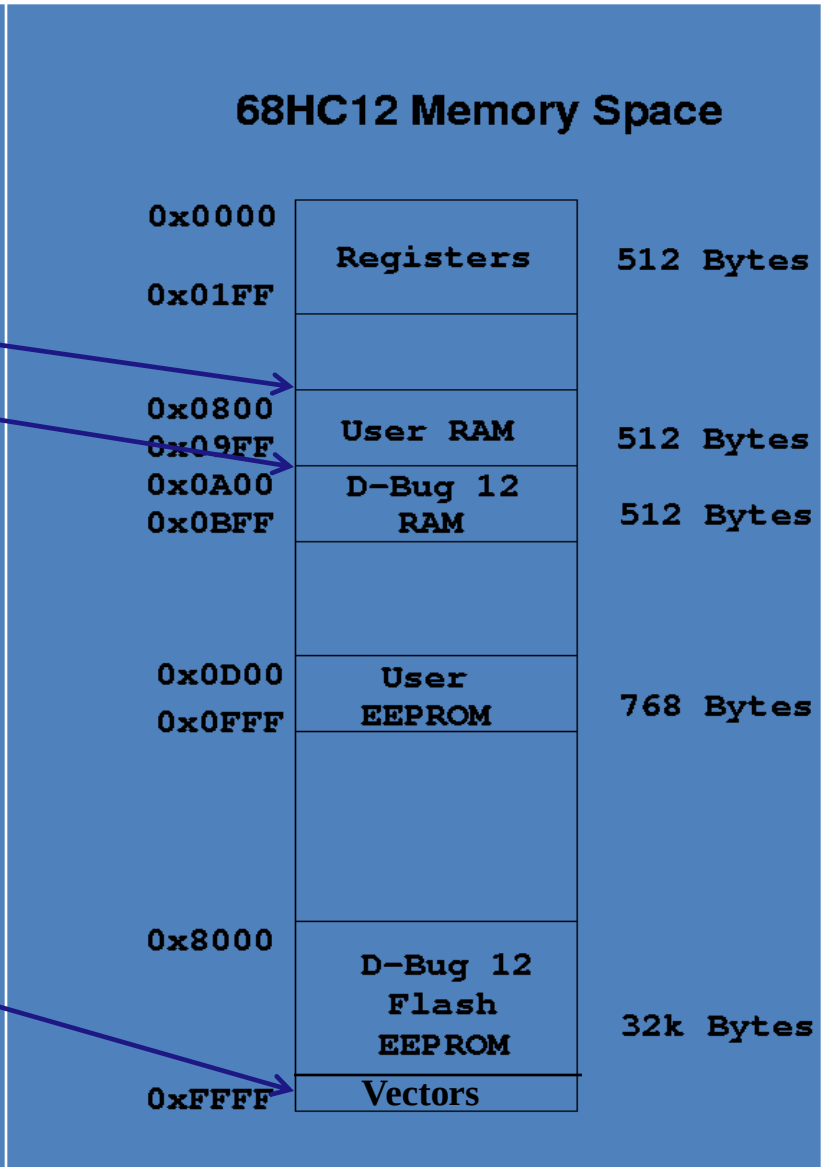
Memory Allocation in Assembly

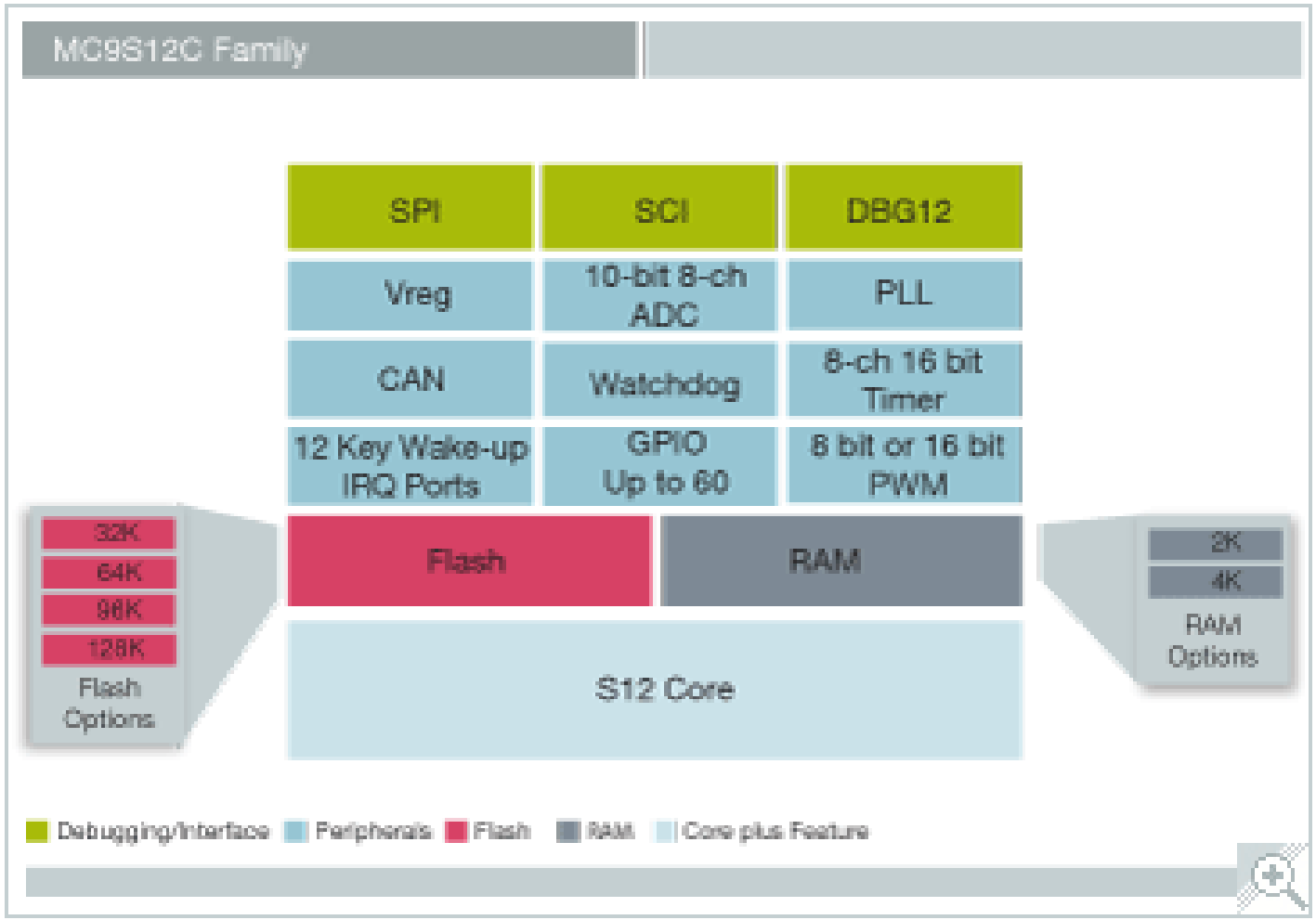
```
PORTA:   equ   $0000
PORTB:   equ   $0001
DDRA:    equ   $0002

         org   $0800
         lds   #0A00  ; initialize stack pointer
         ldaa  #$ff   ; make PORTA output
         staa  DDRA
loop:    ldaa  PORTB ; input from switches
         beq   last
         staa  PORTA  ; output to LEDs
         bra   loop   ; continue
Last:    swi

         org   $FFF6
         dw    $8000
```



