Proving CANDO

Leo Freitas

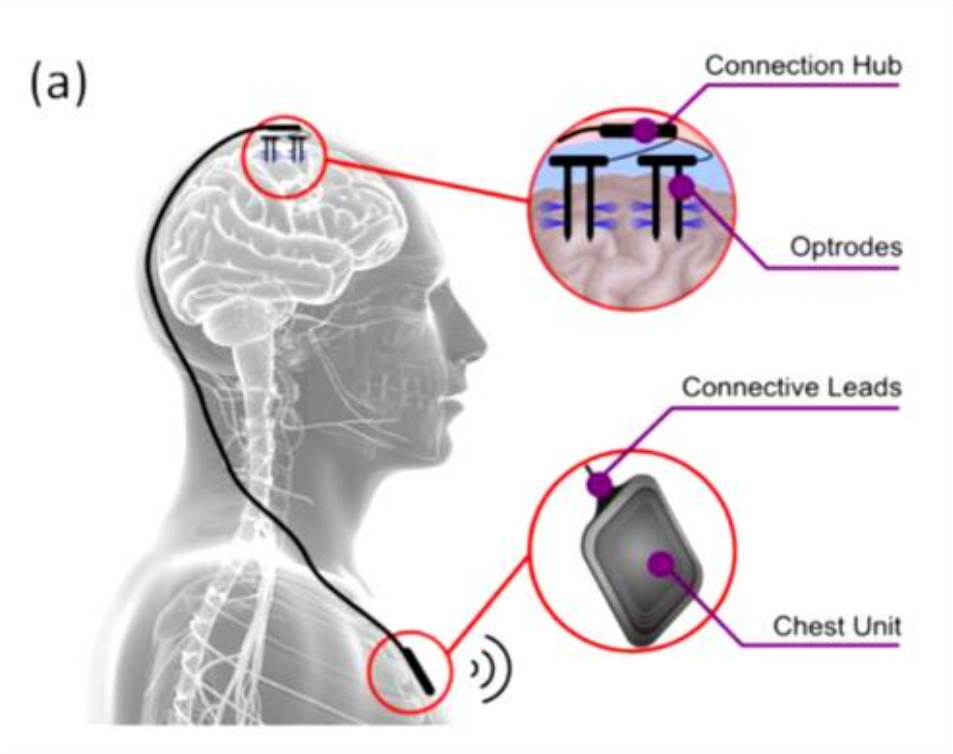
Overture 11/06/2025, Aarhus

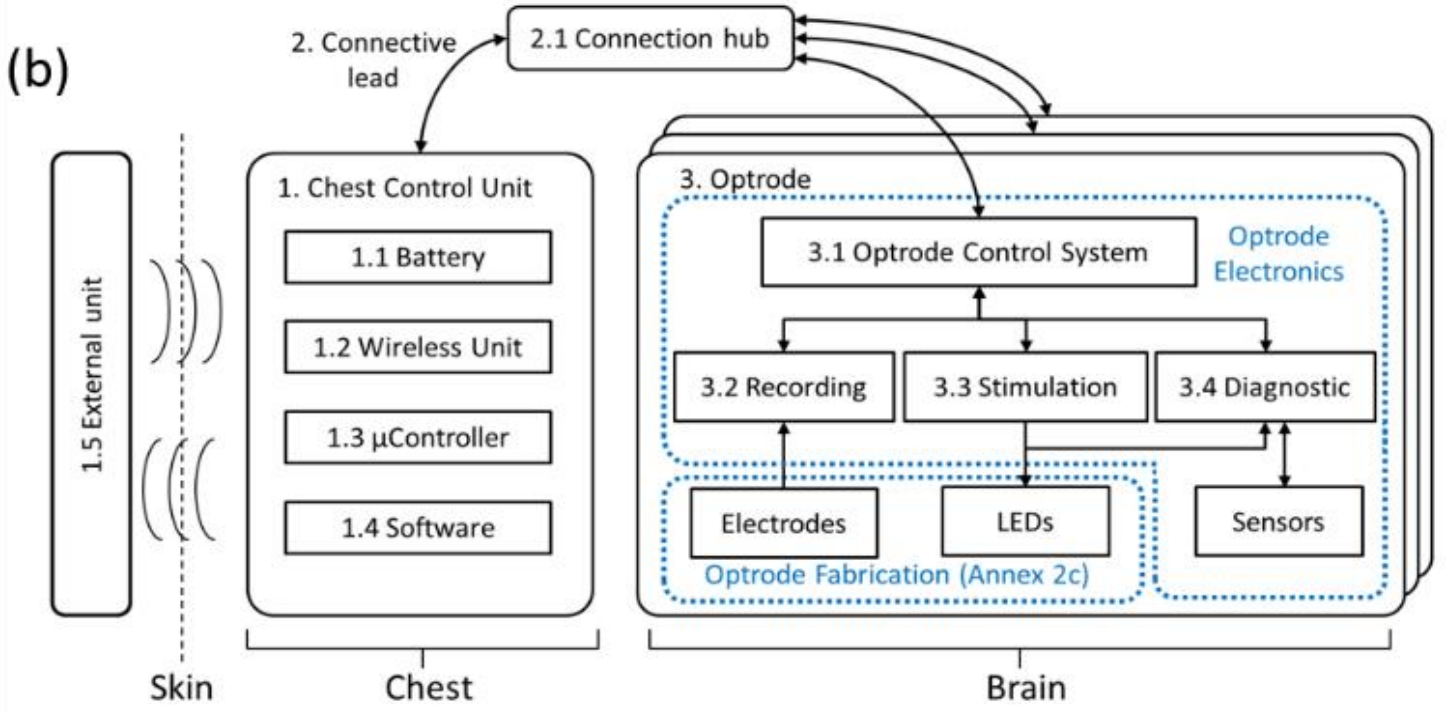
In collaboration with :

