

## Opgave 3

Opgave 3 er en direkte forsættelse af Opgave 1 (og Opgave 2). Formålet med opgave 3 er, at vi går ned på et lavere niveau og dermed frigør os lidt fra Arduino tankegangen. Arduino tankegangen er som bekendt helt ok, hvis man er programmør på amatør niveau, men for os professionelle SW udviklere halter den lidt.

Så derfor se på punkterne herunder i relation til jeres Opgave 1 (og Opgave 2).

- 1) I jeres program skal I lave om, således det bliver "forbudt" at bruge Arduinos Digital Pin nummerering og ligeledes forbudt at bruge de Arduino funktioner, der er koblet til Port manipulation på Arduinos Digital Pins. Det vil sige, at funktionerne **pinMode** og **digitalWrite** også er "forbudte". Tag udgangspunkt i linket her (linket er også medtaget i Opgave 1) : <https://docs.arduino.cc/hacking/hardware/PinMapping2560>
- 2) Erstat Ticker timer funktionaliteten med et eller flere "rigtige" Timer Interrupt(s). Og erstat det eksterne Interrupt (attachInterrupt) ligeledes med et "rigtigt" eksternt Interrupt. Sluttelig hvis man har mod på det, så lav også kode til et "rigtigt" UART Interrupt. I alle tilfælde kan der være god hjælp at hente i koden fra filerne **avr8-stub.h** og **avr8-stub.c** som er placeret i avr-debugger direktoriet under **libdeps** i jeres projekter. Jeg vil også lægge nogle (små) Microchip Studio projekter op i Moodle læringsrummet, som I kan bruge som udgangspunkt for jeres arbejde i VSCode. Nærmere info om placeringen - og indholdet af disse projekter følger.
- 3) Implementer evt. en C++ klasse i jeres projekt og ét eller flere objekter af denne klasse. Det kunne f.eks. være en klasse, der håndterer port håndtering. I klassen kunne der være specificeret felter som: Port Adresse og Port Value. Eventuelt kunne der også være et array af 8 stk. booleans. Dette array kan så hurtigt give et overblik over hvilke bits i porten, der er høje/lave, idet man kan antage at true = høj og false = lav for en specifik port bit. Der skal også være funktioner i klassen, der kan gå ind og manipulere på de nævnte variable her.
- 4) Lav et program i f.eks. Python, et C# WPF-program eller andet, der kan kommunikere med jeres AtMega2560 board via Seriel PORT/UART.
  - I programmet skal/kan der være mulighed for at have simulerede lysdioder tilkoblet (Rød, Gul og Grøn lysdioder). Disse lysdioder skal så være tændte/slukkede synkroniseret med, hvad de tilsvarende lysdioder er på jeres AtMega2560 board.
  - I programmet skal der være mulighed for at have et "spejl" af alle Porte på jeres AtMega2560 board. I spejlet skal det være muligt at se tilstanden på alle bits i alle porte. Det vil sige, om en port bit er høj eller lav. Ligeledes skal det være muligt at ændre tilstanden på alle bits på alle porte. Altså mulighed for at ændre fra høj til lav og fra lav til høj på en vilkårlig Port bit.

Opgave 3 her bliver den opgave, sammenholdt med Opgave 1 (og Opgave 2) bliver den gennemgående opgave her på IOT-modulet. Jeg regner med, at I skal aflevere opgaven her sidste dag I har IOT (d. 12/12-2023).