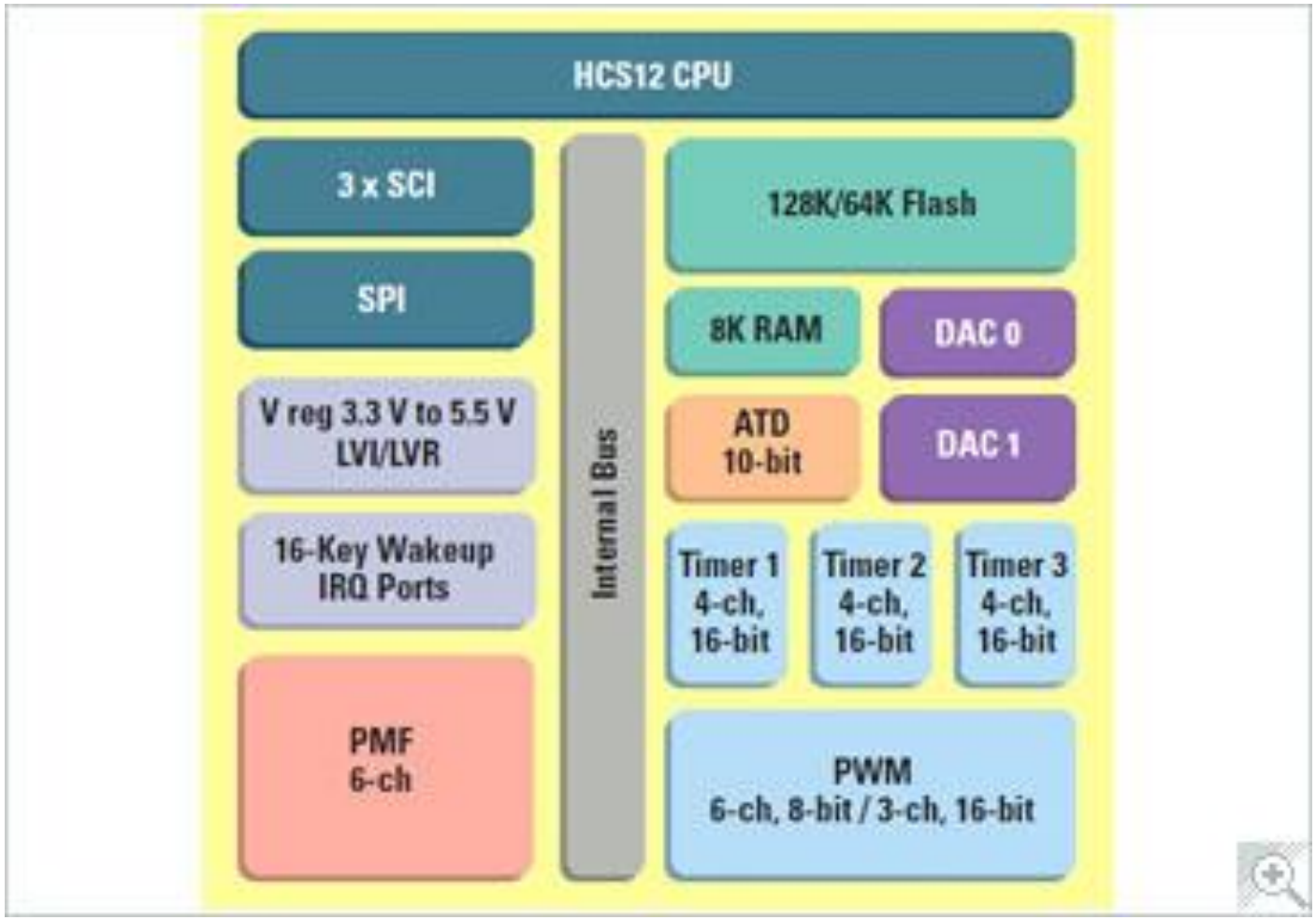
Register Map (MC9S12C)

