Heartbeat – Rapport:

Denne oppgave handler om å lage et system med SD i fokus. I tillegg vil den oppgave lære oss hvordan lagringsplass fungerer og hvilke type vi har.

Kravspesifikasjonen er som følgende:

* Et system som kan generere et analogt outputsignal.
* Man skal kunne simulere en hjertefeil ved å trykke på en knapp.
* Systemet skal kunne simulere hjertesignalet og lagrer data på SD-kortet.
* Programmet skal kunne lese fra SD-kortet.
* Til slutt skal man kunne hente data fra SD-kortet og plotverdiene i regneark.

Det kan være nyttig med litt generell informasjonen rundt lagring, før vi går over til vår løsningen av oppgaven.

I arduinoen har vi 3 forskjellige typer memory. Vi har data memory, program flash memory, og EEPROM memory. Hoved forskjellen mellom dem ligger i både hastighet og størrelse.

**Program flash memory** er lagringsplass dedikert til programmer. De en read-only, og den er ganske raskt. Programmet vil bli lagret her for godt, så selv om man kobler ut strømmen vil programmet fortsatt være der.

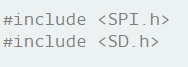
**Data memory** eller **SRAM** er øremerket variabler fra programmet. Den er også rask, og i tillegg er den volatile. Her blir alt som er lagret fjernet når man kobler ut strømmen. Lagringsplassen er riktignok stor. Denne kan man skrive til utallige ganger før den blir ubrukelig.

**EEPROM** virkelig treg i forhold til de to andre, og det vil ta riktig lang tid hvis man skal skrive noe til denne. Men fordelen er at det man skriver her vil bli lageret og ikke fjernet. I tillegg til den trege hastigheten, så finnes det en grense for hvor mange ganger man kan skrive til denne. Grensen ligger på rundt 100 000 ganger.

Det viktigste i oppgaven er å kunne lese/skrive fra/til SD-kortet. Hvordan bruker man SD-kort i arduinoen?

****Det er flere ting man trenger før man kan bruke SD-kortet. I hardwaredelen så trenger man en sd-kort adapter. Det finnes flere varianter, men vi har brukt den til høyre. I Software-delen så trenger man å bruke sd-kort biblioteket som man må legge til øverst i kode. Og i tillegg skal man bruke SPI biblioteket for at arduino skal kunne skrive til sd-kortet.

**SPI – Serial Peripheral Interface:**

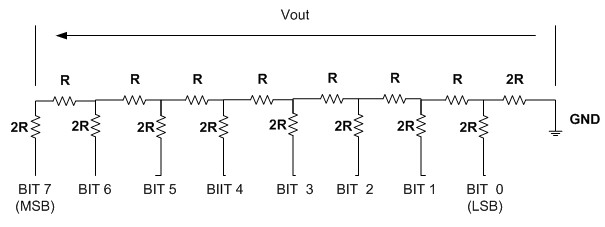
SPI er en kommunikasjonsmetode mellom microkontroller og små tilkoblingsenheter. Data overføringen er synchronous, som betyr at data overføringen mellom de to er synkronisert etter en klokke som microkontrolleren velger. Dersom overføringen hadde skjedd over serialport hadde overføringen vært asynchronous, som betyr at overføringen ikke er synkronisert etter en klokke. Og det kan jo bli et problem når du enheter prøver å kommunisere. En annen forskjell er at ved asynchronous overføring, så sendes dataen et byte av gangen, men ved synchronous så overføres det en del av gangen.

For å kunne kommunisere skal man koble mikroprosessoren og enheten på en spesifikk måte. Man skal bruke disse koblingene. Enhetene kobles etter en master og slave kobling. Der mikroprosessoren er master og enheten er slave. Mikroprosessoren er master fordi den bestemmer mye, som for eksempel klokken som overføringen skal synkroniseres etter.

