College of Science and Engineering

**Embedded Systems Engineering**

**(6EJ540)**

**Embedded Systems Technology**

**(6EJ541)**

**PIC18F25K22**

**Derbot**

**C Summary Manual**

(All diagrams copyright Microchip)

**Module Leader: David Wilson**

[**D.Wilson@Derby.ac.uk**](mailto:D.Wilson@Derby.ac.uk)

**MS226**

**Ext 3202**



# PIC18F25K22 Block Diagram

PIC18F25K22 block dagram taken from the datasheet.

Figure 1 PIC18F25K22 Block Diagram

# PIC18F25K22 Pin Diagram

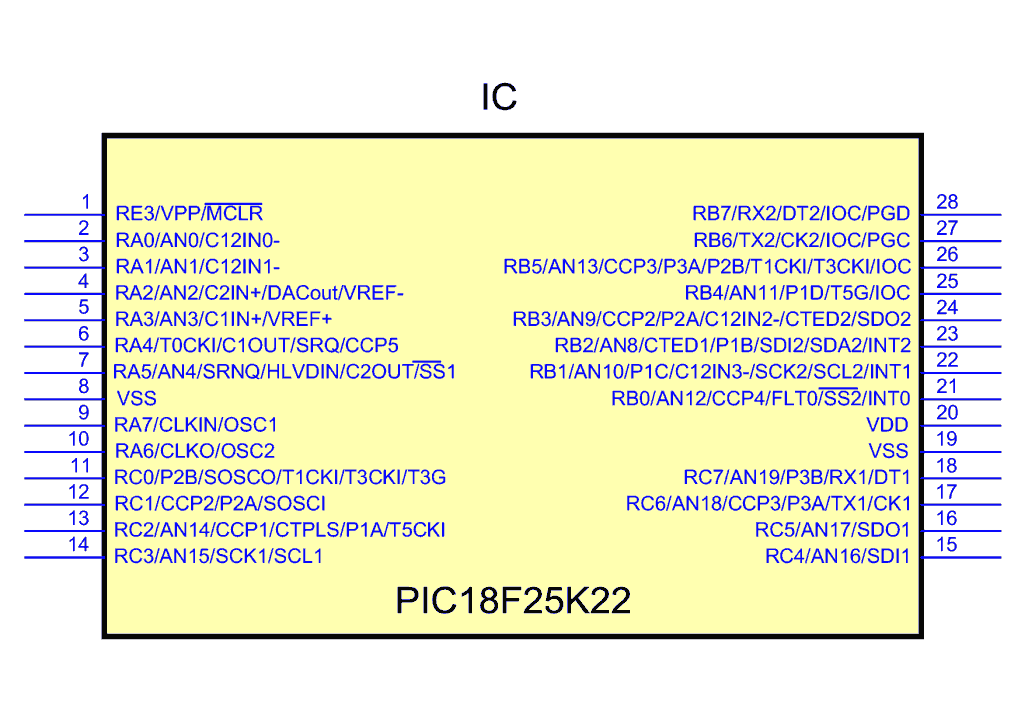


Figure 2 18F25K22 Pin Diagram

# Configuration Bits Summary of function

|  |  |
| --- | --- |
| Bit Name | Function |
| IESO | Internal/External Oscillator Switchover bit |
| FCMEN | Fail-Safe Clock Monitor Enable bit |
| FOSC3:FOSC0 | Selects one of 12 oscillator modes |
| BORV1, BORV0 | Selects Brown-out Reset voltage |
| BOREN1, BOREN0 | Selects Brown-out mode |
| PWRTEN | Power-up Timer Enable bit |
| WDTPS3:WDTPS0 | Watchdog Timer Postscale bits |
| WDTEN | Watchdog Timer Enable bit |
| MCLRE | MCLR Pin Enable bit |
| LPT1OSC | Low-Power Timer1 Oscillator Enable bit |
| PBADEN | PORTB A/D Enable bit |
| CCP2MX | CCP2 Multiplex, selects RC1 (1) or RB3 (0) |
| DEBUG | Background Debug Enable bit |
| XINST | Extended Instruction Set Enable bit |
| LVP | Low-Voltage Program Enable bit |
| STVREN | Stack Full/Underflow Reset Enable bit |
| CP3:CP0 | Code Protection bits |
| CPD | Data EEPROM Code Protection bit |
| CPB | Boot Block Code Protection bit |
| WRT3:WRT0 | Program Memory Write Protection bits |
| WRTD | Data EEPROM Write Protection bit |
| WRTB | Boot Block Write Protection bit |
| WRTC | Configuration Register Write Protection bit |
| EBTR3:EBTR0 | Table Read Protection bits |
| EBTRB | Boot Block Table Read Protection bit |
| DEV2:DEV0 | Device ID bits: 010 = 18F2420, 100 = 18F4520 |
| REV4:REV0 | Revision ID bits |
| DEV10:DEV3 | Further Device ID bits |

Figure 3 PIC18F2420 configuration bits and device identifications

# PIC18F25K22 Special Function Registers

PIC18F25K22 Special Function Registers taken from the datasheet.