Address	Module	Size
0x0000–0x0017	Core (ports A, B, E, modes, inits, test)	24
0x0018	Reserved	1
0x0019	Voltage regulator (VREG)	1
0x001A–0x001B	Device ID register	2
0x001C–0x001F	Core (MEMSIZ, IRQ, HPRIO)	4
0x0020–0x002F	Core (DBG)	16
0x0030–0x0033	Core (PPAGE[1])	4
0x0034–0x003F	Clock and reset generator (CRG)	12
0x0040–0x006F	Standard timer module (TIM)	48
0x0070–0x007F	Reserved	16
0x0080–0x009F	Analog-to-digital converter (ATD)	32
0x00A0–0x00C7	Reserved	40
0x00C8–0x00CF	Serial communications interface (SCI)	8
0x00D0–0x00D7	Reserved	8
0x00D8–0x00DF	Serial peripheral interface (SPI)	8
0x00E0–0x00FF	Pulse width modulator (PWM)	32
0x0100–0x010F	Flash control register	16
0x0110–0x013F	Reserved	48
0x0140–0x017F	Scalable controller area network (MSCAN)[2]	64
0x0180–0x023F	Reserved	192
0x0240–0x027F	Port integration module (PIM)	64
0x0280–0x03FF	Reserved	384

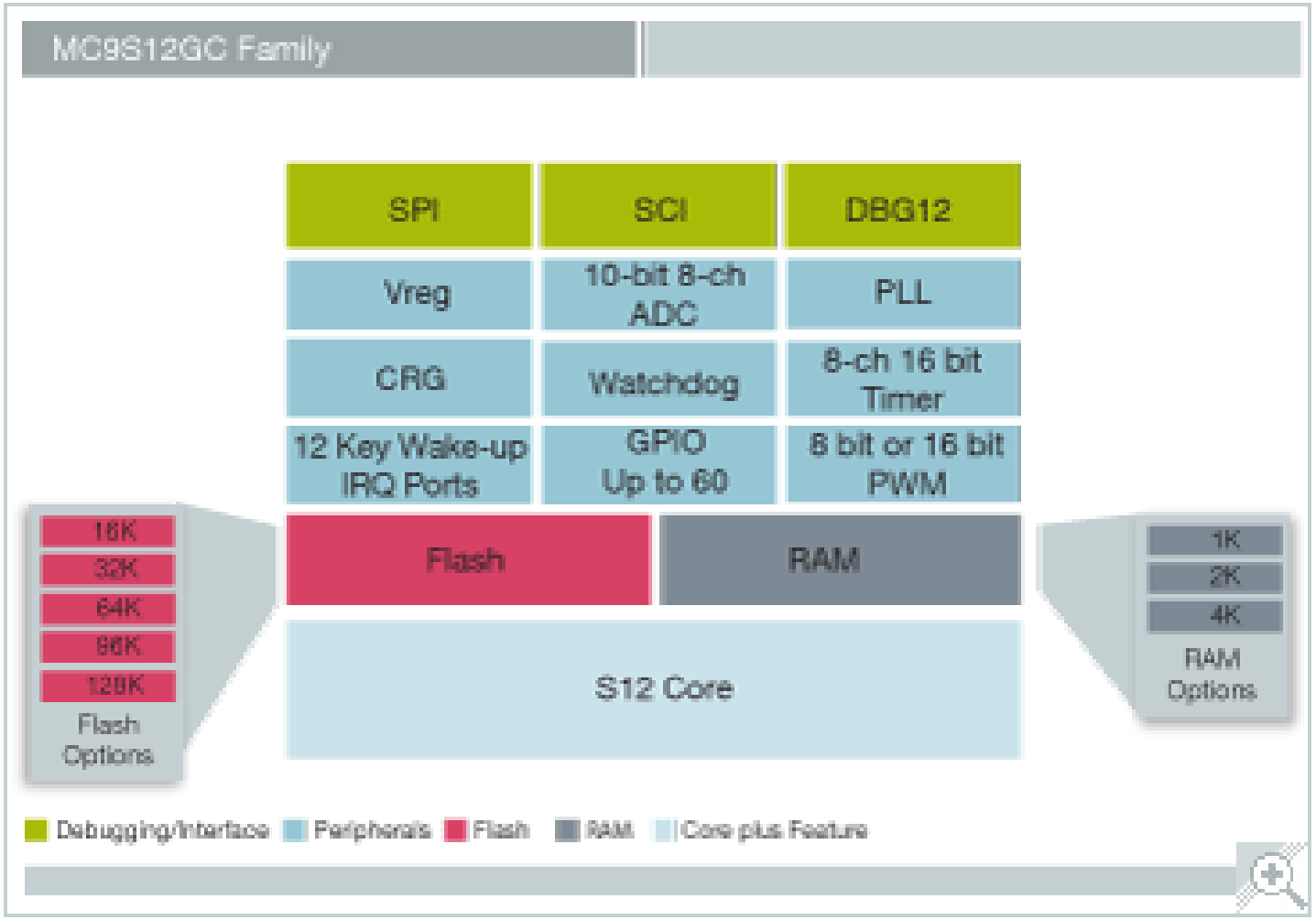
S12C Family (48/52/80 pin)



MC9S12E-Family (112/80 pin)



MC9S12GC128 (80 pin)



