# SD kort driver

I forbindelse med projektet er der behov for en permanent lagring af data som også kan bevare de gemte data ved eksempelvis en strømafbrydelse. Til dette anvendes der et kingston 8 GB sdhc kort der anvendes i SPI mode, kortet følger sd simplified specification for sdhc kort, der er vedlagt som bilag, hvilket gør det muligt at skrive en fornuftig driver til at håndtere kortet, driveren her vil virke med alle sdhc kort der følger sd simplified specification. UML klassediagrammet for sdCard klassen ses på Figur 1, hvor det også ses at der anvendes et SPI objekt til kommunikation med sd kortet, begge klasser beskrives i de efterfølgende sektioner.



Figur 1 - sdCard UML klassediagram

## SPI driver klasse

Klassen SPI anvendes her som kommunikationsvej mellem vores sdCard klasse og vores 8gb SDHC kort. Denne implementeres via det indbyggede SPI interface i vores atmega2560, hvor atmega2560 anvendes i master mode.

I beskrives SPI klassens funktioner.

|  |  |
| --- | --- |
| Funktion | Beskrivelse |
| SPI(int freq) | Klassens konstructor, freq er den frekvens angivet i kHz der ønskes anvendt som clock frekvens på spi modulet. Der kan vælges mellem følgende frekvenser:  125 kHz  250 kHz  1000 kHz  4000 kHz |
| writeByte(unsigned char data) | Skriver 1 byte (8 bit) data til SPI bussen på MOSI porten. |
| recieveByte() | Læse 1 byte (8 bit) data fra SPI bussen på MISO porten. |
| setFreq(int freq) | Ændre clockfrekvensen på SPI bussen til den ønskede frekvens, angives i kHz og kan være en af følgende frekvenser: 125 kHz, 250 kHz, 1000 kHz, 4000 kHz. |

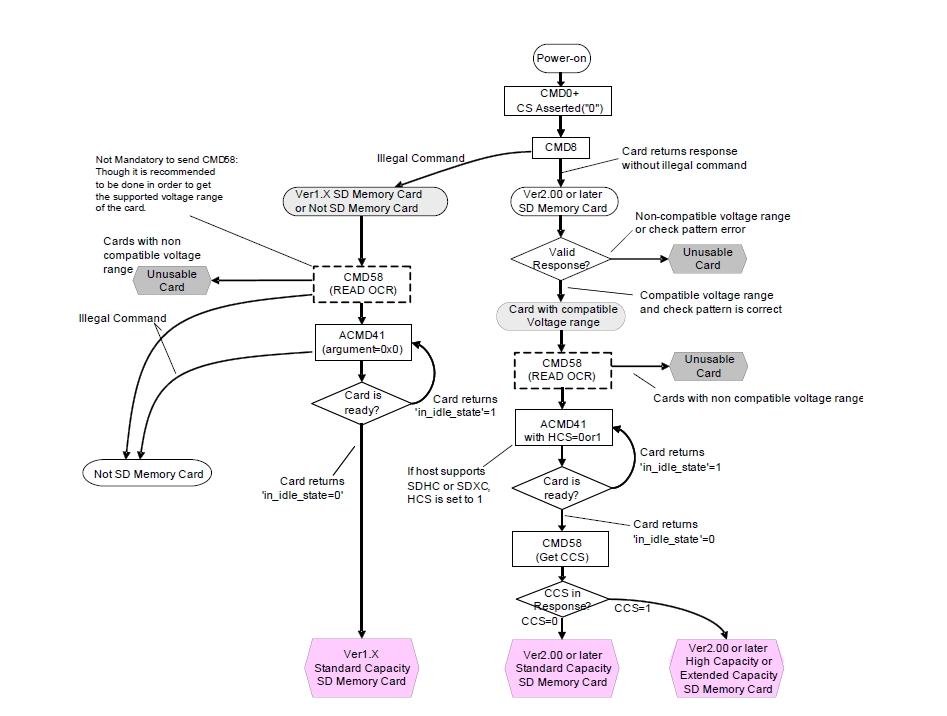
Tabel - funktionsbeskrivelser for SPI klassen

## sdCard klassen

sdCard klassen følger som nævnt sd simplified specifikation som er vedlagt som bilag. Der er i nogle sammenhænge set bort fra dele af specifikationen der vedrøre ældre sd-kort samt mmc kort, da vi i projektet kun anvender sd-kort af typen SDHC.

Initialiseringsprocessen for et sd-kort er rimelig kompliceret, og kræve en længere sekvens af kommandoer der skal sendes til sd-kortet og kan ses på Figur 2. Der er udeladt alle dele af initieringsprocessen der ikke vedrøre kort af typen SDHC.

Det er ifølge specifikationen nødvendigt at starte med en SPI clock frekvens på under 400 kHz for samt at der sendes mindst 70 clock cycles inden initieringssekvensen startes. Når initieringsprocessen er udført og kortet er klar til brug kan frekvensen hæves til den ønskede clock frekvens for at øge datatransmissionshastigheden.



Figur 2 - Initieringssekvens for sd-kort, udklip fra sd simplified specifikation.

I denne version af sdCard driveren anvendes initieringsprocessen for V2.00 eller senere high capacity kort, er kortet ikke af denne type vil initieringsprocessen fejle.

Ud over den specielle initieringsproces, skal man herudover være opmærksom på at sd-kort af denne type kun kan læse og skrive en hel block af gangen. En block på et sd-kort er 512 bytes (1 byte = 8 bit), hvorfor sd-kortet adresseres via blocklocationer, dvs. adressen 0 = første 512 bytes på kortet, adressen 1, er de næste 512 bytes på kortet.

|  |  |
| --- | --- |
| Funktion/attribut | Beskrivelse |
| sdCard(int speed) | Constructor der initiere et nyt objekt af typen sdCard, speed angives til den ønskede frekvens i kHz der ønskes når initialiseringsprocessen er udført. Speed kan være en af følgende frekvenser:  125 kHz  250 kHz  1000 kHz  4000 kHz |
| Init() | Udføre initialiseringsprocessen af sd-kortet, returner true hvis initialiseringsprocessen lykkedes, false hvis den fejler. |
| writeBlock(unsigned long adress, const unsigned char data[]) | Modtager en 32 bits adresse samt et 512 bytes array, og skriver det 512 bytes array til den block der ligger på den angivne adresse på sd-kortet. Returnere true hvis det lykkedes at skrive på sd-kortet, false hvis der opstår en fejl. |
| readBlock(unsigned long adress, unsigned char data[]) | Tager en 32 bits adresse samt et 512 bytes array og henter den block på sdkortet der ligge på den 32 bits adresse der modtages. Denne block fyldes nu i det 512 byte array der er modtaget som parameter. Lykkedes det ikke at læse blokken returnes der false. |
| SPI spi\_obj | SPI objekt til at håndtere selve SPI kommunikationen. |
| speed | Speed attributten indeholder den ønskede hastighed der blev valgt ved oprettelsen af objektet. |

Tabel 2 - funktionsbeskrivelser for sdCard klassen.

Angives der adresser der ligger uden for det antal blokke der er tilgængeligt på sd-kortet vil funktionerne returnere false, hvorved der kan reageres på de opståede fejl.

Ønskes der yderligere informationer om sd-kort via SPI protokollen henvises der til sd simplified specifikation der er vedlagt som bilag.