- Ben Wooding PhD Student
- Patrick Degenaar Brain pacemaker eng. (NCL) lead

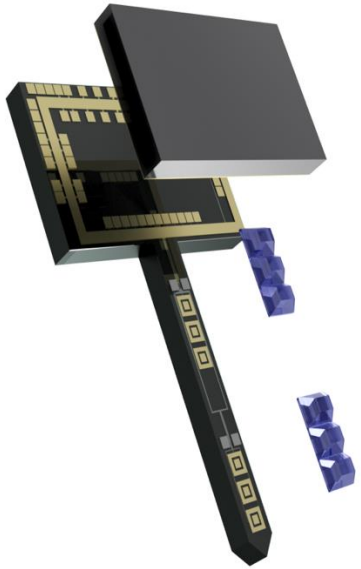
Brain Pacemaker (mid point adoption)

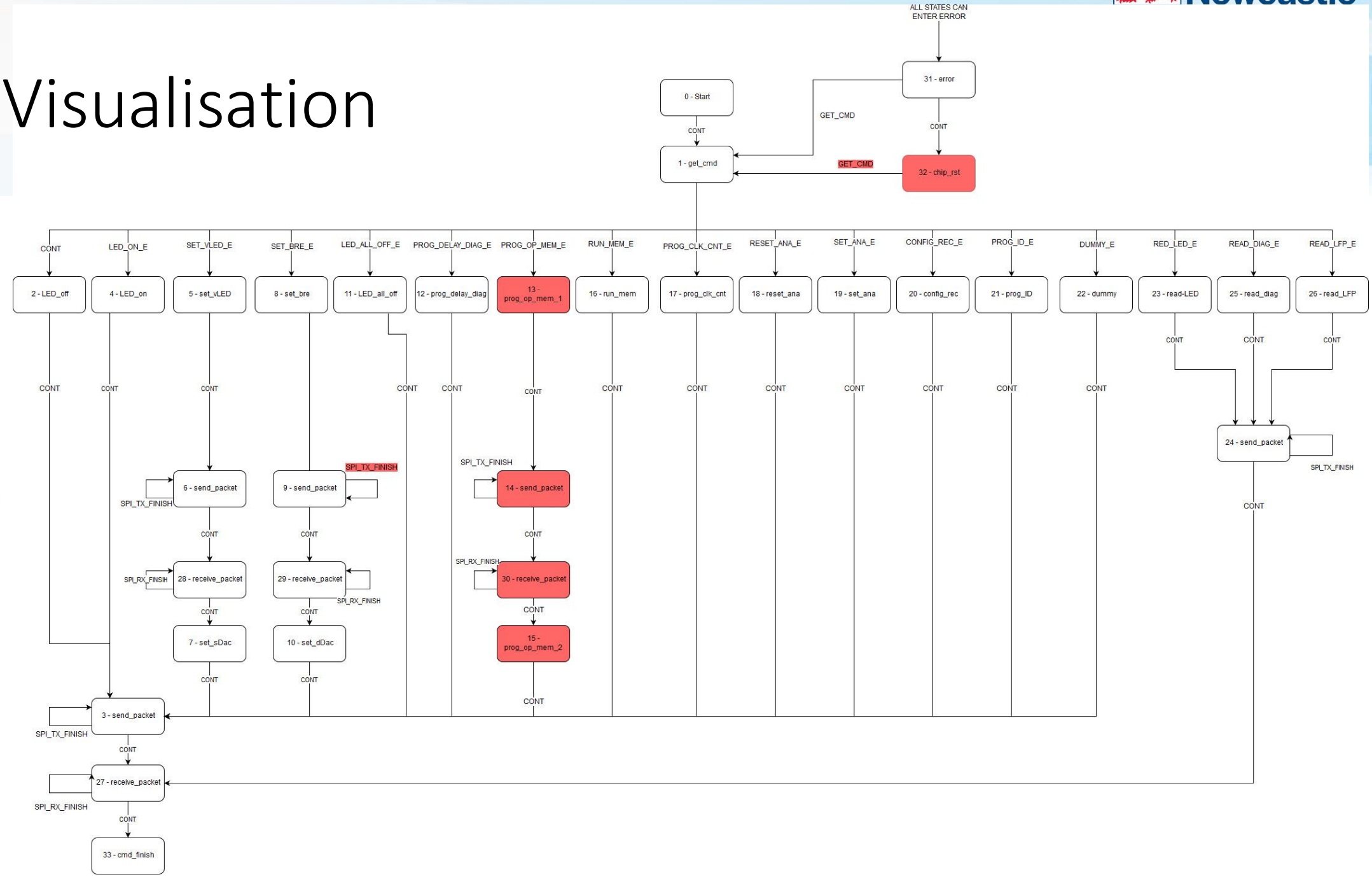
- Optogenetic stimulation (i.e. electrical signal input; light signal output)
- Embedded software within the controller unit for closed-loop stimulation
- External software for wireless download diagnostic information and upload control parameters





