Month März

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2 Meeting	3	4
5	6	7	8	9 Meeting	10	11
12	13	14	15	16 - "Mini" Meeting	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Month April

Mon	Tue	Wed	Thu	Fri	Sat	Sun
						1
2	3	4	5	6	7	8
9	10					15
16						22
23	24					29
30						

Month Mai

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7						
14						
21						
28	29	30	31			

Month Juni

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1		
				8		
				15		
			Due date	22		
				29	30	

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Integrated Systems Laboratory

Group project at the
Department of Information Technology and
Electrical Engineering

for

Menachery Kiran, Hombach Paula, Scherer Moritz

Design and Implementation of a long-lifetime miniaturize smart hearing aid for stutterers.

Advisors: Michele Magno

Professor: Prof. Dr. Luca Benini

Handout Date: 26.02.2018

Due Date: 21.06.2018

1 Project Goals

The main goal of this project is to develop a wearable devices with smart capability to help stutterers. The system will be design using the best sensor(s) and actuator to give a feedback to the user. The main goal of this project is develop and test the hardware-software system in a real application scenario. Both lab and in-field test are planned to evaluate the system developed in terms of performance (ie. Accuracy, latency,etc) and lifetime.

2 Tasks

The project will be split up three phases, as described below:

Phase 1 (4-6 Months)

1. An important investigation of the state-of-the-art off-the-shelf components, processing and sensing will be the first step of this project. The student together with the supervisors will decide the main components according of the power consumption, the functionality and the availability on the market.)
2. The first preliminary algorithm and sensors interface will be studied and implemented to cover the stringent requirements of the wearable device and application scenario. This part can be done using demobaord of the selected hardware components(microcontroller, sensors, speakers)
3. Preparation of the measurements set up and preliminary lab test of the developed algorithm implemented in the hardware platform.
4. Preliminary schematic and layout of the first version of the hardware selected.

→ Start design PCB

Phase 2 (4-5Weeks)

1. Testing, measurements and eventually simulations for estimation of performance of the developed system.
2. In-field evaluation of the developed solution.

3. Optimization of the algorithms and sensors interface to improve the performance

Phase 3 (1-2 Weeks)

1. Finalizing the tests and optimizations.
2. Write final document and prepare presentation.

3 Milestones

By the end of **Phase 1** the following should be completed:

- Preliminary control system that include the sensor acquisition and the algorithm
- Preliminary schematics, layout and eventual PCB ready.

By the end of **Phase 2** the following should be completed:

- Accurate in-field test and performance evaluation
- A working version of the control system in the application scenario.

By the end of **Phase 3** the following should be completed:

- Fianl version and working demo.
- Final Presentation
- Final Report, including final results.

.

4 Project Organization

4.1 Weekly Report

There will be a weekly report sent by the student at the end of every week. The main purpose of this report is to document the project's progress and should be used by the student as a way to communicate any problems that arise during the week. The report, along with all other relevant documents (source code, datasheets, papers, etc), should be uploaded regularly to remote repository.

4.2 Final Report

PDF copy. All copies remain the property of the Integrated Systems Laboratory. A copy of the developed software needs to be handed in on CD or DVD at the end of the project.

4.3 Final Presentation

At the end of the project, the outcome of the thesis will be presented in a 15 minutes task, again during a group meeting of the Integrated Systems Laboratory.

9.3.18

- [x] Presentation: before or after due date? → up to us, important is the report
- Skizze Moritz: (about phase 1)
  - Analoger part: Amplifier, Pre- and Postfilter (Kiran)
  - ADC and DAC (Paula)
    - ADC: im Bord
    - DAC: outside → buy! → depends on speaker PWM used
  - Algorithm (Moritz)

- [x] Can detector output an interrupt signal? → micro controller, which can be tuned
- [ ] Information about "Chor-Effect"
- 1 Mal pro Woche treffen & besprechen
- new Version of detector paper (detector still in work)
- detector output can't be used as data output
- first stage: get data, test components
- store data: accelerator output
  - ↳ find algorithm
- DAC and output signals not the most important
- Solder station: ask Hansjörg (he instructs too)
- for accelerator an amplification is needed
  - ↳ if needed, ask Michele
  - ↳ Filter because of Aliasing

→ try microphons and input out and accumulate data

→ try Apollo out

- not everything is possible with microcontroller
- meeting 15' to 30' to see progress
  - ↳ E-Mail when there are questions

- [ ] Amplifier
- [ ] Draht anlöten

- 2 Wochen für Daten → Microphon testen & ADC
  - ↳ in Apollo
  - ↳ Anschlüsse finden

Paula

- [ ] ADC auf Apollo testen & Verstehen
- [ ] 1st: Spitzenspannung Input ADC → Kiran

Deadline 21.03

Kiran

- [ ] Amplifier für Input
  - ↳ bis zu ~20kHz

Deadlines: Amplifier 16.03.
Filter 21.03.

- [ ] Schauen, was es schon gibt (AIC/EC-Notes) auch auf Markt verfügbar
- [ ] Lötstation & Jumper-Kabel

Moritz

- [ ] Mikrophon & Accelerator Testen
- [ ] Löten von Accelerator (mit Hansjörg reden)

Deadlines: 21.03.

