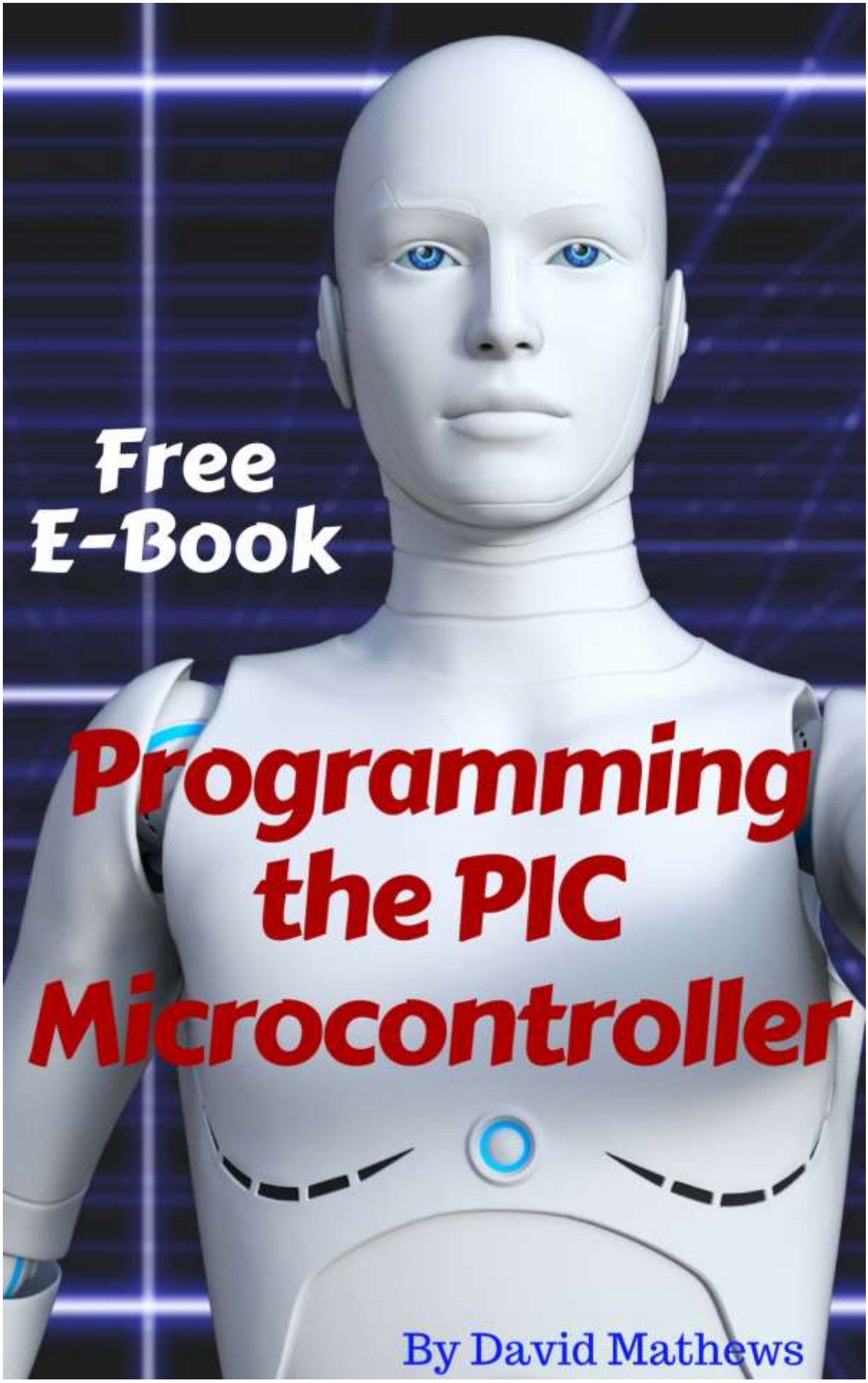
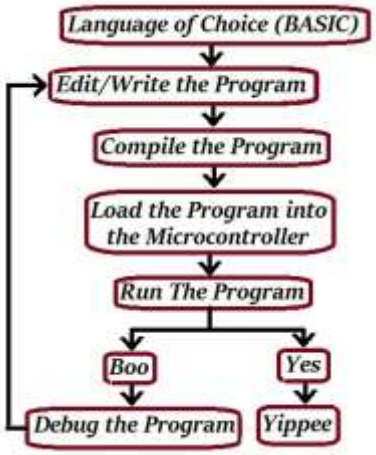
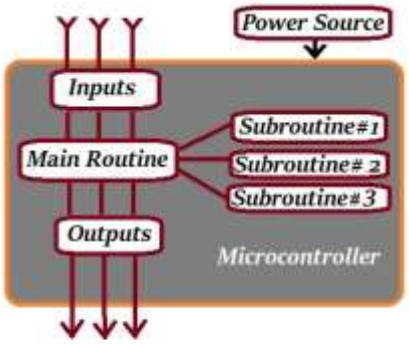
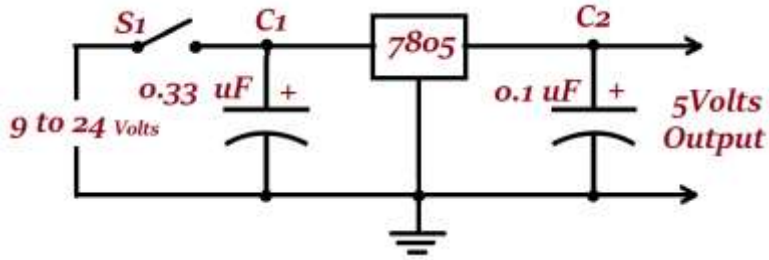

