**Grunnfunksjon/maler for Atmega 328p.**

To Do Knapp, led og delay(Grunntest for mikrokontrollern)

To Do Uart

To Do Komparator

To Do ADC

To Do Hardware interrupt

To Do Timer

To Do PWM

To Do Servo

To Do SPI

Andre ting

To Do Assembly

**Eksamen temaer**

**Moduler:**

* + ADC+++
    - Husk tid
    - freerunning
  + Timer++++
    - Husk tid
  + PWM+
    - Husk tid
  + UART+
    - Sette baud og alt
  + Servo

**Assambley:**

* + SREG , r5
  + Laste inn, r18
  + Kopier. R16, r0
  + Hvis PB7 er lav, hopp over.
  + Sette OUTPUT bit +

* + Operasjonskoden++++
    - Koden maskinen tar inn i enere og nuller
    - Har frie deler som kan velges
      * D og k som har
      * LDI: 1110 kkkk dddd kkkk
  + Programteller
    - Bestemmer hvor i programminne koden neste instruksjon skal hentes

**Tema:**

* + Minne og hva de gjør. ++
    - SRAM, FLASH, EEPROM
    - Husk størrelsen til minne og antall ganger den kan overskrives
    - SRAM:
      * Volatile, lagres ikke ved strømbrudd.
      * Brukes for variabler og stack(mellomlagring)
      * Også I/O
    - FLASH
      * Der selve programmet ligger
      * Kan lagre konstanter
      * Lagres i words/pages?
        + Words/pages: Annerledes enn bites.
      * Begrenset antall ganger den kan skrives til.
    - EEPROM
      * Ikke volatile minne
      * Må spesifiseres at det skal skrives hit.
      * Tabeller, parametere og debug info.
      * Treg
      * Begrenset antall kan skrives
  + Stack og rekkefølgen.
    - Stack: Midlertidig minne(SRAM)
    - Stack peker: peker hvor man skal starte å hente informasjon fra SRAM, ut ifra tidligere operasjoner
  + Avbrudd (stikkord: Programteller, stack, prioritet, responstid)++
    - Avbrudd bryter stackrekkefølge(sniker i køen)
    - Hvis du avbrudd forekommer så blir det med lavest adresse prioritert
    - Responstid: Antall sykluser det tar å utføre et avbrudd.
  + Arkitektur. Von Neuman eller Harvard. Tegning?
    - Minne arkitektur
    - Von Neuman:
      * Instrukser og data ligger på samme minne enhet,
      * Seperat prossessor(Kontroll enhet)
      * En del som gjør mattematikk og logikk(ALU)
      * Inn ut enheter som kontrollere kommuikasjon mellom bruker og kontrollenhet

* + Harvard:
    - Seperat minne for data og instrukser
    - Ofte "modified" harvard.
  + RISC:
    - Reduced instruction set computing
    - Bryter ned oppgaver til enkle deler og utfører en etter en hver klokkesyklus
    - Mindre strøm
  + CISC:
    - Complex instruction set computing
    - Gjør store oppgaver samtidig, men kan være ueffectiv og kan ta flere sykluser
    - Mer menneskeligvennlig
  + Pekere(X,Y,Z)

* + Strømsparing
    - Kan settes i strømsparing
    - Power-down, power-save, standby, extended standby, idle
    - Hver har forskjellig strømtrekk og forskjellige våknemuligheter

* + Asynkron vs synkron overføring
    - Asynkron: Begge er eneige om komunnikasjonshastigheten på forhånd
    - Sykron: En master kommuniserer klokkesyklusen
  + 4 hovedtyper med instruksjoner
    - Aretmethic and logic
    - Bit og bittest
    - Branch
    - Data-transfere
    - (MCU)
  + Avbrudsrutine vs funksjon
    - Avbruddsrutiner blir kalt på av interrupt vektoren
    - Funksjon kaller du selv på.

**Ord oguttrykk:**

* + Kommunikasjonsgrensesnitt
    - USART, SPI, TWI(I2C), ISP(?), USB
  + ASF
    - Atmel software framework
    - Atmels sitt bibliotek samling
  + Aktiv lav
    - En side koblet til jord andre siden koblet til GPIO
    - Setter GPIO lav for å slå på dioden
  + Bootloader
    - Kode som blir utført for at du skal kunne kode microkontrolleren
    - "Oversetter" koden for mikrokontrolleren
  + Simplex, halv-duplex og full-duplex
    - Begrensningene til kommunikasjonsprotokolene
    - Simplex: Kommunikasjon kun en vei
    - Halv-duplex: Kommunikasjon begge veier, men kun en om gangen
    - Full-duplex: kontinuerlig kommunikasjon begge veier
  + Endian
    - Hvilken vei bytrekkefølgen er i.
    - Big-endian: Most significant byte i de minste addressene.
    - Small-endian: Least significant byte i de minste adressen.

Register oppbygning og assembler:

AVR-Atmega328p har 32 «general purpose» registre. De 16 første er reserverte, og de 16 siste kan brukes som registerminne, til å legge inn variabler før flytting av dem til SRAM. De 6 siste registrene av de 16 siste er X,Y, og Z pekerne, som er register som har en spesifikk lagringsposisjon i SRAM. Assembler kommandoen «LDI» kan kun benyttes til å skriv verdier til de 16 siste registrene, «MOV» kan brukes til å flytte registre.