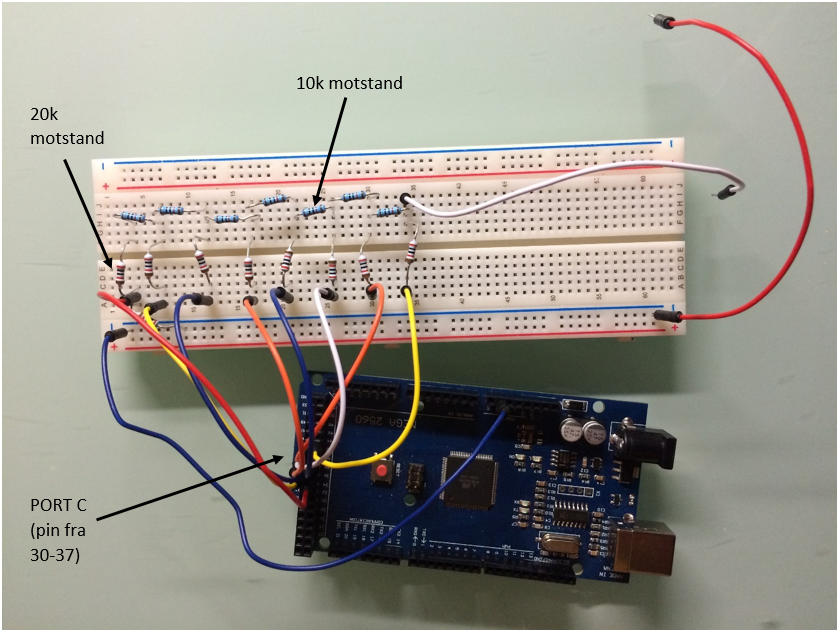
* MOSI – Master Out Slave In, denne brukes når mikroprosessoren vil overføre data til enheten.
* MISO – Master In Slave Out, Her er det tilkoblede enheten som overfører data til mikroprosessoren.
* CS - Chip Select, Siden en master kan kommunisere med flere slaves. Dersom den skal kommunisere med en bestemt, så velges denne av mikroprosessoren før det kan kommunisere.
* SCK – Dette er klokken kommunikasjonen synkroniseres etter. Klokken velges av master.

Nå har vi det vi trenger for å kunne bruke SD-kort, men for å kunne skrive ut et analogt signal trenger vi en DAC, digital to analog konverter. Vi har laget denne i en av tidligere oppgavene og har valgt å bruke samme design og konsept.

Vi har valgt å lage en R2R eller resistor ladder.Vi har løst oppgaven ved å lage en R-2R/resistor ladder digital to analog konverter. Den består av 8 pins, som igjen utgjør 8 bits. Hver pin kan enten være high eller low, 0 eller 5. Og ved hjelp av hvilken pin som er av og hvilken som er på, blir output strømmen påvirket. Siden strømmen går igjennom motstandene blir strømmen mindre. Output strømmen kan være alt mellom 0-5v. Hvor mye strømmen av et pin påvirker den totale strømmen avhenger av hvor i rekken pin befinner seg. 8 bit gir oss 256, det kan vi bruke til å regne ut verdier for hvilken som helst waveform.



På bildet under kan man se hvordan min R2R ser ut. Jeg har valgt å bruke 10k og 20k motstand. De hvite er 20k og blåe er 10k. Den hvite ledningen er output og rødt er ground. Disse kan kobles til for eksempel et oscilloskop. På bildet under kan man se settoppet vi har brukt.

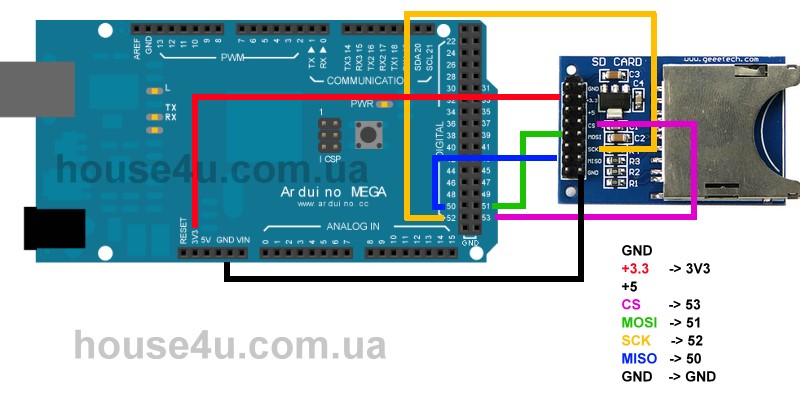


**Interrupt:**

Jeg har brukt interrupt i forbindelse en knapp som endrer over til hjertefeil. Interrupt virker ved at når man f.eks. trykker på en knapp, som så sender en signal til pinen, så brytes programmet ut av kodens flow. Programmet har blitt” interrupted”.

Hvis normal program flow er det blå, så er interrupt funksjonen det lilla. Så hver gang man sender et signal til en bestemt pin, vil programmet stoppe og kjøre den lilla koden. Når den lilla koden er ferdig, så vil den fortsette der den stoppet på den blå koden.