Features of the HCS12 Microcontroller

- 16-bit CPU
- Total memory: 64 KB (also supports expanded memory up to 1 MB through a 16-KB window)
  - EEPROM: 0 ⟶ 4KB
  - SRAM: 2 ⟶ 14 KB
  - FLASH: 32 ⟶ 512 KB
- Timer functions: input capture, output compare, pulse accumulators, real-time interrupt, and COP timer
- Serial communication interfaces: SCI, SPI, CAN
- Background debug mode (BDM)
- 10-bit A/D converter
- Instructions for supporting fuzzy logic function

MCU Development Tools

Hardware

- Demo boards
- Development boards
- Evaluation boards
- Cables (USB / RS232)
- Pods

Software

- Text editor
- Cross assembler
- Cross compiler
- Simulator
- Source-level debugger
- Integrated development environment (IDE)

Software Development Tools

On your PC:

- A text editor to enter/edit the program.
- A cross assembler to assemble the assembly programs.
- A cross compiler to compile programs written in HLLs (C).
- A simulator to run the application program without having the actual hardware.
- A terminal program to communicate with the hardware board.
- A source-level debugger to set breakpoints in the program, trace program execution, watch variable values …

An IDE combines all of the above programs in one package so the user can perform all software debugging activities in one environment.

Hardware Development Tools

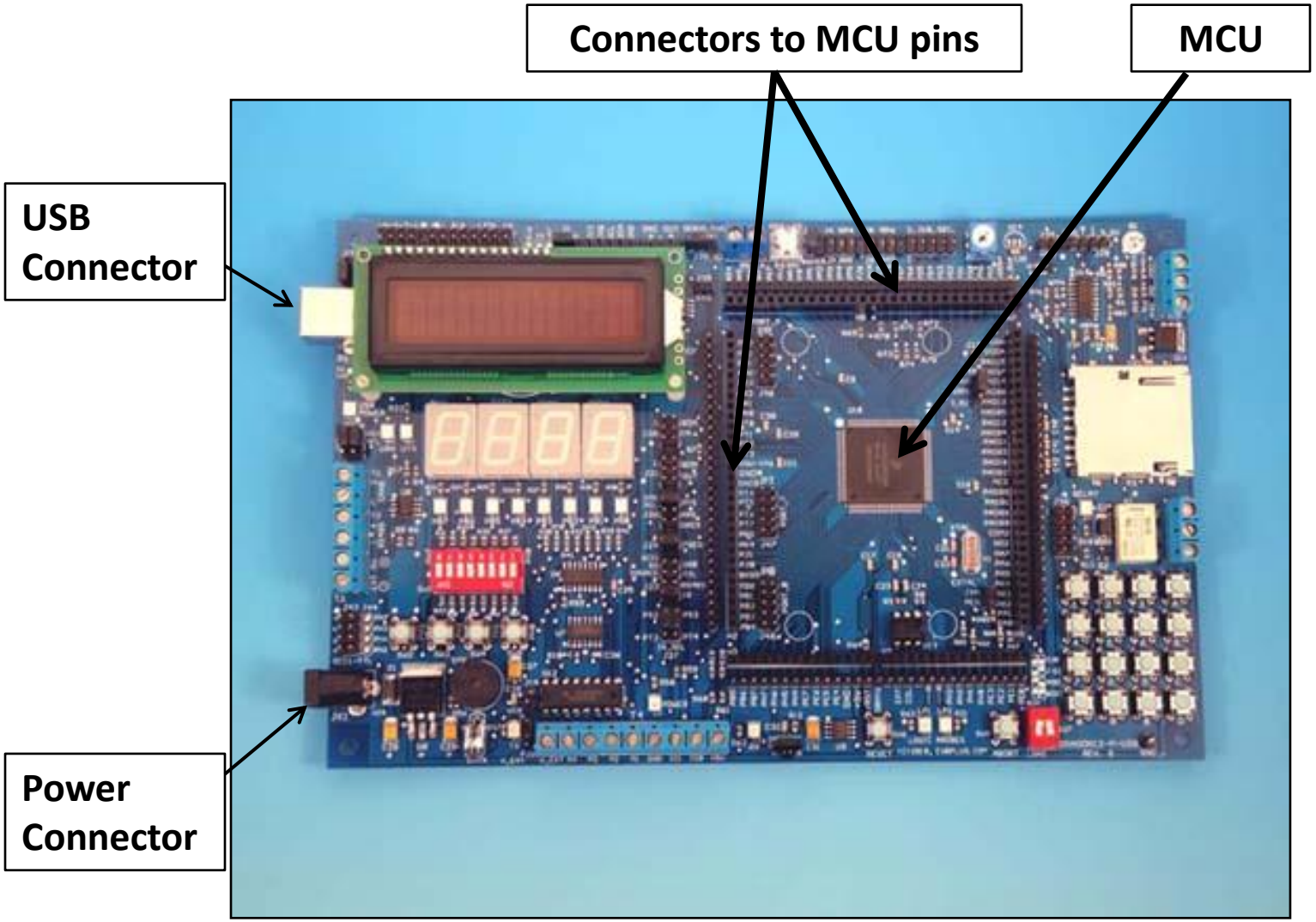
An EVB has:

- A MCU
- Many peripheral connectors:
  - Connectors to MCU pins
  - Connectors to on-chip I/O subsystems (SCI, CAN, ATD, …)
- Input through keypad, DIP switches, or push buttons
- Output through LEDs, 7-segment displays, or LCDs
- On-board D-Bug monitor or Serial monitor program to communicate through a USB or serial port cable with a PC on which the user develops the application program.

