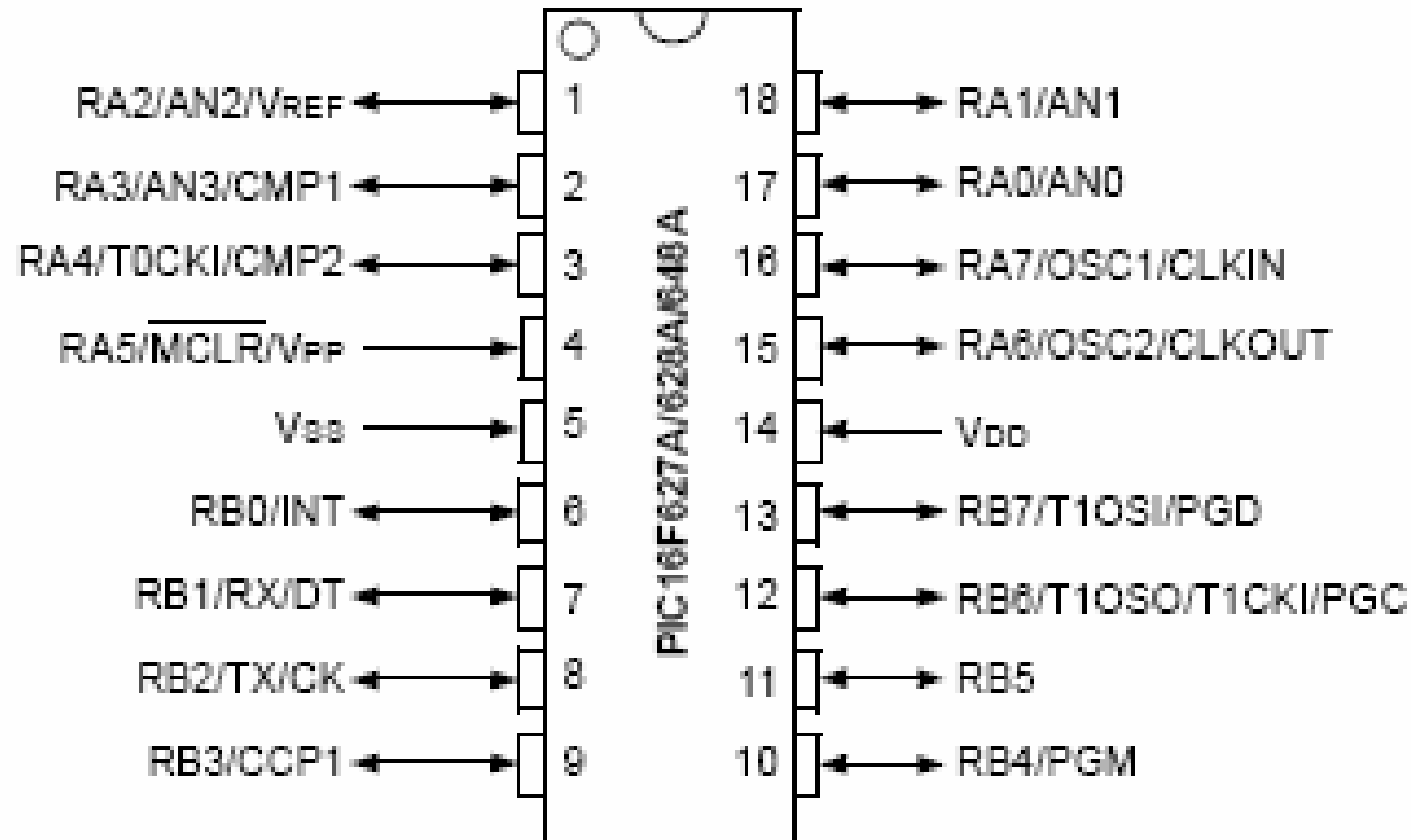
PIC16F628



- Anbefalet frem for PIC16F84
- Features:
 - 18 pins
 - 16 I/O
 - Indbygget 4 MHz oscillator
 - Program memory: 2048 bytes
 - RAM: 128 bytes
- Billigere og nyere end PIC16F84
- http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010210

