Opgave / Case

Hardware: STM32F746G-DISCO + Grove-starter Kit
Software: C/C++ compiler som f.eks. mbed (*eller lignende).
*Kræver selvstændigt arbejde – fokus i undervisningen er på mbed

Case:

Den Embedded controller skal bruges til at måle rum temperatur, lys og lyd i det lokale, den er opsat i. Data skal senere sendes til en database (TCP/IP kommunikation i Embedded III).

Indtastning:

1. Angiv en kode for bygningen.
2. Angiv en kode for lokalet.

Ved opstart af programmet skal begge koder indtastes, før programmet forsætter.

Note: Brug seriel kommunikation til at skrive disse værdier.

Avanceret/Ekspert: Brug Touch screen til at indtaste værdierne.

Display funktioner:

1. Vis hvilken kode, der er angivet for bygning og lokale, f.eks. bygning 8 lokale 27
2. Vis temperatur i °C eller Fahrenheit
3. Hvis det er lyst, skriv: Goddag ellers godnat
4. Ved høje lyde, skriv: Advarsel støj

De fire forskellige værdier skal vises samtidigt uden ”flikker”.

Andre funktioner:

1. Knap med toggle funktion: Første tryk = Temperatur vises i fahrenheit, andet = Temperatur vises i celsius (F = °C * 1,8 + 32)
2. *Seriel kommunikation af måleværdier:
    a. Ved tastning af 1, skal hex værdien for temperaturen returneres
    b. 2 = Værdien for lys
    c. 3 = Værdien for lyd

*Denne kommunikation er forberedelse på Embedded III – kommunikation via TCP/IP, hvor enheden sender alle data til en database.

Dokumentation:

1. Kommentarer i selve programmet, der beskriver program sektioner (hvor det giver mening). Start gerne i Main med en generel beskrivelse af dit program.
2. Ekstern tekstfil med en kort beskrivelse af programmets virkemåde i sin helhed. Det skal være mulig at tilslutte sensorerne, så programmet virker korrekt uden at se i koden.
3. Startdato og navn på udvikler
4. Log over væsentlige ændringer / problemer / fremtidige ændringer.
5. Husk at opdatere programdokumentationen løbende. Den endelige dokumentation skal beskrive projektet i sin helhed

Forventninger til programmet:

1. Minimum af kode i Main. Koden skal være opdelt i separate funktioner. De forskellige målinger og display funktionen skal kunne kører uafhængigt af hinanden (trådet programmering).
2. Tænk over måling af lyd: Hvad er støj? – hvornår skal der skrives en advarsel?
3. Det færdig programmet optimeres og struktureres, så der bruges minimum af hukommelse.
    a. Bruge pointers / adresser i programmet, hvor det giver mening.
    b. Hvis den samme stump kode gentages flere steder i programmet, skal den erstattes af funktioner.
    c. Eventuelt brug af header fil (start på eget bibliotek)

Avanceret:

1. Brug af header, klasser og objekter.
2. Mulighed for at se den laveste/højeste værdi eller et gennemsnit af de sidste 10 værdier, der er målt (lys, lyd og temp).
3. Overvejelser omkring kvaliteten af målingerne (støj, sampling rates)
4. Enheder kobles sammen via I2C – master/slave princip (brug gerne Grove-LCD RGB Backlight)
5. Temperaturen kan vises korrekt uanset indstillingen af spænding på ”Base Shield”
6. Arbejde med et simpelt dato/tidsstempel eller en kode for tidszone

Eller andre avanceret ting – snak med din lærer omkring mulighederne.

Ekspert:

1. Mere avanceret måling af lyd – brug eventuelt Nyquist teori for sampling af data.
2. Grafisk visning af kurve eller bargraf på display
3. Brug af et mere avanceret tidsstempel – evt. real time.
4. Arbejde med visning af lys/lyd med passende enheder (lux/dB - i stedet for deres analog værdi).
5. Kreerer eget ”bibliotek”

Eller andre funktioner ud over standard opgaven.

Eventuelt kan Raspberry PI f.eks. anvendes til dataopsamling.

Links

Serial Communications

https://os.mbed.com/handbook/Serial

C++ Tutorial

https://www.programiz.com/cpp-programming

C++ Pointers

https://www.programiz.com/cpp-programming/pointers

Sampling af data

https://www.embedded.com/design/prototyping-and-development/4024581/Sampling-rates-for-analog-sensors

Eventuelt:

C programming turtorial

http://fresh2refresh.com/c-programming/

Tutorialspoint:

https://www.tutorialspoint.com/cprogramming/index.htm

https://www.tutorialspoint.com/c_standard_library/