  The D-Bug or Serial monitor program allows the user to:
  - Download programs onto the EVB for execution
  - Display register and memory location contents
  - Set register and memory location contents
  - Set program breakpoints
  - Trace instruction execution
- The PC or workstation communicates with the EVB using the terminal program or the IDE.

MCU Evaluation Boards (EVBs)

Example: The Dragon12-Plus-USB



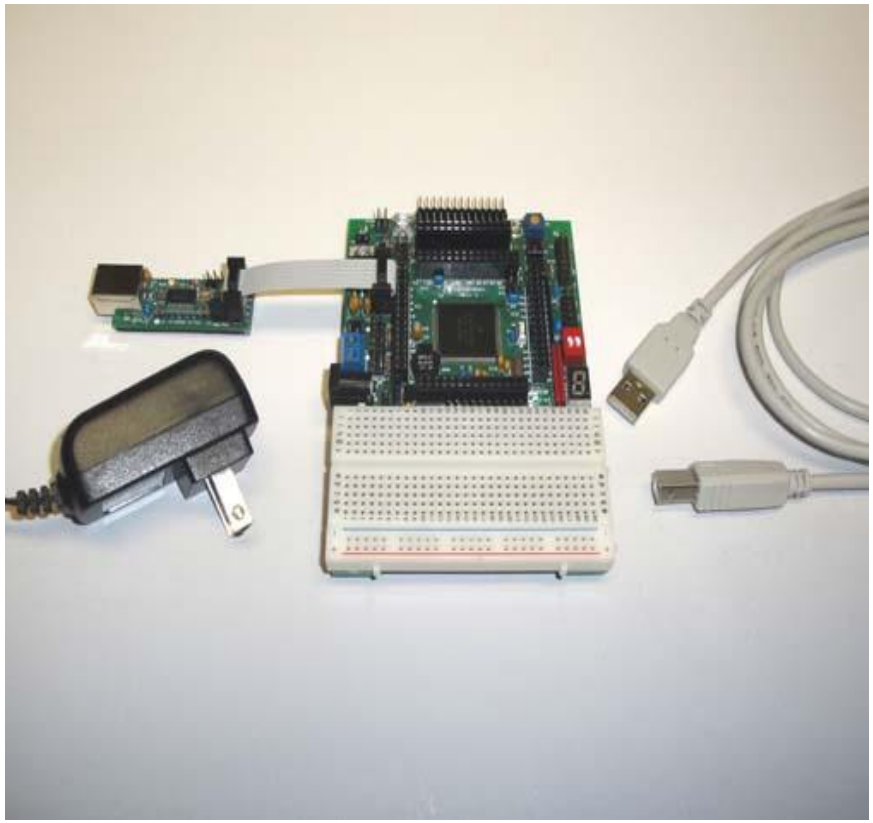
The Dragon12-Plus-USB EVB in testing Motor Control