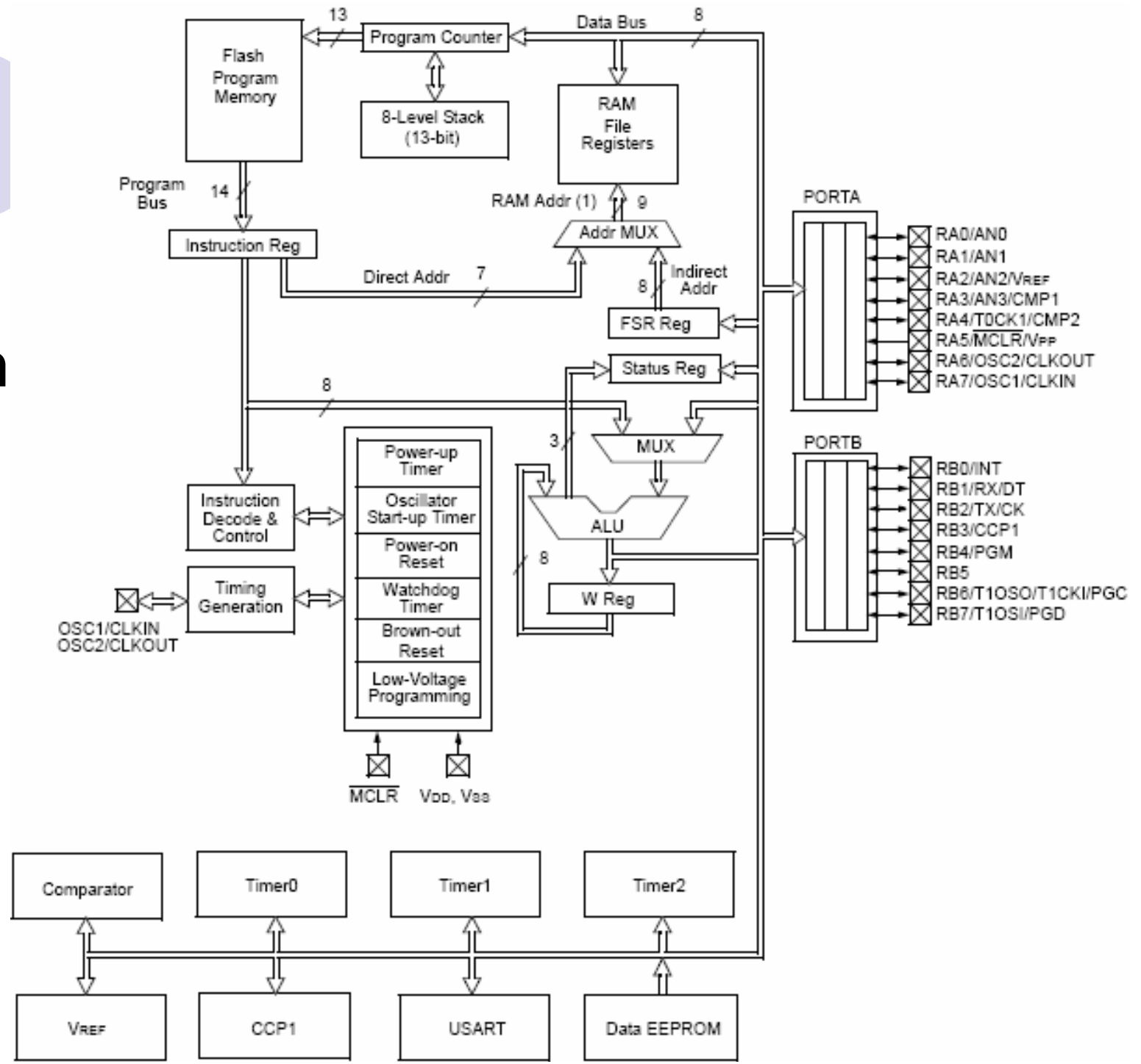
PIC16F628

- Pin-layout





● Block diagram



Udviklingsværktøjer

- Mulighed 1:

- Ren tekst (Notepad kan bruges, eller færdige programmer hentes)
- Kræver compiler (MPASMWIN)

- Mulighed 2:

- Integreret udviklingsmiljø (IDE): **MPLAB**
- Gratis (Microchip)
- Assembler og C
- Compiler, simulator, debugger
- Link:

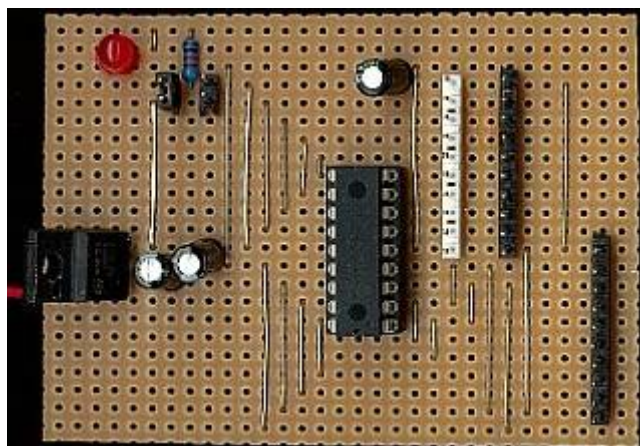
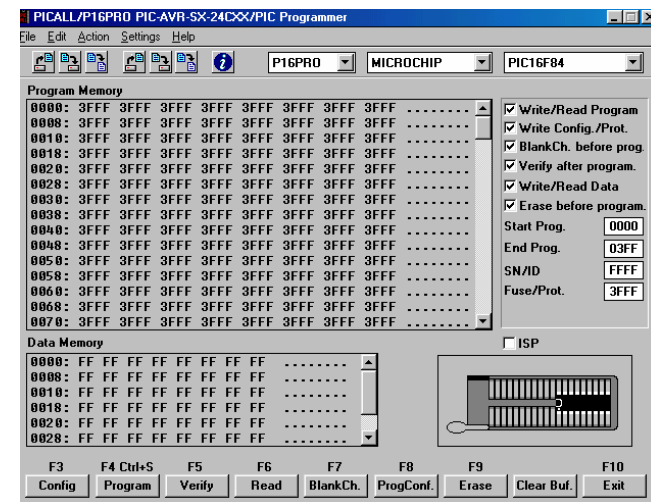
www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

Udviklingsprocessen

Kodning i MPLAB

```
your name
:-----kn_rldr.asm-----18/22/02
:shifts LEDs to the left in a knight rider-like display
:-----
list p=16F84
radix hex
:
: assign labels to registers
portb equ 0x06
shift equ 0x0c
mcount equ 0x0d
mcount equ 0x0e
:-----
: set up program
start org 0x00          ;start program at address 0x
movlw 0x00             ;load w with all 0s
tris portb             ;load all of portb to be outputs
clrf portb            ;initialize port B
movlw b'00000001'     ;loading w
movwf shift           ;move contents of w into shift
:-----
: main program
again movf shift, 0    ;move contents of shift back into w
movwf portb          ;load contents of w into portb
call pause          ;call subroutine delay
rrf shift, 1         ;rotate bits in shift one to the left
goto again          ;loop continuously
```

Programmering vha. PIC16PRO-boardet med PICALLW



Destinationsprint



Programmeringssprog

- Assembler
- C
- Basic

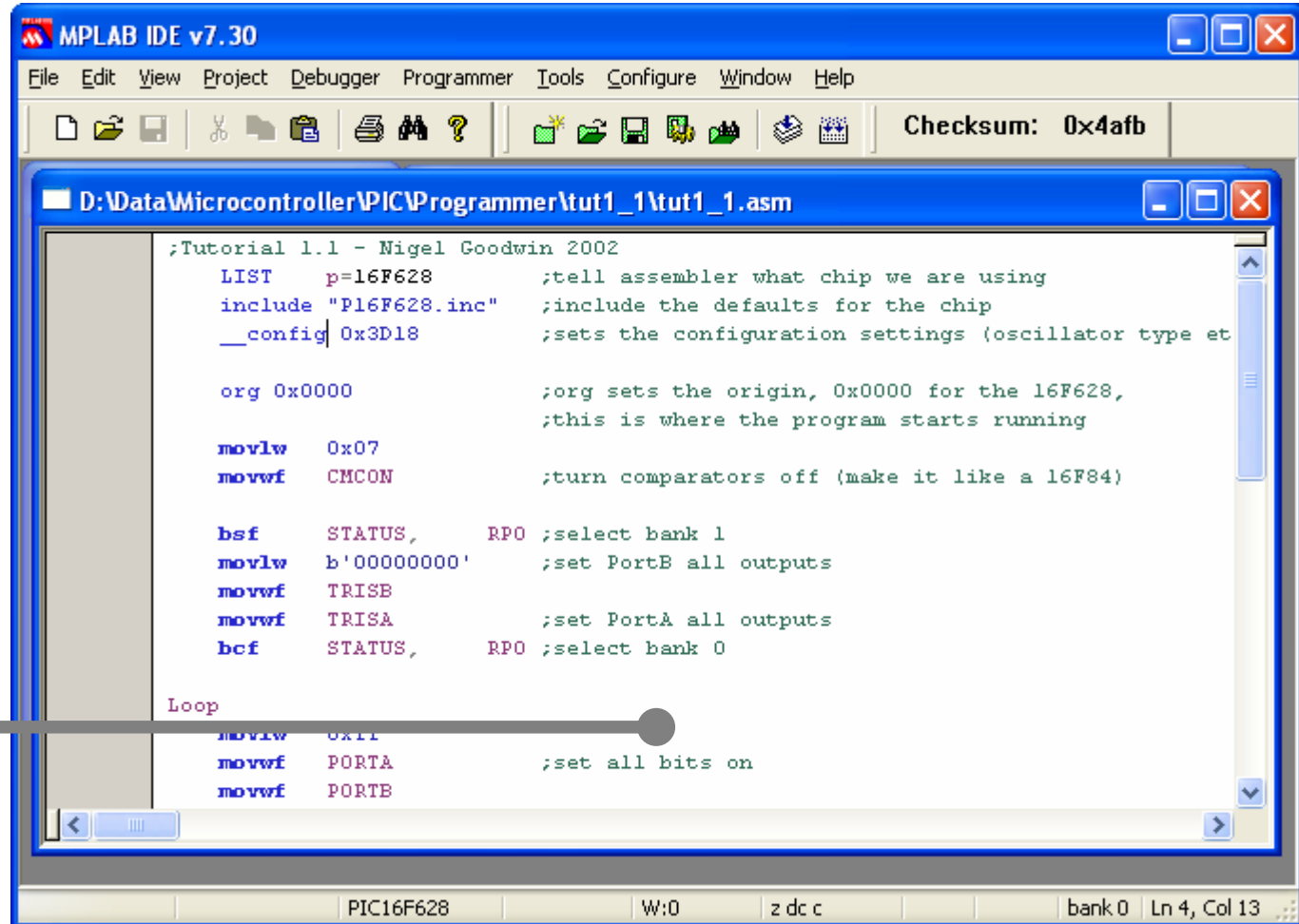
A decorative graphic at the top of the slide consists of two overlapping circles on the left and three separate circles on the right. The leftmost circle is solid light purple, and the one it overlaps is a white circle with a light purple outline. The three circles on the right are arranged horizontally: a solid light purple circle, a white circle with a light purple outline, and another solid light purple circle.

