## Gekozen Opdracht

### Wij hebben gekozen voor: Dotmatrix, liftverdieping indicator

We hebben een lift applicatie gemaakt die omhoog en omlaag gaat en hierbij een animatie toont. De AVR zelf blijft natuurlijk gelukkig gewoon op zijn plek.

## Uitwerking

We zijn begonnen met de standaard uitwerking, die Diederich Kroeske gemaakt heeft. Deze code toont in de basis het aanzetten en aanzetten van de Matrix module.

Aanzetten:

twi\_init(); // Init TWI interface

// Init HT16K22.

twi\_start();

twi\_tx(0xE0); // Display I2C addres + R/W bit

twi\_tx(0x21); // Internal osc on

twi\_stop();

twi\_start();

twi\_tx(0xE0); // Display I2C address + R/W bit

twi\_tx(0xA0); // HT16K33 pins all output

twi\_stop();

twi\_start();

twi\_tx(0xE0); // Display I2C address + R/W bit

twi\_tx(0xE3); // Display Dimming 4/16 duty cycle

twi\_stop();

twi\_start();

twi\_tx(0xE0); // Display I2C address + R/W bit

twi\_tx(0x81); // Display OFF - Blink On

twi\_stop();

Hierin zien we het adres ‘0xE0’ steeds terugkomen. Op dit adres zijn een aantal registry entries beschikbaar die beschreven kunnen worden. We zetten in bovenstaand voorbeeld de ‘osc’ aan door de code ‘0x21’ door te sturen.

Vervolgens moeten we nog data sturen naar de module. Dit doen we weer via het ‘0xE0’ adres. Vervolgens geven we nog een adres mee, het row adres, en hierop zetten we dan een byte. Deze bevat 8 bits, die elk de aan/uit status van een LED bepalen. Combinaties hiervan geven ons de mogelijkheid om dan op een 8x8 matrix letters, cijfers, figuren, etc.. uit te beelden.

for (place; place >= desti; place--)

{

int x=place;

for (int adres = 0; adres <= 14; adres += 2)

{

twi\_start();

twi\_tx(0xE0); // Display I2C addres + R/W bit

twi\_tx(adres); // Address

twi\_tx(liftdown[x - 1].data); // data

twi\_stop();

x++;

}

wait(1250);

}

De data wordt uit een voorgeschreven byte array gehaald. Deze array bevat elke rij van de lift animatie. Deze array is nogal lang en is niet verwerkt in dit document, wel te vinden in het code uiteraard. Tijdens het scrollen wordt deze array geïtereerd en steeds een positie omhoog of omlaag bewogen in een loop om nieuwe data te krijgen die gedisplayd wordt.

Om de lift te laten bewegen wordt simpelweg PIN A uitgelezen om te kijken welke bits aan staan. Een opsomming van de codes:

Pin 1: Etage omhoog  
Pin 2: Etage omlaag  
Pin 3: Clear display  
Pin 4: Smile!

## Reflectie

Tijdens de opdracht zijn we vastgelopen op een aantal stappen, deze hadden te maken met de aansturing van PORTD die niet werkte, dit bleek te maken te hebben met de hardware configuratie. Dankzij Etienne is dit alsnog gevonden en hebben we de opdracht succesvol kunnen afronden. Ook zijn we te weinig in de documentatie gedoken, maar dit heeft ook vooral te maken mijn onze praktische aanleg.

Code is online ook terug te vinden op: <https://github.com/SuperJuiceh/MicroCont/tree/master/MicroContr/Eindopdracht>

## Extra Bilel

Besloten is om als extra opdracht de matrix te laten roteren. De gebruiker kan hiervoor de knoppen gebruiken:

Pin 5: Rotation + 90 graden.  
Pin 6: Rotation – 90 graden.

De rotaties kunnen per 90/180/270 graden gemaakt worden. Bij 0 en 360 gebeurt er natuurlijk niets.

De 8x8 matrix bits worden dan herrekend en gebruikt om te ‘tekenen’.