Figure 4 PIC18F25K22 Special Function Registers

# Derbot Full Schematic

Diagram showing the full Derbot schematic.

Figure 5 Derbot Full Build

This illustrates some options, it is not a target build.

# Component Layout

Derbot component layout



Figure 6 Component Layout – Viewed from Component Side, Version 6.7

# Appendix A C summary

This appendix provides summary information on key aspects of the C programming language, as a set of tables.

# C keywords associated with data type and structure definition

|  |  |  |  |
| --- | --- | --- | --- |
| Word | Summary meaning | Word | Summary meaning |
| char | A single character, usually 8-bit | signed | A qualifier applied to char or int (default for char and int is signed) |
| const | Data that will not be modified | sizeof | Returns the size in bytes of a specified item, which may be variable, expression or array |
| double | A ‘double precision’ floating-point number | struct | Allows definition of a data structure |
| enum | Defines variables that can only take certain integer values | typedef | Creates new name for existing data type |
| float | A ‘single precision’ floating-point number | union | A memory block shared by two or more variables, of any data type |
| int | An integer value | unsigned | A qualifier applied to char or int (default for char and int is signed) |
| long | An extended integer value; if used alone, integer is implied | void | No value or type |
| short | A short integer value; if used alone, integer is implied | volatile | A variable which can be changed by factors other than the program code |

# C keywords associated with program flow

|  |  |  |  |
| --- | --- | --- | --- |
| Word | Summary meaning | Word | Summary meaning |
| break | Causes exit from a loop | for | Defines a repeated loop – loop is executed as long as condition associated with for remains true |
| case | Identifies options for selection within a switch expression | goto | Program execution moves to labelled statement |
| continue | Allows a program to skip to the end of a for, while or do statement | if | Starts conditional statement; if condition is true, associated statement is executed |
| default | Identifies default option in a switch expression, if no matches found | return | Returns program execution to calling routine, causing also return of any data value specified by function |
| do | Used with while to create loop, in which statement following do is repeated as long as while condition is true | switch | Used with case to allow selection of a number of alternatives; switch has an associated expression which is tested against a number of case options |
| else | Used with if, and precedes alternative statement used if if condition is not true | while | Defines a repeated loop – loop is executed as long as condition associated with while remains true |

# C keywords associated with data storage class

|  |  |  |  |
| --- | --- | --- | --- |
| Word | Summary meaning | Word | Summary meaning |
| auto | Variable exists only within block within which it is defined. This is the default class | register | Variable to be stored in a CPU register; thus, address operator (&) has no effect |
| extern | Declares data defined elsewhere | static | Declares variable which exists throughout pro­gram execution; the location of its declaration affects in what part of the program it can be referenced |

# C data types, as implemented by the MPLAB XC8 C compiler

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Description | Length (bytes) | Range |
| char | Character | 1 | -128 to +127 |
| signed char | Character | 1 | -128 to +127 |
| unsigned char | Character | 1 | 0 to 255 |
| int | Integer | 2 | -32,768 to +32,767 |
| unsigned int | Integer | 2 | 0 to 65,535 |
| Short | Integer | 2 | -32,768 to +32,767 |
| unsigned short | Integer | 2 | 0 to 65,535 |
| short long | Integer | 3 | -8,388,608 to 8,388,607 |
| unsigned short long | Integer | 3 | 0 to 16,777,215 |
| long | Integer | 4 | -2,147,483,648 to 2,147,483,647 |
| unsigned long | Integer | 4 | 0 to 4,294,967,295 |
| float | Floating point | 4 | From 1.17549 ×10-38 to 6.80565 × 10+38 |
| double | Floating point, double precision | 4 | From 1.17549 × 10-38 to 6.80565 × 10+38 |