Vi har brukt dette tilkoblingsoversikten for å koble Arduinoen til SD-kort adapteren.



Løsning:

Vi har løst oppgaven på denne måten.

1. Først så slettes alt fra SD-kortet. Dette tenkte vi var viktig for å ikke overskrive eller skrive data som fortsettelse på noe som var der fra før.
2. Neste steg er å skrive data, eller som i dette tilfellet er tallene for et hjerteslag.
3. Videre skal disse tallen leses fra sd-kortet. Disse tallene lese og blir skrivet inn i en array, så tallene blir lagret.
4. Til slutt blir tallene som er i arrayen til output for PORT C, så vi kan se signalet på et osoliskop.

Dette er resultatet fra Excel. Vi har skrevet inn tallene og laget en graf.

Og dette er resultatet fra osoliskopet.



Koden: #include <SPI.h>

#include <SD.h>

#include "TimerOne.h"

File myFile;

unsigned int array[543] = {58 , 58 , 58 , 58 , 59 , 59 , 59 , 59 , 59 , 59 , 60 , 60 , 60 , 61 , 61 , 61 , 62 , 62 , 62 , 63 , 64 , 64 , 65 , 65 , 66 , 67 , 67 , 68 , 69 , 70 , 70 , 70 , 71 , 71 , 71 , 71 , 72 , 72 , 73 , 73 , 74 , 74 , 75 , 75 , 76 , 76 , 76 , 77 , 77 , 78 , 78 , 78 , 79 , 79 , 79 , 79 , 79 , 80 , 80 , 80 , 80 , 80 , 79 , 79 , 79 , 79 , 79 , 79 , 78 , 78 , 78 , 78 , 78 , 78 , 77 , 77 , 77 , 77 , 76 , 76 , 76 , 75 , 75 , 75 , 75 , 74 , 74 , 74 , 74 , 73 , 73 , 72 , 72 , 72 , 71 , 71 , 71 , 70 , 70 , 70 , 69 , 69 , 68 , 68 , 68 , 67 , 67 , 66 , 66 , 65 , 65 , 64 , 63 , 63 , 62 , 62 , 61 , 61 , 61 , 60 , 60 , 60 , 60 , 59 , 59 , 59 , 59 , 59 , 58 , 58 , 58 , 58 , 58 , 58 , 58 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 59 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 59 , 59 , 59 , 59 , 59 , 58 , 58 , 58 , 57 , 57 , 57 , 56 , 56 , 55 , 55 , 54 , 51 , 45 , 37 , 34 , 31 , 39 , 43 , 47 , 50 , 56 , 61 , 63 , 66 , 70 , 77 , 84 , 91 , 130 , 140 , 154 , 159 , 162 , 165 , 168 , 192 , 195 , 199 , 201 , 207 , 212 , 218 , 223 , 237 , 240 , 243 , 237 , 215 , 215 , 206 , 190 , 176 , 175 , 139 , 141 , 118 , 108 , 101 , 62 , 56 , 22 , 23 , 12 , 1 , 2 , 3 , 5 , 7 , 10 , 14 , 17 , 18 , 19 , 20 , 21 , 23 , 24 , 26 , 30 , 33 , 37 , 40 , 43 , 47 , 50 , 52 , 53 , 54 , 55 , 57 , 58 , 60 , 63 , 64 , 65 , 65 , 65 , 66 , 66 , 66 , 66 , 66 , 66 , 66 , 65 , 65 , 65 , 65 , 65 , 65 , 64 , 64 , 64 , 64 , 64 , 64 , 63 , 63 , 63 , 63 , 63 , 63 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 60 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 62 , 62 , 62 , 62 , 62 , 62 , 63 , 63 , 63 , 63 , 64 , 64 , 64 , 65 , 65 , 65 , 65 , 66 , 66 , 66 , 67 , 67 , 68 , 68 , 68 , 69 , 69 , 70 , 70 , 70 , 71 , 71 , 72 , 72 , 73 , 73 , 73 , 74 , 74 , 74 , 75 , 75 , 75 , 76 , 76 , 77 , 77 , 77 , 78 , 78 , 78 , 78 , 79 , 79 , 80 , 80 , 81 , 81 , 82 , 82 , 83 , 83 , 83 , 83 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 84 , 83 , 83 , 83 , 83 , 83 , 83 , 83 , 82 , 82 , 82 , 82 , 82 , 82 , 82 , 81 , 81 , 80 , 80 , 80 , 79 , 79 , 79 , 78 , 78 , 78 , 77 , 77 , 77 , 76 , 76 , 75 , 75 , 75 , 74 , 74 , 73 , 73 , 72 , 72 , 71 , 71 , 71 , 71 , 70 , 70 , 70 , 69 , 69 , 69 , 68 , 68 , 67 , 67 , 66 , 66 , 66 , 65 , 65 , 65 , 64 , 64 , 64 , 64 , 63 , 63 , 63 , 63 , 62 , 62 , 62 , 62 , 61 , 61 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 62 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 61 , 62 , 62 , 62 , 62 , 62 , 62 , 63 , 63};

