PIC18F452

Ken Teo

Lecture Aim

- Overview of the PIC18F452 features
- Demonstrate how to ‘read’ datasheets

Why PIC Microcontroller?

- Motorola (eg. 6800) #1 rank in market share of micrcontrollers before 2002.
- In 2002, Microchip PIC (eg. 18F452) overtook Motorola as #1
- Havard architecture, pipelined (average 1.2 instructions per machine cycle) – fast!
- Modern features (eg. FLASH, A/D, USB – 18F2455)
- Free software/example code (assembler/C etc)
- Able to BOOTLOAD (ie. no need programmer or can update program/firmware in circuit)

16F84

- Page 1 & 2
- 16F84 – relatively simple micro
- 10MHz (400ns per instruction / 4 clocks)
- 1K EEPROM (for program)
- One timer (see WDT as well)
- PORTA/TRISA and PORTB/TRISB
- 4 interrupts

PIC18F452 pg 3 & 4

- 40MHz / 100ns per instruction (4 clocks)
- More EEPROM for program (32K vs 1K)
- More DATA RAM (1.5K vs 36 bytes)
- More DATA EEPROM (256 bytes)
- DATA EEPROM useful for lookup tables (eg. ‘sine wave’) or to store state of the machine even when powered off.

PIC18F452 pg 5

- Serial port
- 8 input A-D (10-bit)
- Multiplier

- Page 6/7 – Thevenin equivalent of I/O ports

PIC18F452 pg 8

- PORTA = actual I/O value
- LATA = data latch (useful to ‘catch’ inputs)
- TRISA = determines input or output of PORTA
- ADCON1 = input can also be used as inputs to ADC

PIC18F452 pg 12

- PORTB (bits 7 to 4)
- Interrupt pins on status change
- Also can be used for in-circuit programming

- Page 13
- PORTB (bits 2 to 0)
- Interrupts on rising or falling edge

PIC18F452 pg 17

- PORTD
- Schmitt Triggers (for noisy inputs)
- Can also be used as a parallel port to interface with memory or for data transfer (see page 18)

PIC18F452 pg 22

- Low and High priority Interrupts

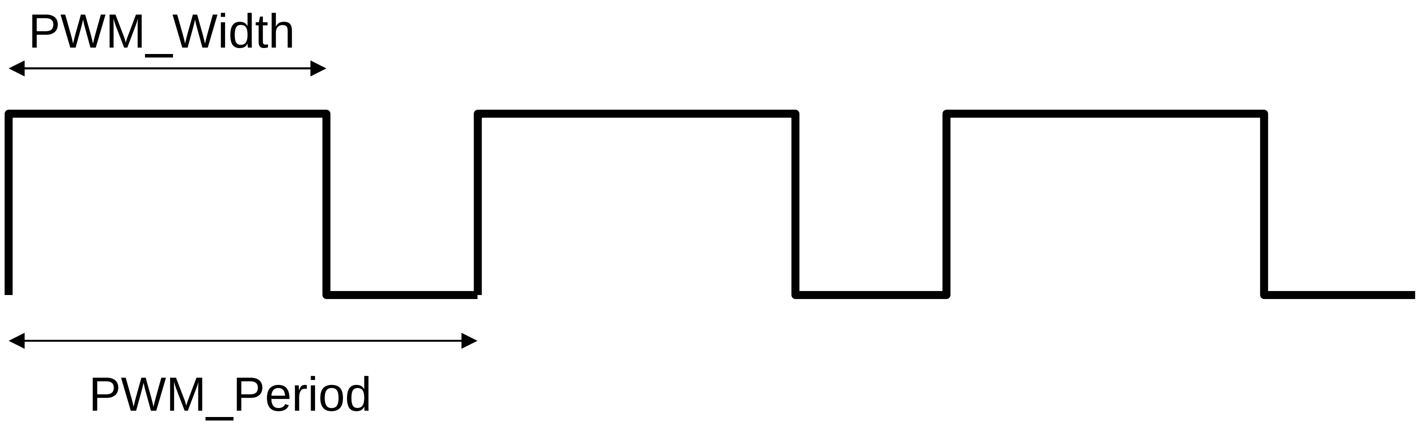
- Page 24
- External interrupts
- PortB interrupts
- Peripheral interrupts (timers, A/D, comms, Capture and Compare, PWM, low voltage)

PIC18F452 pg 26

- Timer0 – 16bits, see how 16bits time value is put onto the 8-bit bus (ie. 2 locations in memory, TMR0L & TMR0H)

- Page 29
- Timer2 – see comparator (ie. can compare with a value, then interrupt). Ie. can be used to generate another clock of defined rate, eg. comms ‘baud’ clock
- Used in Pulse width modulation

PIC18F452 pg 31



- Pg 33
- PWM_DUTY_CYCLE and PWM_PERIOD can be determined using hardware registers
- PWM through a low pass filter gives average voltage (ie. simple DAC)

PIC18F452 pg 34

- 8-channel A/D convertor
- 10-bit resolution, multiplexed (1 at a time)
- Scaled from VSS to VDD or can use AN3 and AN2 for own scaling

PIC18F452 pg 36

- SPI, I^2C
- USART

- Page 43 – USART Transmit
- Baud rate generator / TXREG (data)
- Page 45 – USART Receive
- Baud rate generator / RCREG / Interrupt

PIC18F452 pg 47

- Note – a lot of 8-bit microcontrollers do not provide a hardware multiply
- Hardware multiply increases speed (eg. 8x8 unsigned, 69 cycles down to 1 cycle)

PIC18F452 pg 48

- Program memory
- Program counter gives address of instruction
- Note 31 level stack (holding program counter values)
- Note fixed locations of RESET VECTOR and High and Low priority interrupt vectors (vector = address of subroutine to execute when interrupt occurs)

PIC18F452 pg 48

- Program memory
- Program counter gives address of instruction
- Note 31 level stack (holding program counter values)
- Note fixed locations of RESET VECTOR and High and Low priority interrupt vectors (vector = address of subroutine to execute when interrupt occurs)
- Note 21-bit addresses, 16-bit wide instructions

PIC18F452 pg 50

- Data RAM
- 8-bits wide, since it is a 8-bit micro
- **Either:**
  - **Banked (a=1)**, up to 16 banks of 256 bytes (but not all banks are populated)
  - **Access Bank (unbanked)**, giving 256 bytes
- In banked, note that SFR (Special function registers, ie. timer values, ports etc) are from F80h to FFFh.
- In access bank (unbanked), only first 128 bytes are user RAM, top 128 bytes are SFR's.

PIC18F452 pg 52

- Many oscillator options
- Low power (low speed), since P α freq
- High speed crystal (for timing sensitive applications, eg. for USB, counters)
- RC – cheapest (not timing sensitive applications) – accuracy depends on tolerance of R & C. See page 55

PIC18F452 pg 58

- In circuit programming – via 6 pins.
- Ie. update the program whilst chip is on PCB
- Enables bootloading – on boot, chip can check an input pin to see if it is to ‘run normally’ (jump to normal operation code) or ‘to program itself’ (jump to programming code).