1. 32: I/O, TB0, Slaveout/Masterin, I2C clk, BSL clk, TA 1

33: I/O, TB0

34: I/O

1. Er zijn relatief weinig opties ten opzichte van andere hoofdstukken.
2. P4DIR |= 0x80;
3. Energiebesparing
4. Een interrupt werkt als een onderbreking van de code. Wanneer er een interrupt wordt geactiveerd wordt er per direct naar een andere plaats in de code gegaan. Daar kan iets worden gedaan en dan kan de interrupt weer gedeactiveerd worden.
5. De main loop staat op dat moment stil
6. De originele service routine wordt uit de stack gehaald waardoor het normale programma weer kan lopen.
7. Bij polling moet de CPU continu controleren of er nieuwe data is. Met een interrupt is dit niet nodig, er wordt nu pas gecontroleerd als er daadwerkelijk nieuwe data is.
8. UART, I2C, RTC, SPI
9. Interrupt signaal op de interrupt pin.
10. Tabel met ISR adressen.
11. Interrupts met hoge prioriteit zijn vaak kritiek en met lage prioriteit niet zo belangrijk.
12. WDT Interval Time
13. Afbeelding met tekst, Lettertype, schermopname, wit

    Automatisch gegenereerde beschrijving
14. P4.5
15. Voorkomen van zwevende inputs.
16. Afbeelding met tekst, Lettertype, schermopname

    Automatisch gegenereerde beschrijving
17. Het gaat erg traag, maar dit komt doordat hij elke keer stopt bij de breakpoint.
18. Afbeelding met tekst, schermopname, Lettertype, nummer

    Automatisch gegenereerde beschrijving
19. if (P1IFG & 0x02)