Assembler

- 35 instruktioner
- Fordele
 - Effektivt, hurtig eksekvering
- Ulemper
 - Langsom udvikling, stor fejlrisiko

1. Skriv koden

*Tekst-editor
(.asm fil)*



The screenshot shows the MPLAB IDE v7.30 interface. The main window displays an assembly code file named 'D:\Data\Microcontroller\PIC\Programmer\tut1_1\tut1_1.asm'. The code is as follows:

```
;Tutorial 1.1 - Nigel Goodwin 2002
LIST    p=16F628           ;tell assembler what chip we are using
include "P16F628.inc"     ;include the defaults for the chip
__config 0x3D18           ;sets the configuration settings (oscillator type et

org 0x0000                ;org sets the origin, 0x0000 for the 16F628,
                          ;this is where the program starts running

movlw   0x07
movwf   CMCON              ;turn comparators off (make it like a 16F84)

bsf     STATUS,           RPO ;select bank 1
movlw   b'00000000'       ;set PortB all outputs
movwf   TRISB
movwf   TRISA              ;set PortA all outputs
bcf     STATUS,           RPO ;select bank 0

Loop
movlw   0x11
movwf   PORTA              ;set all bits on
movwf   PORTB
```

The status bar at the bottom indicates the target device is PIC16F628, the window is W:0, the device is z dc c, and the current location is bank 0 Ln 4, Col 13.

35 instruk- tioner

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00	0111 dfff ffff	C,DC,Z 1, 2
ANDWF	f, d	AND W with f	1	00	0101 dfff ffff	Z 1, 2
CLRF	f	Clear f	1	00	0001 1fff ffff	Z 2
CLRW	—	Clear W	1	00	0001 0xxx xxxx	Z
COMP	f, d	Complement f	1	00	1001 dfff ffff	Z 1, 2
DECF	f, d	Decrement f	1	00	0011 dfff ffff	Z 1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1 ⁽²⁾	00	1011 dfff ffff	1, 2, 3
INCF	f, d	Increment f	1	00	1010 dfff ffff	Z 1, 2
INCFSZ	f, d	Increment f, Skip if 0	1 ⁽²⁾	00	1111 dfff ffff	1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00	0100 dfff ffff	Z 1, 2
MOVF	f, d	Move f	1	00	1000 dfff ffff	Z 1, 2
MOVWF	f	Move W to f	1	00	0000 1fff ffff	
NOP	—	No Operation	1	00	0000 0xx0 0000	
RLF	f, d	Rotate Left f through Carry	1	00	1101 dfff ffff	C 1, 2
RRF	f, d	Rotate Right f through Carry	1	00	1100 dfff ffff	C 1, 2
SUBWF	f, d	Subtract W from f	1	00	0010 dfff ffff	C,DC,Z 1, 2
SWAPF	f, d	Swap nibbles in f	1	00	1110 dfff ffff	1, 2
XORWF	f, d	Exclusive OR W with f	1	00	0110 dfff ffff	Z 1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01	00bb bfff ffff	1, 2
BSF	f, b	Bit Set f	1	01	01bb bfff ffff	1, 2
BTFSZ	f, b	Bit Test f, Skip if Clear	1 ⁽²⁾	01	10bb bfff ffff	3
BTFSZ	f, b	Bit Test f, Skip if Set	1 ⁽²⁾	01	11bb bfff ffff	3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11	111x kkkk kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11	1001 kkkk kkkk	Z
CALL	k	Call subroutine	2	10	0kkk kkkk kkkk	
CLRWDT	—	Clear Watchdog Timer	1	00	0000 0110 0100	TO,PD
GOTO	k	Go to address	2	10	1kkk kkkk kkkk	
IORLW	k	Inclusive OR literal with W	1	11	1000 kkkk kkkk	Z
MOVLW	k	Move literal to W	1	11	00xx kkkk kkkk	
RETFIE	—	Return from interrupt	2	00	0000 0000 1001	
RETLW	k	Return with literal in W	2	11	01xx kkkk kkkk	
RETURN	—	Return from Subroutine	2	00	0000 0000 1000	
SLEEP	—	Go into Standby mode	1	00	0000 0110 0011	TO,PD
SUBLW	k	Subtract W from literal	1	11	110x kkkk kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11	1010 kkkk kkkk	Z