Free
E-Book
Programming
the PIC
Microcontroller
By David Mathews

Programming the PIC microcontroller





+5 volts is needed to power the PIC. A regulator circuit is required if you are using higher voltage adapters.



The most popular programming languages for the PIC are Basic and C although C++, assembly language and others are possible. The instruction set for assembly language programming of the PIC is predetermined by the chip's RISC (reduced instruction set computing) design. The language only has 35 commands. They are divided into five separate categories depending on their type of operation. I've listed two of the five categories although most people program in one of the higher level languages.

Data Transfer Instructions

This classification includes instructions that can be used to change bit value (Location) or move data bytes.

MOVLW

This instruction is used to write a constant Literal to the W register.

Purpose: The purpose of the operation is to move eight bit constant k to register W.

Operand: k is a constant greater than zero but less than 256

Memory needed: 1 word

Machine Cycles: 1

Flags changed: 0

Syntax: Label MOVLW k

MOVWF

This instruction is used to move data bits from the W register to the Flag register.

Purpose: The purpose of the operation is to copy the content of the W register into the flag register.

Syntax: Label MOVWF f

Operation: W to (f) (W register to Flag register)

Operand: 0 < f < 127

Memory needed: 1 word

Machine cycles: 1

Flags required: none

MOVF

This instruction copies the contents of the flag register to the destination register.

Purpose: The purpose of the operation is to copy the content of the flag register to either the W or F register, depending on the status of the D register.

Syntax: Label MOVF W, f

if d=0, the destination is register W

if d=1, the destination is register f

Operation: (if Status D=0) f to W, (if Status D=1) f to D

Operand: 0 < k < 127

Memory needed: 1

Machine Cycles: 1

Flag: Z

CLRW

This is a clearing instruction that reset the values of W register to '0'

Purpose: W register reset to zero and status register Z flag is set to one

Z flag in status register is set to one

Syntax: Label CLRW

Operation: 0 to (W)

Operand: nil

Memory needed: 1

Machine Cycles: 1

Flag: Z

CLRF f

This is a clearing instruction that writes a zero in the f register

Purpose: writes a zero in the f register & sets the status register Z flag to a one

Z flag in status register is set to one

Syntax: Label CLRF f

Operation: 0 to (f)

Operand: nil

Memory Needed: 1

Machine Cycles: 1

Flags: Z

SWAPF

This instruction is used to swap the upper and lower nibbles (4bits) of either the W the F register.

Purpose: To exchange Upper, Lower nibbles of the destination register

Operation: f (0:3) to d(4:7) and f(4:7) to d(0:3)

Operand: 0 < f < 127

No. of words: 1

No. cycles: 1

Flags: —

Arithmetic and Logic Operations Group

Instructions for the PIC micro controller. The Arithmetic and logic operations of the micro controller (PIC) perform all math and logic internal functions. The mathematical operations performed include are addition, subtraction, multiplication, and division. The Logic operations performed include AND, OR, NOT, and XOR.

ADDLW

This instruction performs the addition operation by adding a constant to the content of the W register.

Purpose: To add a given constant to the content of the W reg.

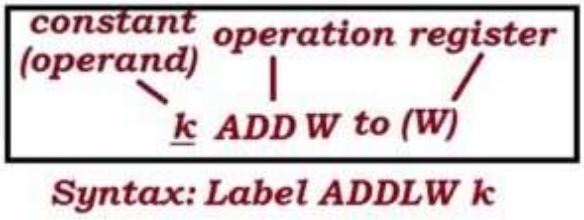
Operation: (w) + k to w

Operand: 0 < k < 255

Memory needed: 1

Machine Cycles: 1

Flags: C, DC, Z



ADDWF

This instruction adds the contents of the f register with that of the W register. The result is stored in the W register if 'd' is zero. The result is stored in the f register if 'd' is one.