DC Motor

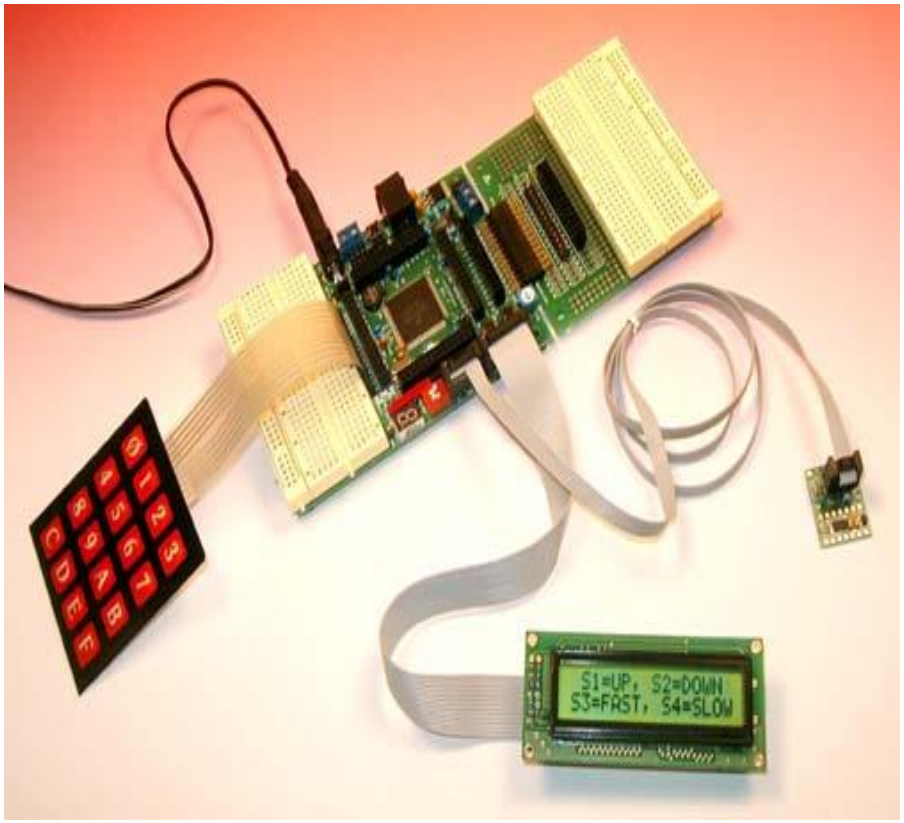
Stepper Motor

Other EVBs

MiniDragon+USB

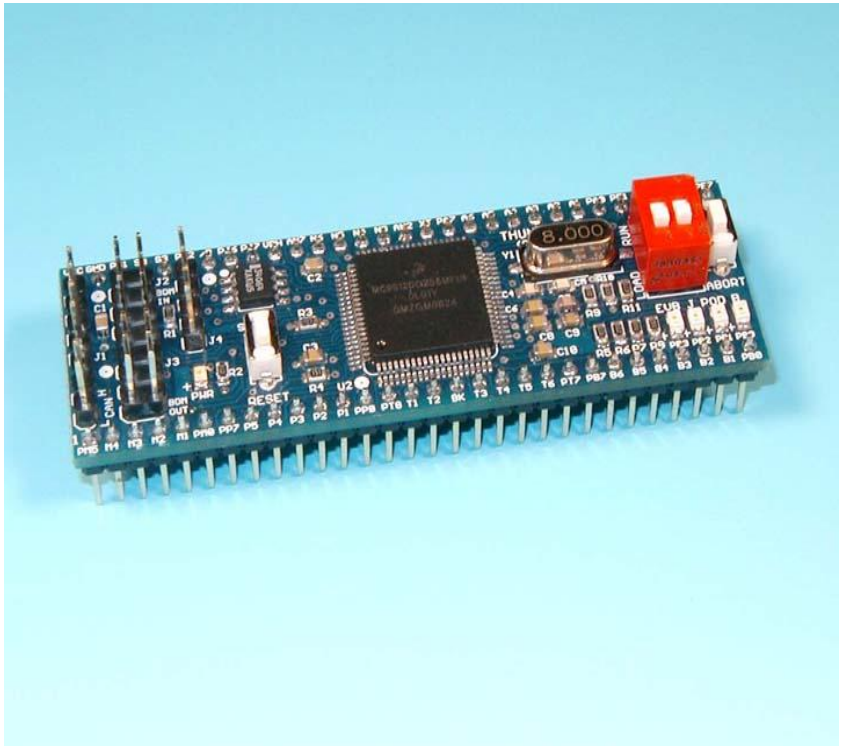


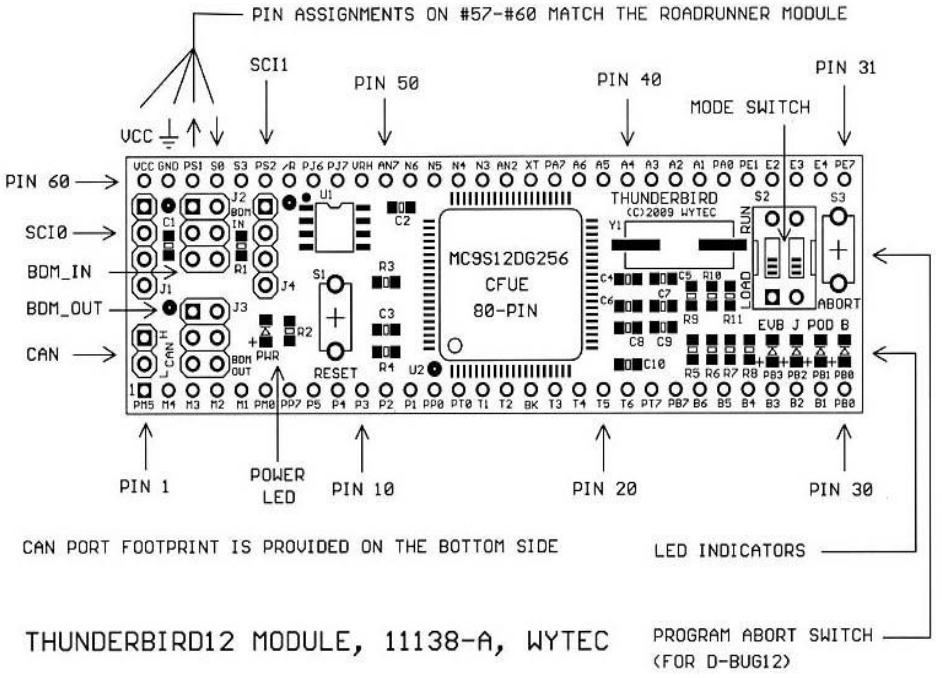
MiniDragon+USB with I/O



Cheaper Development HW

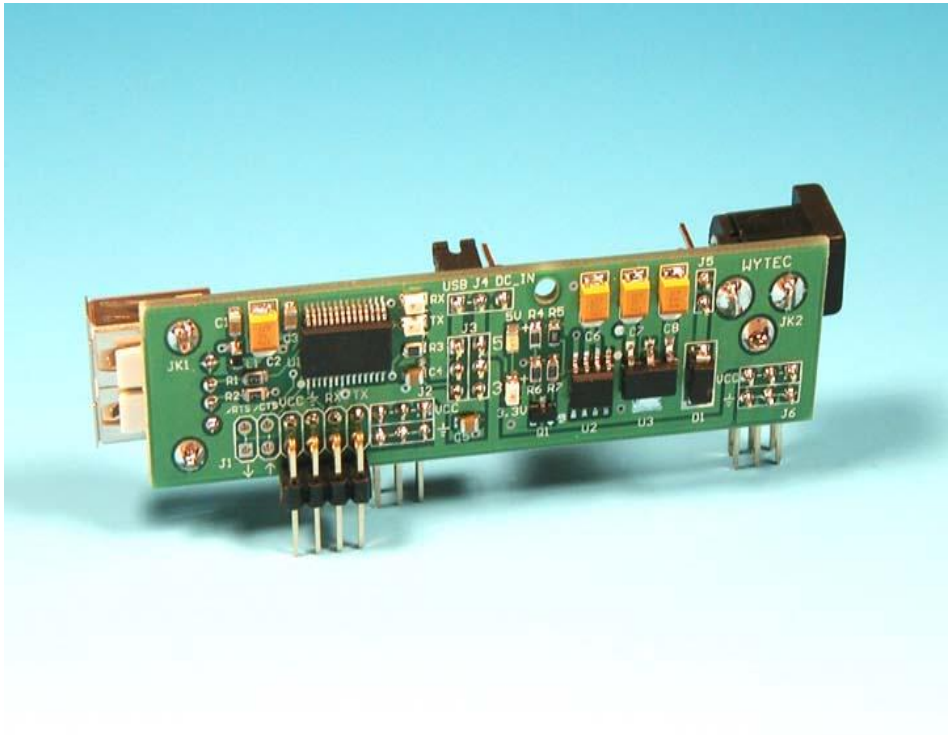
Example: The ThunderBird12 DIP module