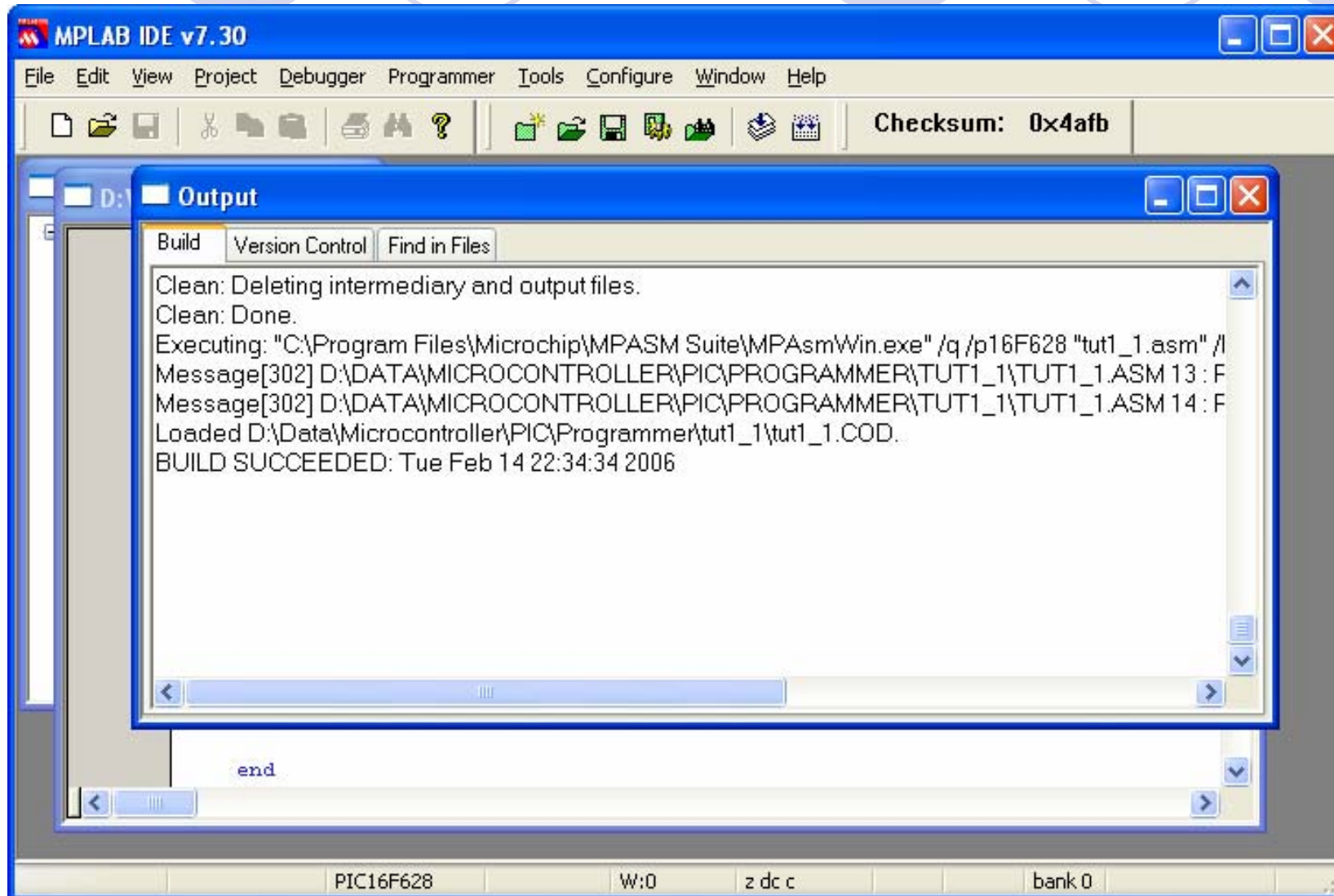
Kode-strukturen

```
c:\progra~1\mplab\pic1.asm
;NORTH
;turn the first 4 LEDs on
;=====PIC1.ASM=====10/20/02
    list    p=16f84
    radix   hex
;-----
;
;       equate port B to 0x06 register
portb  equ   0x06
;-----
;
;       org    0x000
;
;start  movlw  0x00    ;load W with 0x00
        tris   portb  ;copy W tristate into, making port B outputs
;
        movlw  0x0F    ;load W with 0x0F
        movwf  portb  ;load port B with contents of W
;
;circle goto   circle ;done
;
;       end
;-----
```