FSM Visualisation





What's been verified

- CANDO VDM verification went as far as POG
  - QC didn't exist at the time of modelling (2017/18)
  - Retrospective QC failures corroborated by failed Isabelle proofs
  - **Better to have QC counter examples in secs than hours of Isabelle frustration!**

- Motivation for the VDM to Isabelle translator
  - Translation strategy developed since 2010 as part of AI4FM project
  - UG / PG taught module on applying translation manually (2012-2022)
  - Isabelle translator implementation (2020-)
  - Improvement of translation strategy (still some constructs are missing)

What's been verified?

- Manual translation of CANDO FSM and its (136 [2018/9 POG]) POs
  - ~96% of POs discovered automatically (e.g. Isabelle's sledgehammer tool)
  - Higher automation requires a specific "VDM modelling style" (a.k.a. exu)
  - Proofs were mostly similar, 1-3 lines on average
  - 72 lemmas and expertise needed for initialisation and state invariant POs
  - POG varied since MSc work (i.e. 272 POs [2025] x 136 POs [2018/9]).

- Proof of initialisation / state invariant
  - Required about 34 auxiliary lemmas for a single proof!
  - Overall 5 proofs took most (90+%) of the (2 weeks) verification effort
  - Smooth verification due to adequate VDM style for Isabelle proofs.

Results Overview

- Translator now applied to various models of different complexity
  - Geneal infrastructure (e.g. now also translates VDM to Dafny)!

- Translation infrastructure as VDMJ and VSCode plugins:
  - exu: VDM style checker for ease of proof
  - exu can suggest/enforce style choices (~100 of unpacking/rewrite rules), e.g.

```
types
    Okay  : set of nat inv s == forall x in set s & x < 10;

    Value : nat         inv v == v < 10;
    --@Witness({1,2,3})
    Better: set of Value;
```

Results Overview

- VDM to Isabelle
  - VDM math toolkit 1605LOC (406 lemmas, 11 theorems, 84 unpacking rules)
  - FSM: VDM (1053LOC); Isabelle (1456LOC)
  - FSM POs proofs: 820LOCs; 72 auxiliary lemmas needed
  - Not all proofs completed (i.e. many are to be repeated with slight variation)

- VDM Toolkit (https://github.com/leouk/VDM_Toolkit)
  - Numerous examples, experiments and demos
  - Let's have a look

Conclusion

- Translation infrastructure as VDMJ and VSCode plugins:
  - vdm2isa:
    - Model organizer (declaration before use, dependency checks, etc.)
    - Model translator (VDM to Isabelle strategy with 3 flavours)
    - Configurable proof strategy generator (i.e. 4 different target model flavours)
    - **Unearthed ~50 VDM AST / TC / POG / VSCode issues**
  - VDMJ (more developed), as well as VSCode (less developed) plugins
- VDM mathematical toolkit in Isabelle
  - Community resource for VDM proofs in Isabelle/HOL.