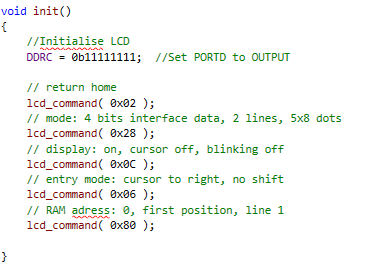
B1. Ontwerp een LCD ‘C module’ (dus een \*.c en een \*.h file, bijvoorbeeld lcd.c en lcd.h).

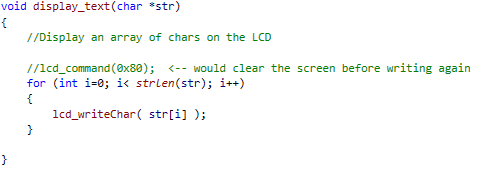
Het idee van deze opdracht is om generieke LCD functies te schrijven waarmee verschillende dingen gedaan kunnen worden met het LCD scherm. Men moest 3 functies (init, display\_text en setcursor) uitwerken.

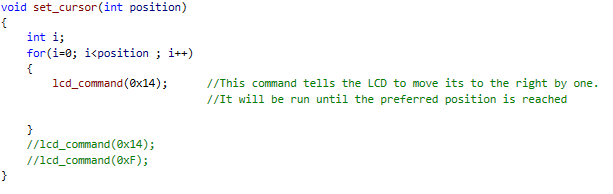
Bij onze oplossing hebben we gebruik gemaakt van de al gegeven methodes lcd\_writeChar en lcd \_command. De init methode maakt het LCD klaar voor gebruik. Deze methode haalt onder andere het scherm leeg en zet de cursor op positie 0,0.

Bij display\_text maken we gebruik van de lcd\_writeChar methode. Ook staat er (in comments) een mogelijkheid om eerst het scherm leeg te maken voor het schrijven. De functie roept per character in de array van characters de desbetreffende methode aan. Bij setcursor maken we gebruik van de lcd\_command methode. We gebruiken deze methode om de cursor positie op te schuiven naar een gewenste positie. De code om de cursor een plek naar rechts te doen bewegen is 0x14 (0001 0100 in binary). Deze methode met deze code wordt aangeroepen tot de aangegeven position is bereikt.

Hieronder samples van de code:







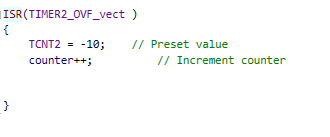
Link naar de repository:

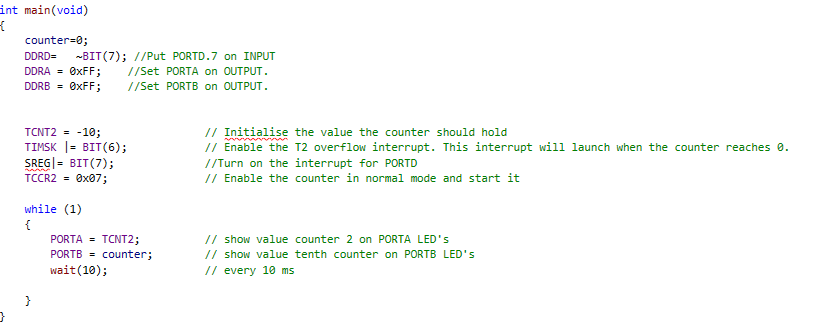
B2. Realiseer met AVR T/C-2 (counter mode) een ‘teller’ die het aantal keren ‘toets indrukken’ telt en weergeeft op het LCD (met de code uit B.1).

De opdracht is hier om een teller te maken toets indrukken moet bijhouden. Deze moet gebruik maken van de AVR T/C-2. Ook moet dezecode werken met de IR-sensor).

Bij onze oplossing was er een klein probleem bij het het gebruiken van de code van B2 en de LCD code van B1.Beide onderdelen werkte n gescheiden van elkaar maar in combinatie kregen we niks te zien op het scherm. Daarom hebben we er op moment van schrijven voor gekozen om de reacties te tonen op de LED’s van PortA en PORTB. De code initialiseerd als eerste PORTD.7 voor INPUT en PORTA en PORTB voor OUTPUT en wordt de counter value op 0 gezet. Vervolgens gaan we de teller initaliseren. Eerst geven we de teller een start value van -10 waarna we de teller instellen om te reageren op een overflow interrupt (zie code documentatie). Vervolgens zetten we interrupts aan voor PORTD en zetten we de teller aan. In de while methode die hierop volgt laten we de values zien op PORTA en PORTB. Hier zou je ook de LCD code kunnen aanroepen. Wat de applicatie moet doen bij deze interrupt is ook gedefinieerd. Deze moet de teller resetten en bijhouden hoe vaak de interrupt is aangeroepen. Deze code werkt ook met een IR-sensor als deze is aangesloten op PORTD.

Hieronder wat sample code:





Link naar de repository:

dd