Label-felt

2. "Assemble" (kompiler) koden



Objektfil

```

_LVP_OFF          00003F7F
_LVP_ON           00003FFF
_MCLRE_OFF       00003FDF
_MCLRE_ON        00003FFF
_PWRITE_OFF     00003FFF
_PWRITE_ON      00003FF7
_WDT_OFF        00003FFB
_WDT_ON         00003FFF
_XT_OSC         00003FED
__16F628        00000001
MPASM 5.01          TUT1_1.ASM  2-14-2006  22:34:34      PAGE 5

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : XXXXXXXXXXXXXXXXXXXX -----
2000 : -X-----

All other memory blocks unused.

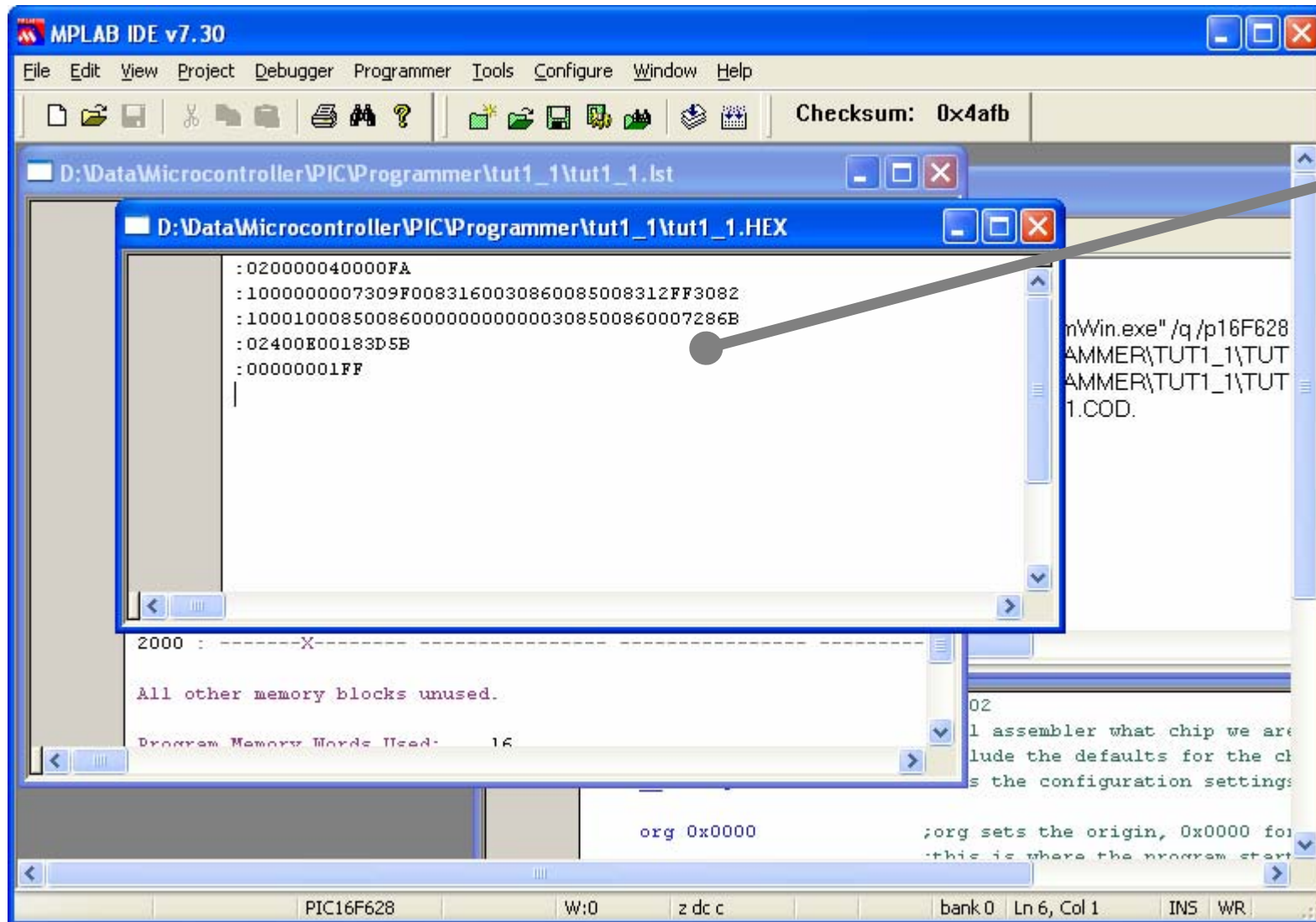
Program Memory Words Used:    16
Program Memory Words Free:  2032

Errors      : 0
Warnings   : 0 reported,    0 suppressed
Messages   : 2 reported,    0 suppressed

