Mobilt EEG

Detta är en projektplanering för fortsättningsarbete gällande mobilt EEG (elektroencefalografi). Detta är ett exjobb som tidigare utförts av Christoffer Olsson samt John Croft och handleddes av Sakib Sistek. Tanken är att fortsätta och fullborda ett fungerande mobilt EEG, samt eventuellt lägga till extra funktionalitet.

Rekommenderas, om inte ett måste, att det är blandat mellan data och elektro, då det är stora fokus på både mjukvara och hårdvara, något som bara en grupp från bara antingen data eller elektro antagligen inte har så bra koll på.

Vad ett EEG är och hur det fungerar kan man lätt kolla upp via en snabb visit till Wikipedia.

Huvudtanken med projektet är att skapa ett fristående program och hårdvara för att skanna hjärnsignaler som i princip samma nivå som ett vanligt EEG. Då mycket fick göras från grunden så hann exjobbet inte klart med att uppfylla projektets mål.

Däremot kan det ges tips om hur man kan fortsätta, vad som behöver göras, hur man kan göra vissa saker osv.

(Märk väl att det finns ingen som helst uppgift att tolka signaler/EEG-data, utan försök endast att uppnå att signaler liknar EEG signaler, det finns bibliotek online som man kan kolla hur riktig EEG-data ser ut, och så länge programmet ger resultat som kan uppfattas av t.ex. EDFBrowser så bör allt fungera som det ska).

Följande är tips på lösningar och struktur på arbete (Från Christoffer och John).
Tips Mjukvara:

- Exjobbet använde *Python*, vilket var ett stort misstag när det kommer till realtidsdelen av kommunikation mellan hårdvara. Använd *C* istället.

- Det finns ett färdigt och komplett bibliotek för att hantera EDF (European Data Format) formatet (vanligaste formatet för EEG data) som heter *edflib* som är enkelt att implementera och använda (Det finns en python-version av edflib vid namn Pyedflib som användes för exjobbet).

- Gällande GUI så finns det fria händer, tänk bara på att man kommer få bygga ett interagerbart program för att t.ex. starta och stoppa inspelning, samt extra funktionalitet som annotering. Det finns även en mängd utveckling man kan göra i detta området, som att lägga in en EDFspelare samtidigt som videon spelas (Embedded).

- För att kolla på EEG-data så finns det ett program vid namn *EDFBrowser* som både kan hantera EEG-data, samt konvertera mellan olika format och som man kan använda för att kontrollera att EDF-formatet fungerar.

- Uppspelning kan antingen hanteras direkt på enheten, med inkopplad skärm, eller via en dator eller något annat. Det är fria händer här. Det pythonbaserade projektet har uppspelning direkt via GUIt på hårdvaran.

Tips Hårdvara:

- Börja med realtidslänken mellan eventuell ADC (eller annan motsvarande system) och styrkrets. Det är lätt att underskatta mängden data som måste hanteras; vanliga styrsystem som studenter redan kan vara bekanta med (som Arduino, PIC16, RasPi osv...) uppfyller med all sannolikhet inte kriterierna för hastighet, stabilitet, kommunikationskanaler, minne osv…
    - Använd ett realtidssystem; Raspberry Pi och andra fullfjädrade datorer med OS och kernel är **INTE** realtidssytem. Om systemspecifikationen kräver att man ska sampla med en viss frekvens (t.ex. 250S/s) så har man en väldigt kort tid där varje sample måste ske, och då behöver man exakt kunna styra när man ska exekvera sin samplingssubrutin. Använd istället en snabbare mikrokontroller eller eventuellt en FPGA (med denna lösning tappar man dock möjligheten att tillämpa befintilig kommunikationshårdvara, man får designa eget! Detta kan vara en fördel om samplingsutrustningen använder en icke-standard kommunikationsprotokoll).

- Det skall betonas att systemet ska vara batteridrivet, då man inte under **några** omständigheter ska använda elektroder som är kopplade till ett nätanslutet system (även om man tror att den är isolerad!).

- En ADC som rekommenderas att använda är TI:s ADS1299-x, en 4- till 8-kanals integrerad krets som är speciellt lämpad för biosignaler (i synnerhet EEG, som kräver hög precision och goda brusegenskaper). Den används i bl.a. OpenBCI-systemet. Det är denna som mycket av exjobbets arbete baserats på och det finns några prototyper på hur ett kopplingsschema och kretskort skulle kunna se ut.
    - Som utgångspunkt för samplingssytemet finns ett s.k. *Evaluation kit* from Microchip med exempel på kompletterande kringutrustning, kretskortupplägg och styrmjukvara.

- Den fysiska hårdvaran kommer att sitta på någon form av huvudbonad, dock så kan en del potentiellt sitta vid midjan eller liknande, så länge sladdarna räcker. Poängen här är att skapa en 3D-designad huvudbonad som kommer att 3D-printas här på Chalmers, för att kunna eventuellt placeras på en patients huvud och sedan börja spela in EEG-signaler med hjälp av hårdvaran och mjukvaran av produkten.

Sakib Sistek
sistek@chalmers.se