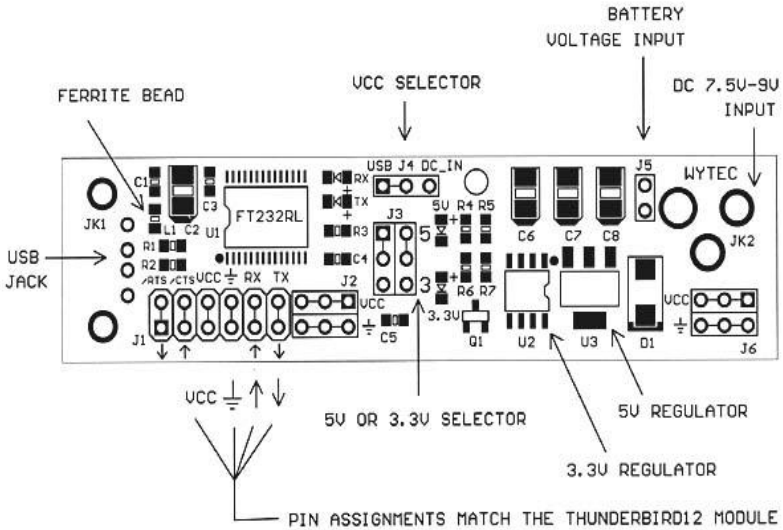




Accessories

Example: The RoadRunner12 USB and Power Supply

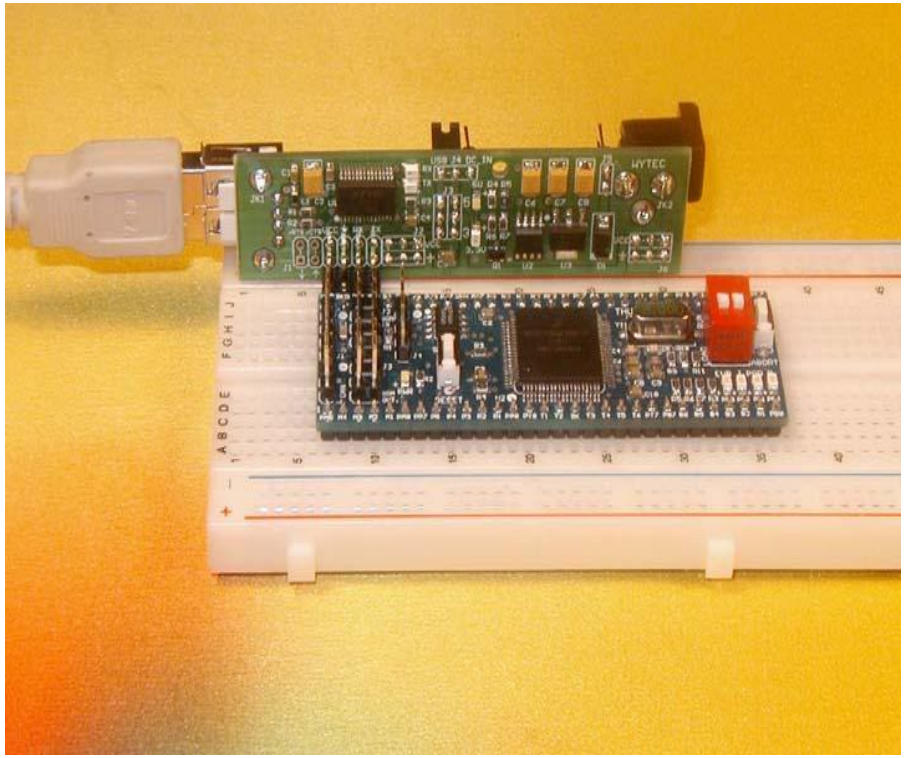




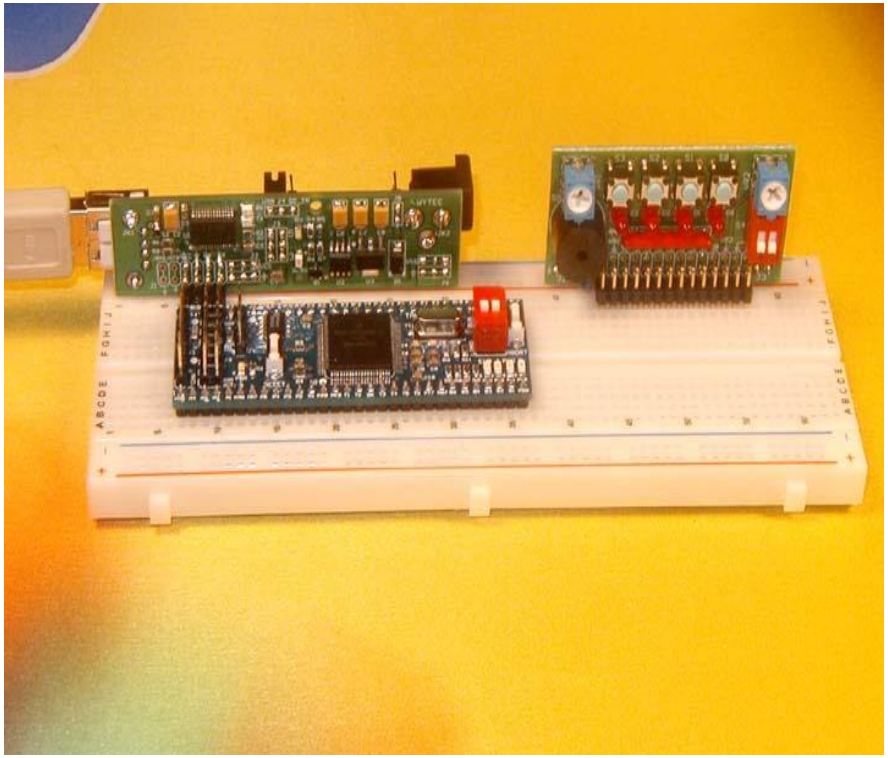
ROADRUNNER MODULE, 11139-A, WYTEC

ThunderBird12 DIP module + Accessories

ThunderBird12 DIP module + RoadRunner12 USB and Power Supply