```

.lst (list) fil

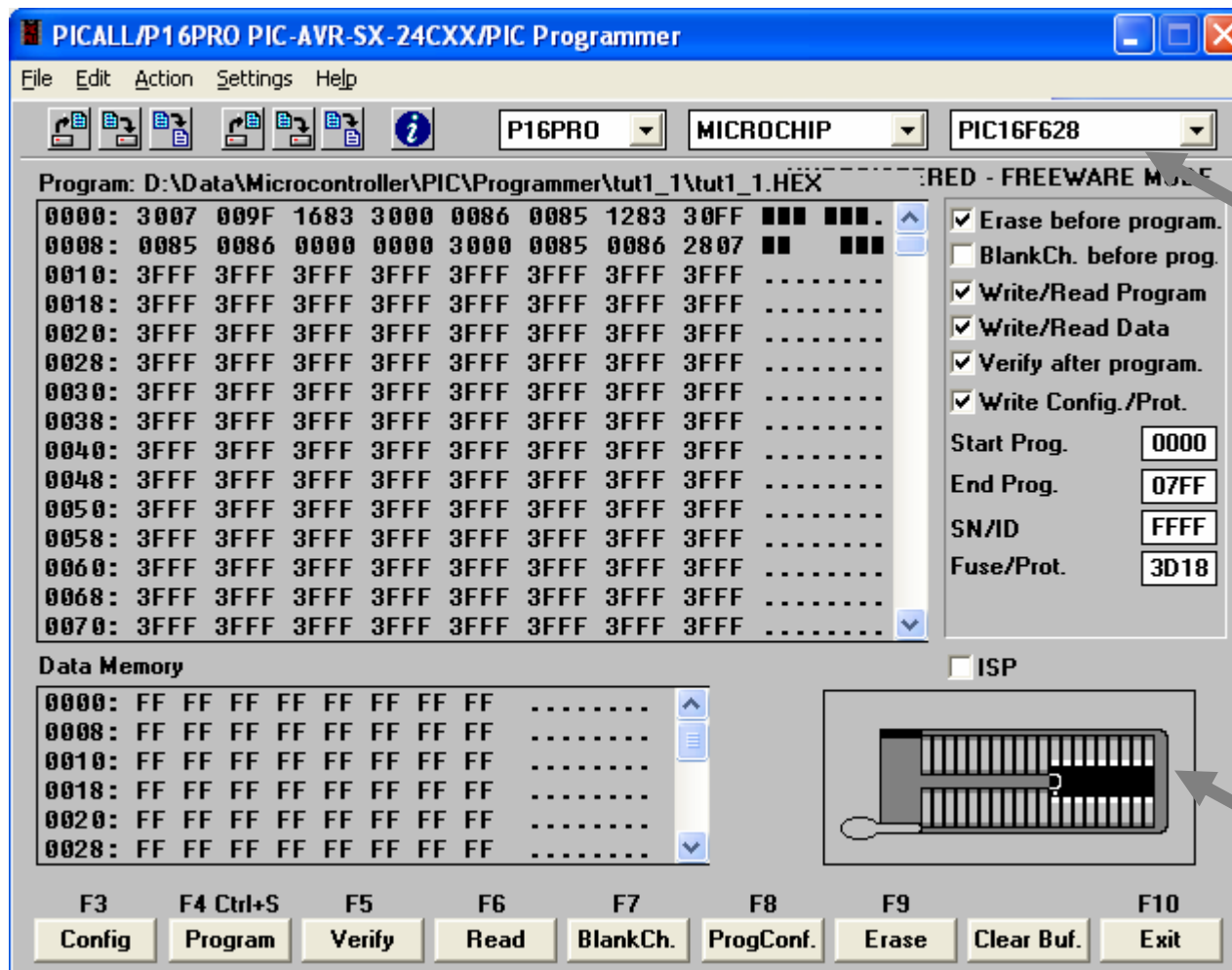
Hex-fil (som senere overføres til kredsen)