- [ ] Algorithmus weitertreiben
- [ ] RC-Filter für Output informieren
- [ ] Informationen Chor-Effekt suchen & weiterleiten
- [ ] Github-Infos schicken

16.03.18

- Moritz:
  - FFT / IFFT: 2 Verfahren für variable Länge
  - Hanningfenster (eher simpel): Overlap add
  - Blackman-fenster: Overlap add
  - Löten (Kontakte an Kehlkopfmikros)

  } Fensterfunktion: Abtasten nicht periodischer Signale verbessern → Kohärentes Abtasten

  - ☐ Filter schreiben
  - ☐ Ordentlicher Aufbau von Testmikrophon

☐ Blasenpflaster / Plastiksack für Mikrophon

- Kiran:
  - Verstärker gefunden (LT1996)
    - ↳ Gain einstellbar
    - ↳ Vorfilter included
    - ↳ Problem: Effizienz da Angabe nur max. 1W
    - ↳ Michele aproved
      - ↳ Idee Amplifier von Bird-paper
  - ☐ Abklären ob Amplifier von Michele oder bestellen
- Paula:
  - Infos zu Apollo 2
  - ☐ Interface zum Programmieren

- ☐ ADC bis aller spätestens 23.03 → Paula
- ☐ Output auf Pin und mittels UART (oder so) auf PC geben → Kiran
- ☐ Programmieren Filter: Tiefpass & Frequenzschiebefilter → Moritz

} Deadline 23.03.18

- Git Hub:
  - Source Tree → downloaden → bei Fragen an Moritz
  - "Alles vormerken" → hochladen mit Notiz, was passiert ist "Comit"
  - mit Push ins Internet laden
  - Möglichkeit auch PDFs hochzuladen

- Interface Apollo:
  - µvision 5, ARM
  - Apollo 2 ambig Seite

- Es gibt Beispielprojekte zum Anschauen & Lernen

- ☐ Meeting mit Michele nächsten Freitag → Kiran

How to Git:

- 2 Philosophien:
  - Merge
  - Rebase

- Workflow:

Anfordern / Pull

1) Anfordern → von überall
2) Bei Branches Pfeile nach unten
   → drauf gehen und pull
   ⚠ Immer zuerst pull vor dem Arbeiten

Commit

3) Arbeiten auf richtigem Branch
   immer wieder commiten, da man dann Veränderungen sieht
   (file erstellen, Aufbau, TODOs, ...)

Rebase

4) Bis jetzt noch kein Push ⚠
   ~~zu master wechseln~~
   zu Algorithmus (mein Branch)
   → Änderungen auf master Abbilden (Rebase)
   Änderung vom master bei uns integriert

Push

5) Pushen (Algorithmus & master auf einer Linie)
   ↳ unser Arbeitszweig

Merge

6) Funktion von anderem Branch übernehmen → Merge
   gehe zu master, Rechtsklick auf Branch, Merge von Algo
   in den aktuellen Branch
   dann Push

Allgemein

7) branch → PC-Kopie
   origin/branch → online-Kopie

Rebase

8) immer Rebasen wenn hinter master
   ↳ dann möglich zu pushen

⇒ Arbeiten auf einem Branch → master vorraus, wenn andere Änderungen wollen dann auf master tun (merge), wenn man Änderungen haben möchte dann rebase

bevor man etwas macht: Anfordern (ausser bei commiten)

23.03.18 Protocol

- [x] How to program → questions to whom?
- [x] Better thing available as Keil? IAR workbanch
- [ ] Get UART to USB cabel from Michele
- [ ] Get tutorials from Michele
- [ ] Look if DAC needed or RC for output
- [ ] Get amplifier from Michele
- [ ] Look for PCB

- Keil doesn't support UART in debug
- with SWO it works
- For Ambiq Keil is good, the proplem is Ambiq
- For debug UART to USB works nice
- Michele gives access to tutorials about microcontrollers
  ↳ polybox
- Ambiq with bluetooth arrived
  → try it out for later (bluetooth is secondary)
- Output driver, we need DAC
  normally done with PWM (normally already in speakers)
  → DAC probably not needed
  → perhabs RC filter needed
- Amplifier: basic amplifier (see paper)
  tune resistance for gain
  → Michele gives us amplifier
- test also ordered device → get demoboard / breakoutboard for it → ask Hansjörg
- Possible to start PCB (ask Alfonso)
- GET DATA
- Michele gives UART to USB cabel

- [x] Breakoutboard besorgen
- [ ] DebugOutput vom ADC
- [ ] ADC mit Oszi. testen
- [ ] Samplerate & Fenstergrösse
- [ ] Collect data

Compiler Probleme lösen

- Errors weil files / libraries nicht gefunden:
  1) Options for Target (Zauberstab & Hierarchie)
     - richtiger compiler wählen
     - C/C++: Include paths → anpassen
  2) Files → Options → path anpassen

- Schreibfehler normalerweise nicht angezeigt
  → selber prüfen