ThunderBird12 DIP module + RoadRunner12 USB and Power Supply + I/O Module

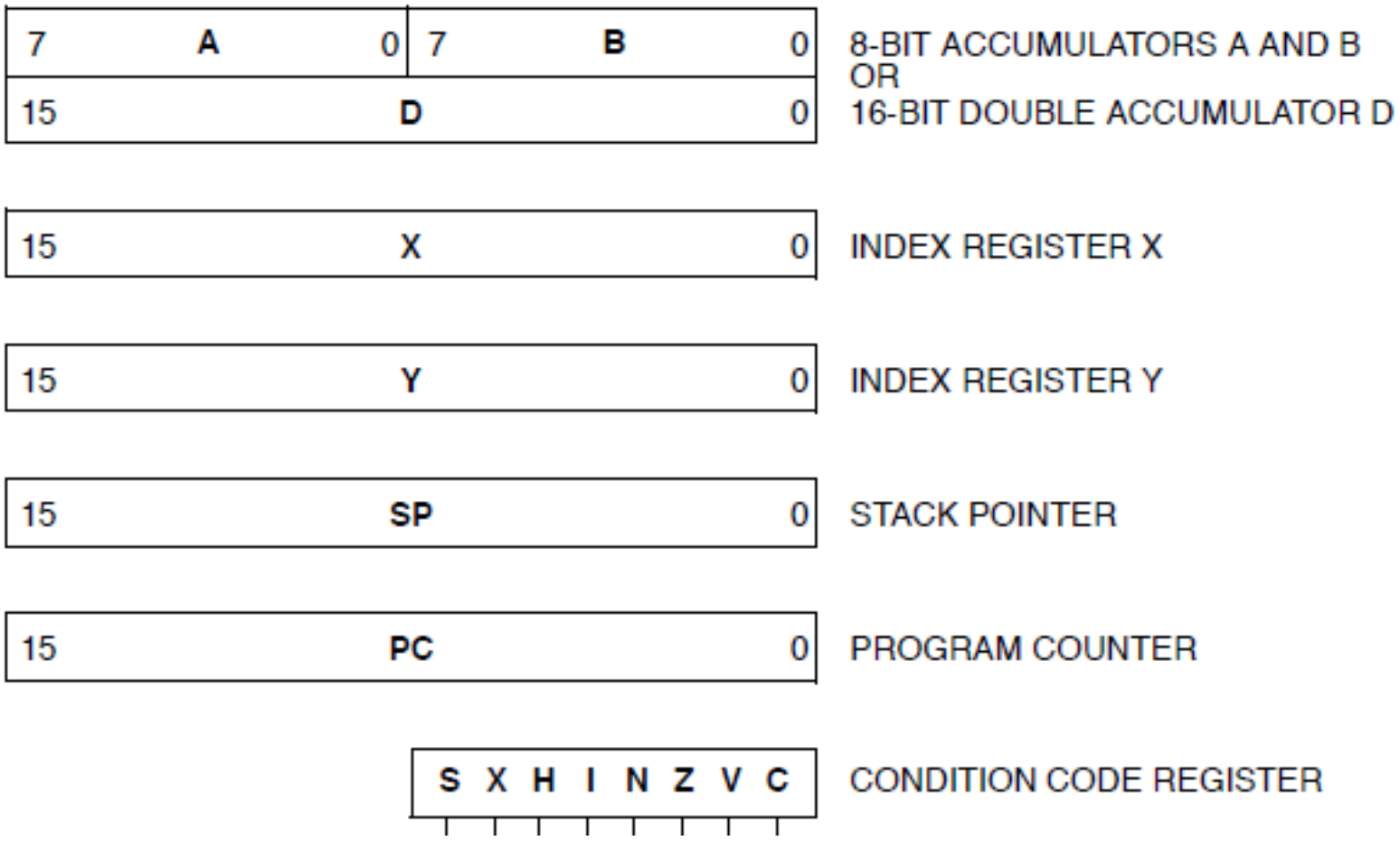


Assembly Programming

Things to know to write Assembly programs:

1. CPU Programming Model
2. Addressing Modes
3. CPU Instruction Set
4. Assembler Directives

CPU12 Programming Model



Condition Code Register (CCR)