Purpose: Add the W register content to the f register content and place the results in W or f.

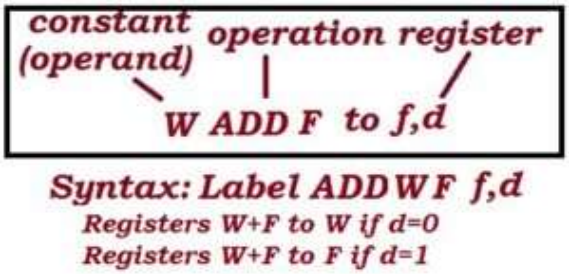
Operation: (w) + (f) to w if d = 0 and (w) + (f) to f if d = 1

Operand: 0 < f < 127

Memory needed: 1

Machine cycles: 1

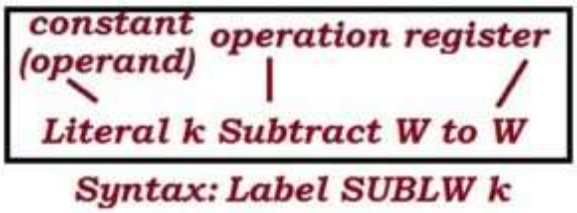
Flags affected: C, DC, Z



SUBLW

This instruction subtracts the contents of the W register) from the eight-bit literal 'k'. The result is placed in the W register

Purpose: Subtract W from Literal 'k' and place the results in the W register.



Operation: k - (W) ® (W)

Operand: 1< k < 255

Memory needed: 1

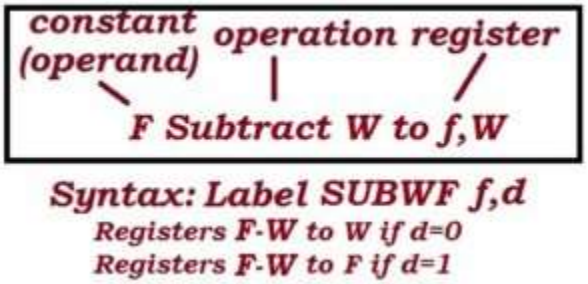
Machine cycles: 1

Flags affected: C, DC, Z

SUBWF

This instruction Subtracts W content from f register.

Purpose: W register content is subtracted from that of register f



Operation: f – (w) to w if d = 0 and f – (w) to f if d = 1

Operand: 0 < f < 127

Memory required: 1

Machine cycles: 1

Flags affected: C, DC, Z

ANDLW

This logical instruction is used to perform a Logic AND function with a given constant and the content of the W register.

Purpose: Given constant is .and. with W reg.

Operation: (w) .and. k to w

Operand: 0 < k < 255

Memory required: 1

Machine cycles: 1

Flags affected: Z

Here are a few more terms you may need clarified:

IDE--- Integrated Development Environment is a software program that runs on a PC to develop applications for a microcontroller.

Assembler—a program for converting instructions written in low-level symbolic code (assembly language) into machine code (ones and zeros).

Compiler---a program used to convert high level programming language into machine-code so that it can be understood by a computer.

Mnemonic ---in assembly language, it is a group of letters used as a command. They make it easier for humans to remember each command rather than a bunch of ones and zeros.

Some of the websites that will provide great information on PIC programming in Basic and C are listed here:

- https://www.markhennessy.co.uk/pic/start.htm
- http://www.microcontrollerboard.com/pic_microcontroller.html
- https://www.elprocus.com/pic-microcontroller-programming-using-c-language/
- http://embedded-lab.com/blog/embedded-lab-experiments/
- https://www.mikroe.com/blog/50-great-tutorials-and-projects-for-mikroc-mikrobasic-and-mikropascal
- https://circuitdigest.com/microcontroller-projects/writing-your-first-pic-microcontroller-program

Here is the "standard" layout for a PIC program:

- Introductory Description of your program, resources, & restriction
- Variable declarations (named for programming ease)
- Setup of I/O and other pins requiring initialization
- Loop/Timing/Calculation operations

I hope that the information in this E-Book is useful. You should sign up to receive my monthly newsletter. Every Month I am highlighting a device that may be of significant interest to inventers. Leave me a message if you have a chip or sensor of special interest to you.

Also, I'm looking for people to become Patreon members, so I can spend more time creating educational content. Even at the $1 a month level it would be greatly appreciated. I've offered escalating rewards for each level of participation. Hopefully I will be able to offer more informative free E-Books to my patrons.

http://www.patreon.com/roboticsup/