int removeButton = 2;

bool readyState = false;

const int ISRbtn = 2;

int numberValues = 0;

const int maxValues = 1000;

int valueArray[maxValues];

int index = 0;

long debouncing\_time = 15; //Debouncing Time in Milliseconds

volatile unsigned long last\_micros;

void setup() {

// put your setup code here, to run once:

DDRC = 0b11111111;

pinMode(13, OUTPUT);

pinMode(removeButton, INPUT);

pinMode(ISRbtn, INPUT);

attachInterrupt(0, debounceInterrupt, HIGH); //Interrupt 0 is Digital Pin 2

Serial.begin(9600);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

Serial.print("Initializing SD card...");

if (!SD.begin(53)) {

Serial.println("initialization failed!");

return;

}

Serial.println("initialization done.");

removeFile();

sdWrite();

sdRead();

readyState = true;

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(500); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(500);

}

void loop() {

// put your main code here, to run repeatedly:

sdRead();

}

void sdWrite() {

myFile = SD.open("test.txt", FILE\_WRITE);

// if the file opened okay, write to it:

if (myFile) {

Serial.print("Writing to test.txt...");

for (int i = 0; i < 543; i++) {

myFile.println(array[i]);

Serial.println(array[i]);

}

// close the file:

myFile.close();

Serial.println("done.");

} else {

// if the file didn't open, print an error:

Serial.println("error opening test.txt");

}

}

void sdRead() {

if (!readyState) {

// re-open the file for reading:

myFile = SD.open("test.txt");

if (myFile) {

Serial.println("test.txt:");

// read from the file until there's nothing else in it:

Serial.println("reading from sdCard");

if (myFile) {

//for (index = 0; index <= numberValues; index++) {

while (myFile.available()) {

int input = myFile.parseInt();

valueArray[index] = input;

index++;

Serial.print("input:");

Serial.println(input);

Serial.print("index:");

Serial.println(index);

}

numberValues = index;

Serial.println(index);

myFile.close();

}

else {

Serial.println("error opening test.txt");// if the file didn't open, print an error:

}

}

}

Serial.print("numberValues");

Serial.println(numberValues);

if (readyState == true) {

for (int i = 0; i < numberValues - 1; i++) {

//PORTC = valueArray[i];

valueArray[i];

delayMicroseconds(5);

Serial.println(PORTC = valueArray[i]);

}

}

}

void removeFile() {

SD.remove("test.txt");

}

void HeartFail() {

Serial.println("HEART IS FAILING ");

PORTC = 128;

}

void debounceInterrupt() {

if((long)(micros() - last\_micros) >= debouncing\_time \* 1000) {

HeartFail();

last\_micros = micros();

}

}

**Konklusjonen:**

Vi har laget et system som kan lese fra og skrive til et sd-kort. I vår oppgaven så har vi skrevet tallene for et hjerteslag til sd-kortet og lest fra den igjen. Resultatet av systemet har vært glimrende. Vi har lykkes med oppgavene og alle delene i oppgaven fungerer. Vi har ikke støtt på noen store problemer i denne oppgaven, spesielt fordi vi hadde R2R DAC fra forrige oppgave. I tillegg så leste vi om SD-kort og satt oss virkelig godt inn i hvordan det fungerer før vi startet med oppgaven. Det vil jeg si er hoved grunnen for at vi ikke har møtt på store oppfordringer og problemer. Dette lærte vi i oppgaven om dac, da vi startet rett på det praktiske før vi i det hele tatt leste noe om det. Resultatet var at vi gikk et skritt frem og to tilbake. Men denne oppgaven har vært vellykket og vi har fått resultatet vi ønsket oss.

Kilder:

<http://www.geeetech.com/wiki/images/c/c4/IMG_0283.jpg>

http://house4u.com.ua/articles/solar-controller/en/part13.php