#### **PICmicro Information**

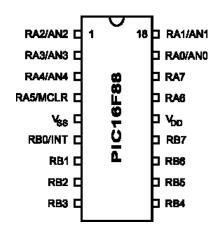
The PICmicro programs include 'equate' statements that define the following labels:

Register	Description			
PORTA	input / output port A			
PORTB	input / output port B			
TRISA the control register for port A *				
TRISB	the control register for port B *			
STATUS the status register				
INTCON	the interrupt control register			

<sup>\*</sup> these registers are on memory page 1

Constant	Description					
W	result into working register (h'00')					
F result into file register (h'01')						
RP0 register page selection bit						
С	the carry flag STATUS bit					
Z	the zero flag STATUS bit					
GIE	the global interrupt controller bit					
INT0IE	the external interrupt enable bit					
INTOIF external interrupt occurred flag bit						

## Pinout for 16F88 PICmicro IC:



# List of commands:

Mnemonic	Operands	Description				
addlw	k	Add working register to literal k (k + WREG)				
andlw	k	AND working register with literal k (k & WREG)				
bcf	f, b	Bit clear in file register (file register f, bit b)				
bsf	f, b	Bit set in file register (file register f, bit b)				
btfsc	f, b	Bit test in file register, skip if clear (file register f, bit b)				
btfss	f, b	Bit test in file register, skip if set (file register f, bit b)				
call	label	Call subroutine at label				
clrf	f	Clear file register f				
comf	f, d	Complement f (to itself if d = 1, or working register if d = 0)				
decfsz	f, d	Decrement f, skip if zero (to itself if d = 1, or working register if d = 0)				
goto	label	Unconditional branch to label				
incf	f, d	Increment file register f (to itself if d = 1, or working register if d = 0)				
iorlw	k	Inclusive OR working register with literal (k   WREG)				
movf	f, d	Move f (to itself if d = 1, or working register if d = 0)				
movlw	k	Move literal to working register				
movwf	f	Move working register to file register f				
nop	-	No operation				
retfie	-	Return from interrupt service routine and set global interrupt enable bit GIE				
return	-	Return from subroutine				
sublw	k	Subtract working register from literal k (k – WREG)				

## **Number system notation**

Decimal	d'153'
Hex	h'20' or 0x20
Binary	b'10010110' or 0b10010110

## Structure of the INTCON register

Ī	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Ī	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

## Structure of the STATUS register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRP	RP1	RP0	TO	$\overline{ ext{PD}}$	Z	DC	С

#### **Extension PICAXE material for project work (not part of syllabus)**

#### **Instruction Set**

The full Microchip MPASM instruction set is supported.

# Pre-defined general purpose registers

The following byte registers are available for general use - B0 to B27. They are assumed to be on memory page 0. Registers and constants can be renamed using EQU e.g.

```
wSave EQU B20
```

#### Assumed sub-routines for project work.

For PICAXE project work use the following sub-routines are assumed predefined to make 'real-life' project work easier.

```
call wait1ms
               ; delay 1 millisecond
call wait10ms
                ; delay 10 milliseconds
call wait100ms
                ; delay 100 milliseconds
call wait1000ms ; delay 1000 milliseconds
               ; read analogue value on A.O and copy into bO
call readadc0
call readadc1
                ; read analogue value on A.1 and copy into b1
                ; read analogue value on A.2 and copy into b2
call readadc2
                ; read DS18B20 temperature value on A.O and copy into bO
call readtemp0
call readtemp1
                ; read DS18B20 temperature value on A.1 and copy into b1
                ; read DS18B20 temperature value on A.2 and copy into b2
call readtemp2
                ; upload byte register (B0-B27) values to computer screen
call debug
call lcd
                ; send byte value in working register out of pin B.O to
                 ; LCD at 2400, N, 8, 1
```