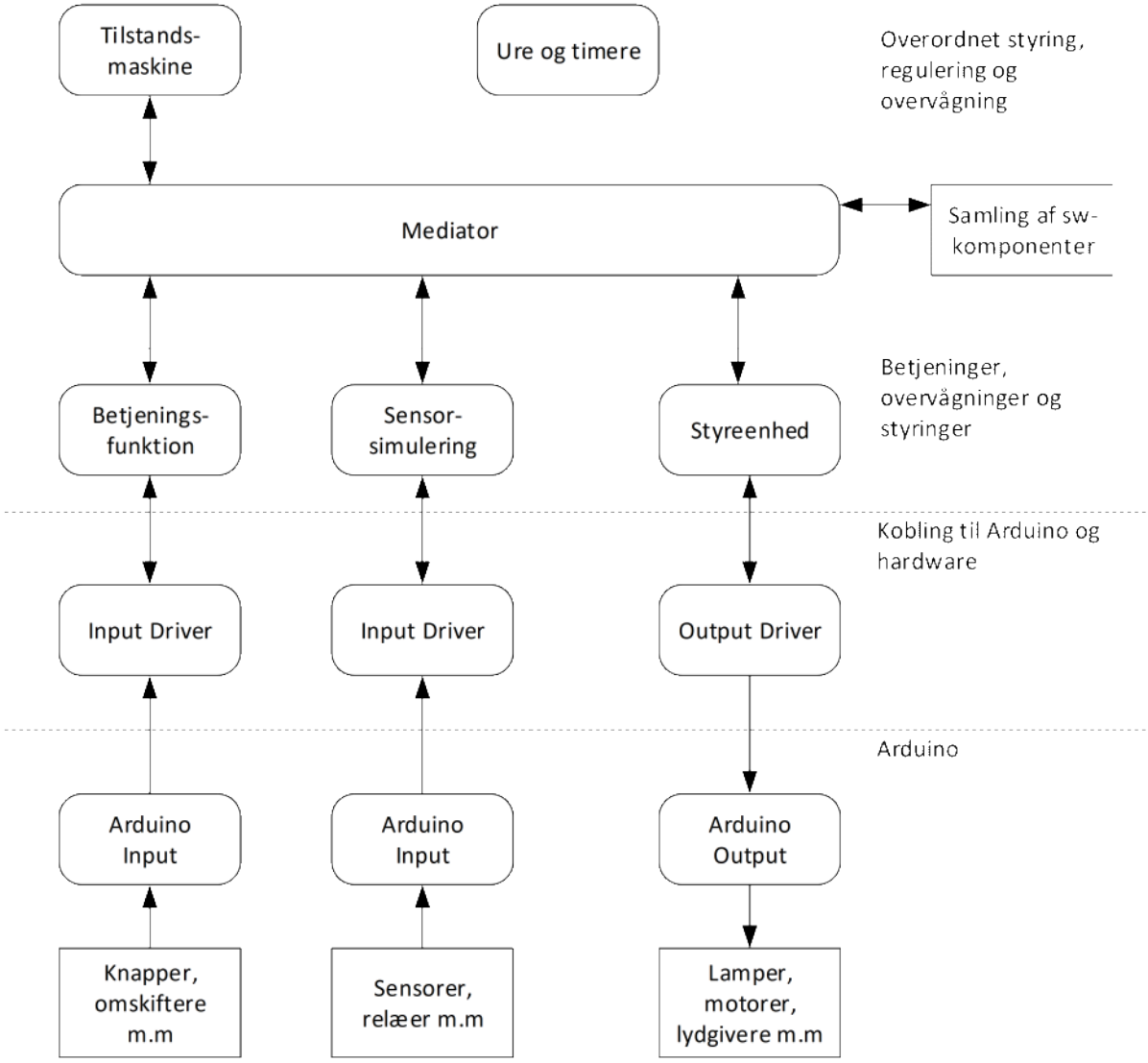
Arduino applikation

7. januar 2023 21:34

Nogle gange er der behov for at kode en simpel Arduino applikation på op til ca. 50 programlinjer. Den kan bruge procedural programmering.

De store applikationer bliver bygget med softwarekomponenter. Det gør applikationen fleksibel og giver mulighed for at opdatere i flere cyklusser efterhånden som kravene bliver afklaret. Softwarekomponenter er bygget færdig og uden behov for modifikation. Nye komponenter kan blive tilføjet.