# C operators

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prec. and order** | **Operation** | **Symbol** | **Prec. and order** | **Operation** | **Symbol** |
| **Parethenses and array access operators** | | | | | |
| 1, L to R | Function calls | ( ) | 1, L to R | Point at member | X–>Y |
| 1, L to R | Subscript | [ ] | 1, L to R | Select member | X.Y |
| **Arithmetic operators** | | | | | |
| 4, L to R | Add | X+Y | 3, L to R | Multiply | X\*Y |
| 4, L to R | Subtract | X–Y | 3, L to R | Divide | X/Y |
| 2, R to L | Unary plus | +X | 3, L to R | Modulus | % |
| 2, R to L | Unary minus | –X |  |  |  |
| **Relational operators** | | | | | |
| 6, L to R | Greater than | X>Y | 6, L to R | Less than or equal to | X<=Y |
| 6, L to R | Greater than or equal to | X>=Y | 7, L to R | Equal to | X= =Y |
| 6, L to R | Less than | X<Y | 7, L to R | Not equal to | X!=Y |
| **Logical operators** | | | | | |
| 11, L to R | AND (1 if both X and Y are not 0) | X&&Y | 2, R to L | NOT (1 if X=0) | !X |
| 12, L to R | OR (1 if either X or Y are not 0) | X||Y |  |  |  |
| **Bitwise operators** | | | | | |
| 8, L to R | Bitwise AND | X&Y | 2, L to R | Ones complement (bitwise NOT) | ≈X |
| 10, L to R | Bitwise OR | X|Y | 5, L to R | Right shift. X is shifted right Y times | X »Y |
| 9, L to R | Bitwise XOR | XˆY | 5, L to R | Left shift. X is shifted left Y times | X Y |
| **Assignment operators** | | | | | |
| 14, R to L | Assignment | X=Y | 14, R to L | Bitwise AND assign | X&=Y |
| 14, R to L | Add assign | X+=Y | 14, R to L | Bitwise inclusive OR assign | X|=Y |
| 14, R to L | Subtract assign | X–=Y | 14, R to L | Bitwise exclusive OR assign | Xˆ=Y |
| 14, R to L | Multiply assign | X \* = Y | 14, R to L | Right shift assign | X »=Y |
| 14, R to L | Divide assign | X/=Y | 14, R to L | Left shift assign | X «=Y |
| 14, R to L | Remainder assign | X%=Y |  |  |  |
| **Increment and decrement operators** | | | | | |
| 2, R to L | Preincrement | ++X | 2, R to L | Postincrement | X++ |
| 2, R to L | Predecrement | – –X | 2, R to L | Postdecrement | X– – |
| **Conditional operators** | | | | | |
| 13, R to L | Evaluate either X (if Z≠0) or Y (if Z=0) | Z?X:Y | 15, L to R | Evaluate X first, followed by Y | X,Y |
| **‘Data interpretation’ operators** | | | | | |
| 2, R to L | The object or function pointed to by X | \*X | 2, R to L | The address of X | &X |
| 2, R to L | Cast – the value of X, with (scalar) type specified | (type) X | 2, R to L | The size of X, in bytes | Sizeof X |

Key: Prec. = precedence; L = left; R = right.

# Example preprocessor directives

|  |  |  |  |
| --- | --- | --- | --- |
| **Directive** | **Summary description** | **Directive** | **Summary description** |
| #if | Used for conditionally compiling code, based on evaluation of associated expression. Must be terminated by #endif | #define | Defines string constants which are used in source code and are substituted before code line is compiled |
| #ifdef | Similar to #if, but tests if specified symbol has been defined. Terminated by #endif | #error | Generates user-defined error message |
| #ifndef | Identical to #ifdef, but tests if specified symbol has not been defined | #include | Include at this point the full text from file specified, which may contain unlimited C code, and is then compiled with the source program |
| #else | Used with #if to provide alternative section for compilation | #line | Allows user to specify line number within code |
| #elif | Used within an #if section to test a new expression | #pragma | Allows further directive-like information to be sent to the preprocessor, generally compiler specific |
| #endif | Terminates an #if section. | #undef | Reverses the action of #define, on string constant specified |

# Some applications of punctuation marks

|  |  |  |
| --- | --- | --- |
| Symbol | Application | Example |
| : | Terminates a label | loop: |
| : | Used in bit-field format | unsigned RB0:1; |
| ; | Terminates a statement or declaration | unsigned RB0:1; |
| . | Denotes member of a structure or union | PORTAbits.RA2 = 1; |
| \ | Next line is continuation of this one | – |
| {} | Defines block of code | – |