VECTOR PCB REQUIREMENTS

Situation:

There is a high demand for unlocked Anki Vector robots. There is no official way to unlock a robot anymore. There was, but the company who currently owns the assets does not sell the unlock kits anymore (they are nearing bankruptcy). Our idea was to create a custom PCB for Vector which goes in place of the old one in the same shell and has about the same (if not, better) performance than the original.

Hardware (part 1):

Vector has two main PCBs: a headboard and a bodyboard. The bodyboard acts as a DSP for all the sensors, a motor controller, and a battery charger. It has a low-tier STM32. It sends/receives data frames to/from the headboard via RS232 serial. This is not what we will be recreating. The headboard is what has the actual power. It has:

- Qualcomm APQ8009 (Quad-core Cortex-A7)
- Kingston eMCP (4GB EMMC+4Gb RAM)
- Qualcomm WCN3660B Wi-Fi+BLE combo chip
- BMI160 IMU (SPI)
- 184x96 LCD screen with ST7789 controller (SPI)
- 1280x720 wide-angle camera (MIPI)

We want to adapt the original (mostly-closed-source) software to run on the new board, so some things need to be the EXACT same:

1. IMU: We need the BMI160 IMU, and it should be in the same exact spot as it is on the official board. Must communicate through SPI.
2. Screen: Vector's software is designed to interface with a screen which has this exact resolution and controller. Also must be SPI.
    a. The original board has the screen soldered on. This should honestly be the case with the new one. I don't think there is enough clearance on the front display cover for an FPC.
3. Body connector and communication: We must have the same 4-pin header on the new board. It doesn't have to be in the same exact spot, it could be higher if you need. This is where power comes in and where the serial communication pins are.

We have not been able to find the exact camera. Any camera can be used as long as it fits inside of the display cover. We can write our own camera driver.

Quick Software Rant:

- Hardware will continue later. I think it would be helpful to explain how the software is laid out.
- Vector runs embedded Linux, built with Yocto. There are a set of programs which communicate with each other over IPC, located in a `/anki` folder.
- The `/anki` folder contains:
    - `vic-engine`, `vic-cloud`, `vic-robot`, `vic-anim`, `vic-gateway`, `vic-switchboard`
    - (there are others, but these are the big ones)
- There are also other little Anki programs spread out throughout the filesystem, like `mm-anki-camera`.
    - `mm-anki-camera` is a custom daemon that deals with the strange proprietary Qualcomm camera communication and translates it to something the engine can use. We have the source code for the camera client, so we will be able to create our own mm-anki-camera program for a different camera and CPU.
        - camera ISP I don't think Anki ever utilized Vector's. It seems like they were requesting raw BGR data from the camera and debayering manually. So, the lack of a camera ISP should be fine.
- There was an effort to open-source all of the programs, but that did not happen. Only some source code was released, and it is nowhere near a buildable state. This is why we must keep some of the hardware the same. We can't modify `vic-robot` to use a different IMU. There are some useful things, though, like the camera client header file.

Hardware (part 2):

This is what I was thinking for the new hardware:

- NXP 8M NANO ULTRA LITE 11x11
- USB pads (DN, DP, GND USB_ID). We don’t need a port. For something this enclosed, it would be easier to solder wires and make a custom header that comes out the top.
- We need speaker pads in the top left of the back side of the board, like where they are on the official board.