Den store applikation er bygget således:



Software består af komponenter, hver med deres grænseflade og specifikke funktion.

- Arduino er koblet til betjeninger, sensorer, lamper, motorer m.m. Det er hardwarelaget.
- Drivere konfigurerer Arduino parallelle porte og seriel kommunikation. En driver kan udskiftes med en anden driver, uden det giver behov for at opdatere applikationen. Dermed bliver applikationen uafhængig af tilkoblingen til Arduino.
- Applikationen indeholder funktioner for betjeninger og sensorer. I det lag kan der også indbygges behandling af inputdata: Beregninger, binær logik, lagring af data til senere brug osv.
- Til styringer indeholder applikationen styreenheder for lamper, motorer m.m. I det lag kan der også indbygges behandling af outputdata: Beregninger, binær logik, blink til lamper, styring af

motor osv.

- Tilstandsmaskinen udfører enhver tilstand: Ændring af tilstand som følge af inputdata. Opdatering af output, både når der bliver skiftet til en tilstand og når en tilstand forlades. En ny tilstand kan også udløses når en tid er udløbet.
- Mediator formidler kommunikation imellem softwarekomponenter. Mediator bruger en samling af softwarekomponenter. Mediator udfører også tidsstyring i betjening, sensorer og styreenheder.

Hvis softwarekomponenter udfører direkte funktionskald, skal komponenten tilpasses de komponenter softwaren består af. Men så er princippet, at komponenter er færdigbygget brudt. Alle softwarekomponenter bliver lagt i en samling, hvor de får en adresse. Dataoverførsel formidles via adresser imellem softwarekomponenterne af mediator. Opsætning af samlingen af softwarekomponenter sker ved opstart af Arduino.

En konkret applikation skal have sit eget design dokument, der forklarer hvordan applikationen er bygget og hvorfor den er bygget på den måde.

Arduino tilpasning

7. januar 2023 21:41

Arduino Uno har ikke et operativsystem. Program skal levere al maskinkode. Der er begrænset memory til data 2kbyte og 32kb til program.

Memory allokering og deallokering kan udvide brug af memory, men indfører risiko for at fylde memory op med døde data. I denne model er alt memory lagt fast på compile tidspunktet.
Når der kun afsættes statisk memory får det som konsekvens:

- Alle objekter bliver instantieret.
- Antal elementer i et array bliver lagt fast.

Normalt håndteres tryk på en knap og udløb af en timer med interrupt. Det er en kompleks mekanisme, der skal programmeres og testes omhyggeligt, for at forebygge runtime fejl. Istedet for bliver der brugt en model med polling. Tidsinterval for polling bliver så kort, at f.eks. et knaptryk bliver besvaret hurtigt nok. Software moduler med tidstyring skal indeholde metoden doClockCycle.

Fejlhåndtering med exceptions og throw er også en kompleks mekanisme, der så vidt vides ikke understøttes af arduino. Softwarefejl må forebygges med grundig testning.

Grænseflader - softwarelag

5. februar 2023 14:41

Mellem hver af de 3 lag software er der en grænseflade, med metodenavne og dataprotokol.

En softwarekomponent har pointere til de softwarekomponenter den skal kommunikere med. Det gør det muligt at udpege en anden softwarekomponent, i stedet for at ombygge.

Softwarekomponenter skal have fast indbyggede navne for metoder, så metodenavne ikke skal tilpasses, hvilket bryder princippet, at komponenter er færdigbygget.

Softwarekomponenter skal have fast indbyggede dataprotokoller, så de ikke skal tilpasses, hvilket bryder princippet, at komponenter er færdigbygget.

- Digitale drivere - grænseflade.
  Datatype er bool.
  Bruger hardware begreber: LOW og HIGH.
- Analoge drivere - grænseflade.
  Datatype er int.
- Digitale funktioner, multivibratorer og lignende - grænseflade.
  Datatype er bool.
  Bruger funktions begreber: OFF og ON.
- Multivibratorer, timere og lignende - grænseflade.
  Datatype er bool.
  Bruger funktions begreber: OFF og ON.
- Betjeninger, sensorer og styreenheder - - grænseflade.
  Datatype er byte.
  Bruger status begreber: OFF, ON, FREE, OCCUPIED, UP, DOWN o.s.v. Der er mulighed for 256 tilstande.

Der bliver brugt positiv logik i applikationen. Så aktiv er for eksempel ON = 1.