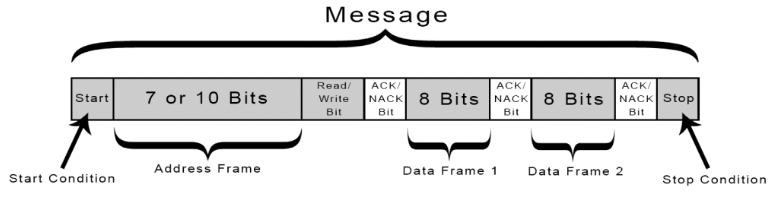
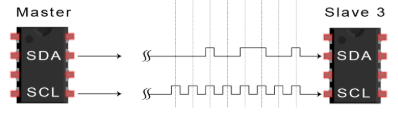
I2C protokol

I2C er en kommunikations protokol, som bruges når der skal kommunikeres mellem forskellige devices. Som protokol har I2C den fordel at den er en blanding mellem SPI og UART. Hvad angår hardware, så er I2C hardware mindre kompliceret end UART. I2C består af to linjer, en SDA og en SCL.

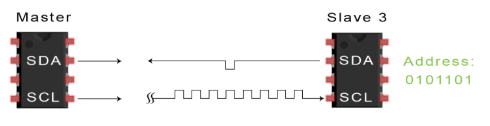


# Read/write



Når masteren skal skrive eller læse fra slaven, gør den det ved at den sender først en 7 eller 8 bits adresse ud til slaven den vil kommunikere med, hvoraf sammen med de bits sender den også en read eller write bit.

# Addressing



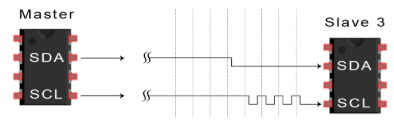
Når masteren kommunikerer med slaven, sender den signaler ud til alle slaver som den er koblet op til. Det den gør er at den sender en adresse ud hvorefter at alle slaverne vil sammenligne adressen som masteren har sendt ud med sit eget, og hvis masteren har fået fat i den rigtige slave, så sender slaven et svar tilbage ved form af en ACK(acknowledge) hvor signalet går low(0) i et bit.

# Data frame

I løbet af at koden kører vil masteren hele tiden prøve og kommunikere med slaverne som den er koblet til, og hvis den har fået et svar fra den slave som den gerne vil snakke med, begynder den sende data frames som er 8 bits lang, hvad slaven egentlig skal gøre, er at den enten skal sende en ACK eller NACK, som betyder at slaven har fået hele beskeden og den så vil svare tilbage til masteren med at den sender et bit hvor SDA linjen bliver sat i Low(0).

Men hvis slaven så ikke få hele beskeden og noget er gået galt vil slaven ikke rigtigt svare tilbage den vil bare ikke gøre noget som helst og SDA linjen vil forblive på High(1) som vi så kalder for NACK altså den har ikke fået alle de 8bits som masteren sendte afsted.

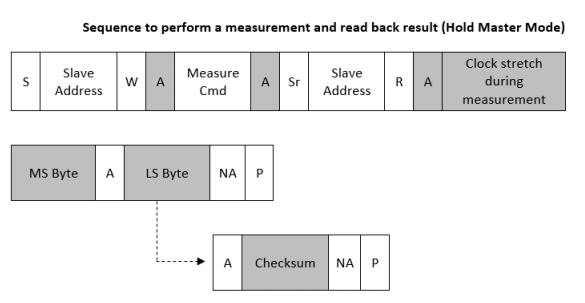
# Stop bits



Når masteren har sent eller modtaget alt det data som den har brug for skal man på en eller anden måde stoppe det, den måde det bliver gjort er ved at når enten masteren eller slaven har fået en ACK, vil masteren så trække SCL signalet til high(1) hvorefter SDA signalet også bliver trukket op til High(1).

# ADAFRUIT Si7021-A20

Si7021-A20 er den temperatur sensor som bliver brugt til at opsamle data(temperatur). I denne opgave skal ADAFRUIT bruges til at måle temperatur. Den fungerer på den måde at der en master som gerne vil have noget data, masteren har så en adresse som den skriver til og det gør den ved at skrive til alle dens slaver, i dette eksempel har temperatursensoren adressen, vil den så svare tilbage og sige at adressen tilhører sensoren ved en ACK. For at begynde og læse fra slaven vil masteren sende en kommando til sensoren. Og igen vil den så svare ACK hvis adressen stadig er den samme som før.



# EEPROM

For at få mest ud af en temperaturføler skal vi bruge et sted til at gemme den data som bliver trukket ud af temperatursensoren. Her kommer 24AA64 som er en EEPROM(Electrically Erasable PROM), med den vil det være muligt for at der kan læses og skrives data på den omkring 100.000 gange. Samme som ADAFRUIT sensoren kan EEPROM både skrive(Write) og læse(Read) som den gør når masteren sender en kommando. Ellers gør den ikke noget af sig selv, altså en slave svarer masteren når den bliver spurgt om noget.