.hex fil

3. Download til kredsen

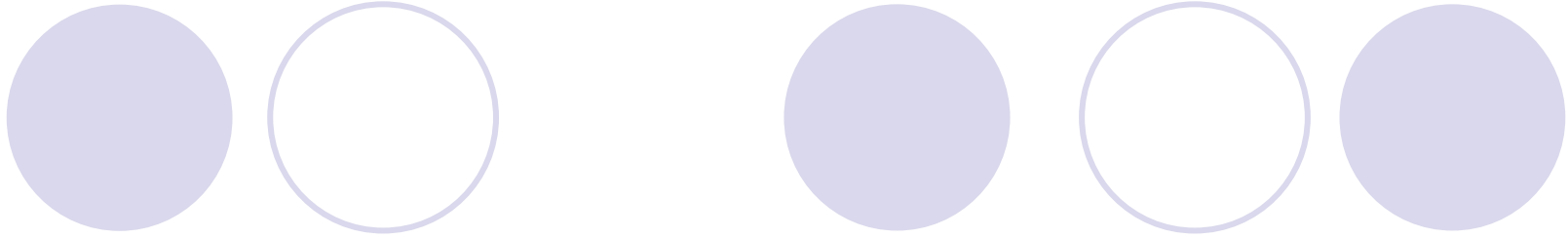
- Vha. PIC16PRO-boardet og PICALLW-softwaren



Valgt PIC-kreds

Hvor den skal sidde

C



- **Fordele**

- Hurtigere udvikling, mere overblik,
kan struktureres mere logisk

- **Ulempe**

- Langsommere eksekvering

Eksempel på C-kode

```
#include <pic.h>

#define PORTBIT(adr, bit) ((unsigned)(&adr)*8+(bit))

static bit button @ PORTBIT(PORTA, 1);

main(void)
{
    unsigned i;
    unsigned char j;

    TRISB = 0; /* all bits output */
    j = 0;
    for(;;)
    {
        PORTB = 0x00; /* turn all on */
        for (i = 16000 ; --i ;)
            continue;
        PORTB = ~j; /* output value of j */
        for (i = 16000 ; --i ;)
            continue;
        if (button == 0) /* if switch pressed, increment */
            j++;
    }
}
```



Basic

- Programming af PIC i BASIC-sproget

- <http://www.mikroelektronika.co.yu/english/product/books/picbasicbook/00.htm>

Programmer

A decorative graphic at the top of the slide consists of two rows of circles. The top row has a solid light purple circle on the left and an outlined light purple circle on the right. The bottom row has a solid light purple circle on the left, an outlined light purple circle in the middle, and a solid light purple circle on the right.

- Mange muligheder

- OZ-programmer

- Fordele: Simpel, billig
 - Ulemper: Afhængig af PC-hastighed,
kun PIC16C84/PIC16F84

- PIC16PRO

- Fordele: Alsidig
 - Ulemper: Mere kompleks
 - Klubben har en liggende til medlemmernes brug

PIC16PRO



- Print og schematics

- www.picallw.com/pdf/p16pro.pdf

- Software (PICALLW)

- <http://picallw.com/zip/paw016.zip>



Projekter

- PIC Tutorial

- http://www.winpicprog.co.uk/pic_tutorial.htm

- Main board
 - LED board
 - LCD board
 - Switch board
 - ADC board
 - + mange flere

Andre nyttige links

- Alt om microcontrollere
 - www.microcontroller.com
- Demo-Tech (demo-boards mv.)
 - www.demotech.dk
- OZ6HR's projektside
 - www.oz6hr.dk

Demonstration!

- MPLAB
 - Nyt projekt
 - Ny .asm-fil, kopier kode fra PIC Tutorial
 - Kompilering (Build All)
- Overførsel til PIC-kredsen vha. PIC16PRO-boardet og PICALLW-programmet
- Demo på "PIC Tutorial Main Board"
- Simulation i MPLAB
- Minimal udvikling
 - .asm-fil og MPASMWIN