# Advanced BLE-CELL Introduction

Q: What I learned
A: Block Diagrams are used in hw & sw engineering
B: Chris chose to use the following components & reasons
-raspberry header = for GPIO + HAT + solar power + easy access
-expansion board = for fine pitch connector +
-2 pin JST = battery (remember particle board demo)
-battery regulator = for lower power management
-3.8v = for celluar
-BG/EG95 = CAT-M1 makes it flexible
-2.4GHz = antenna for cell
-I2C = mic
-USB C 2.0 = connector
-SIM CARD = mechanical
-nrf52/53 = flexible pins + power management + SW development

## Footprint considerations for the advanced components

Q: What I learned
A: PCB delivery time + cost + reading the mechanical & circuit layout
B: Blind & Buried Vias
-For nrf52 the top layer has Blind Vias on the top layer but not on the lower ones
-Buried Via goes through all the layers, folks use this for other purposes
C: Blind & Buried work best for High Density Interconnects + tight tolerance -
> thus driving costs up by 4-5x
D: HDI boards are great tight tolerance & signal integrity
E: Electronics is a heterogenous system that requires a decision tree approach & mindset

## First look at the nRF52840

Q: What I learned

- A: Why embedded? (ie. Why not just use the Pi?)
  - Waiting on a Linux(python/raspberry pi) isn't real-time bc it has to compile
  - One needs low-level deterministic programming in order to respond RT
- Why Bluetooth?
  - nrf52 BLE stack is super developer friendly -> BLE is SW stack that communicates to other radios. Nordic uses SoftDevices (one core does some hmwk & one core does other stuff)
  - demand for BLE is found in high volume & high demand products
  - Celluar is power hungry & expensive
  - WIFI is power hungry & security risk (no in high demand)
- Why this particular chip?
  - Low power modes for the embedded breakout portion of the module
  - Flexible GPIO and fabrice
  - Powerful Cortex M4 processor
  - Wide range of RF software stacks (Bluetooth, Thread, Zigbee, 802.15.4) for different communication
  - Bundled “soft radios” that are implemented without requiring low level control from the application side of things.

# First look at the EG91 / BG95

Q: What I learned
A: Difference between Module & Chip

- module (mini-system) - all the antennae stuff is figured out
- chip (sub system) - gotta add stuff like passive components
- cost effectiveness of modules enables rapid development vs trying out different chip combos can take time & energy

B: UFL

- easy way to connect to antennae

C: Pin compatible

- if you had two modules to them they should work

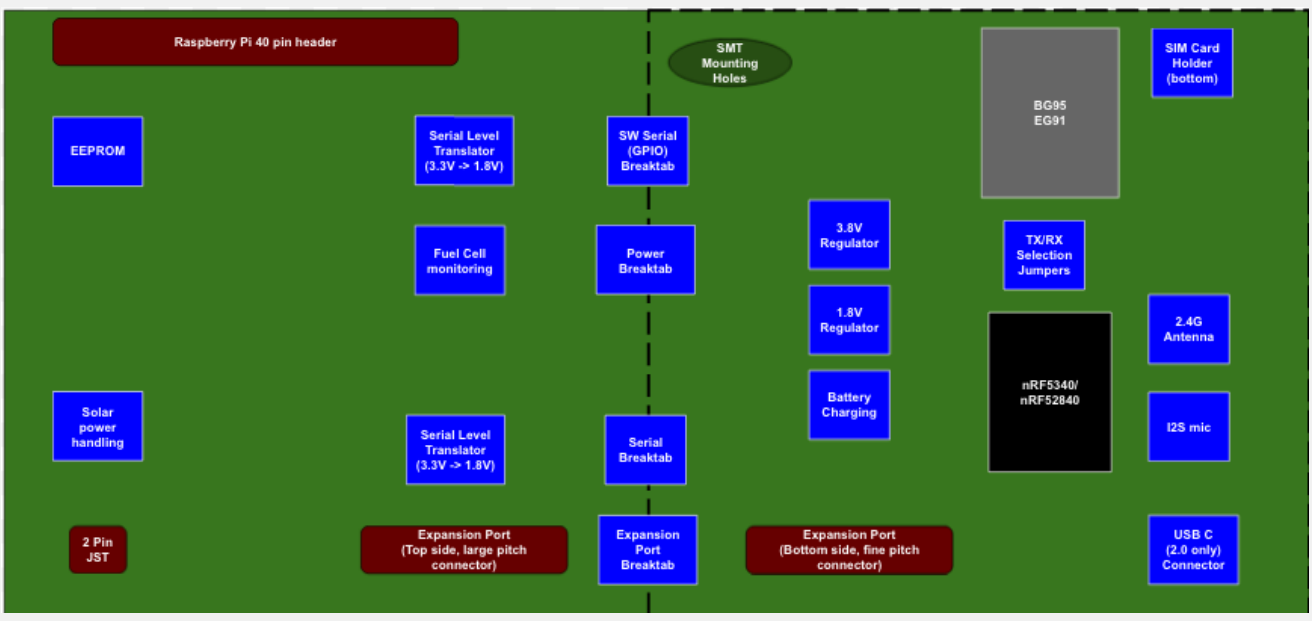
D: CAT 1 vs CAT M1

- CAT 1 has way higher data rates but not enough for video but maybe music
- CAT M1(small data like telemetry and bandwidth is very low so lower cost and lower power aka u cant send videos or high fidelity music)
- In some key markets the infrastructure cant support CAT1/CATM1/CATNB even though u can see the cell tower bc the module may not have the HW/SW to talk to cell tower
- This module is meant to be used as a standalone, We are going to use the nrf52 to control it over serial or UART and make the EG91+BG95 act like modems
    - target market is for micromobility folks

E: AT commands were born back in the 1980s as a way for modems to talk to each other bc before modems were physically attached to each other vs utilizing digital means

# Power Management Decisions

Q: What I learned
A: How will the block diagram change from before EG91 and after EG91




Things to note

- Incorrect power can ruin electronics
- Raspberry Pi takes 5V and will give failures if power drops below 4.6V
- Linux takes more power anyways
- RPI4 takes 5.5V and gives out 3.3V
- Thorough design means you must consider the cost complexity throughout life cycle of the system.
- You must read the HW design guide, bc when they give a minimum value or use case like (LiPo battery for a device) take it seriously
- We may consider using a battery to power this board directly
- ALL THE I/O ON THE EG91 IS 1.8V SO CHRIS suggests having translator & regulators so that voltage is regulated per module
- Its possible to have 4 different power domains:
    - 5V
    - 3.3V
    - 3.8V
    - 1.8V

## Starting the KiCad project and importing components

A HAT looks more or less like this

AS of 7/4/2021 Kicad wouldn't work
Must try again

- First Time Using KiCad
  - Check versions
  - Make sure you have all the files
  - Getting the wrong footprint can ruin project

NS

— DR. MICHAEL RICH

Q: What I learned
GO BRRR EG91 and after EG91

Things to note

- Incorrect power can ruin electronics
- Raspberry Pi takes 5V and will give failures if power drops below 4.6V
- Linux takes more power anyways
- RPI4 takes 5.5V and gives out 3.3V
- Thorough design means you must consider the cost complexity throughout life cycle of the system.
- You must read the HW design guide, bc when they give a minimum value or use case like (LiPo battery for a device) take it seriously
- We may consider using a battery to power this board directly
- ALL THE I/O ON THE EG91 IS 1.8V SO CHRIS suggests having translator & regulators so that voltage is regulated per module
- Its possible to have 4 different power domains:
  - 5V
  - 3.3V
  - 3.8V
  - 1.8V