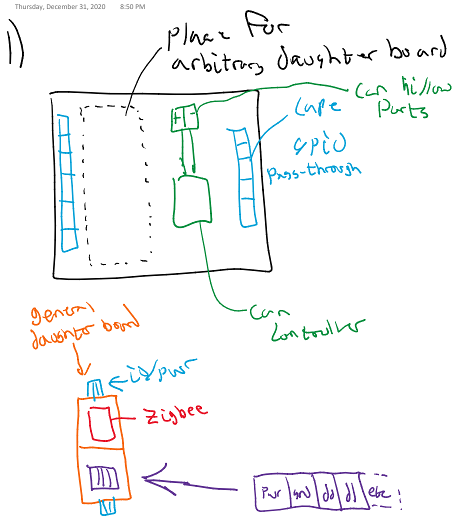
Thursday, December 31, 2020 8:50 PM



- Cape
  - Has dedicated CAN bus tranciever (OBD2)
  - Pass-through beaglebone GPIO breakout
  - Place to plug in daughterboard GPIO/Power
    - Place for daughter board on cape would be specific for a daughterboard we would also create a PCB for
- Daughter board
  - Dedicated Zigbee antenna/microcontroller (common across all daughter boards)
  - Generalized I/O breakout
    - End-user can plug in arbitary sensor (temp, moisture, power relay, etc)
  - Requires us to design a daughter board PCB
  - Possible requires us to design an interconnect PCB (probably just a pin-to-pin connector) to connect a sensor we want to use to the daughter board (if we want to avoid having wires)

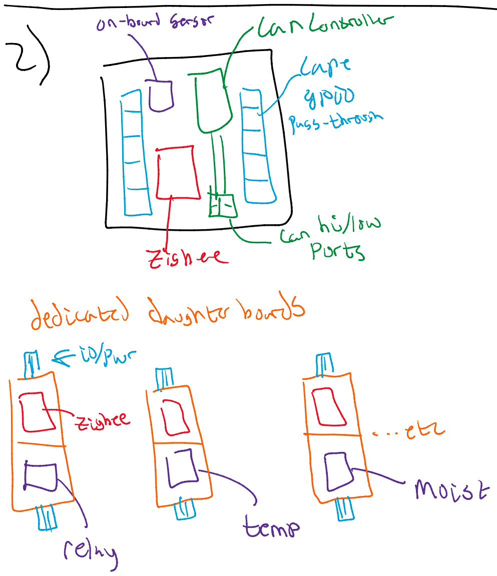
Idea is to develop the cape with a soldered CAN OBD2 port, and a form-fitted space to plug in a daughterboard that has a Zigbee MCU to handle communication with other Zigbee devices. This daughterboard would have a break-out connecter with power, ground, and other data lines to give the ability to plug in any sensor or other control device (relay, temperature, switch, etc).

A thought was that because any daughter boards that needed to be developled would use the same Zigbee MCU that would be used on the cape, we would just plug in a daughterboard to the cape so we wouldn't need to develop the Zigbee connection on the cape seperately. The physical interconnect bwtween the cape and the daughter boards would be made common, so any daughter board can connect to the cape and pass through it's GPIO to the BeagleBoard. There would be a sowftware or hardware switch on the daughter board that would tell if it was connected to the cape to make it a Zigbee reviever or transmitter (assuming the daughterboards only transmit and don't revieve data).

Another thought is that if we're devloping multiple daughterboards each with different sensors, we would develop one daughterboard with a general breakout to make it easier to connect to different sensors, minimizing the amount of PCB designs needed.

Depending on how elegant we want the sensor connections to the daughter boards (plain wires, or pin connections), small pin-mapping boards may need to be developed for each sensor so they can be directly plugged into the daughter boards if their pin-outs don't match our break-out mapping (which they probably won't). These boards would just be a pin-to-pin connection, and any hardware-filtering or logic would be on the daughter board.

It should be noted that in this design, the cape itself doesn't directly support the Zigbee protocol without a daughter board attached, making it not satisfy a design requirement out right. However, it's assumed that in this design, the cape wouldn't be used without a daughter board attached anyway, so that requirement would be satisified.



- Cape
  - Dedicated CAN bus tranciever (OBD2)
  - Pass-through beaglebone GPIO breakout
  - Dedicated Zigbee antenna/microcontroller
  - Dedicated on-board sensor (maybe)
- Daughter board
  - Dedicated Zigbee antenna/microcontroller (common across all daughter boards)
  - Single-purpose I/O with sensor soldered
    - Specific to a sensor (temp, moisture, power relay, etc)
  - Requires us to design multiple daughter board PCBs (one for each sensor we want to have for demo)

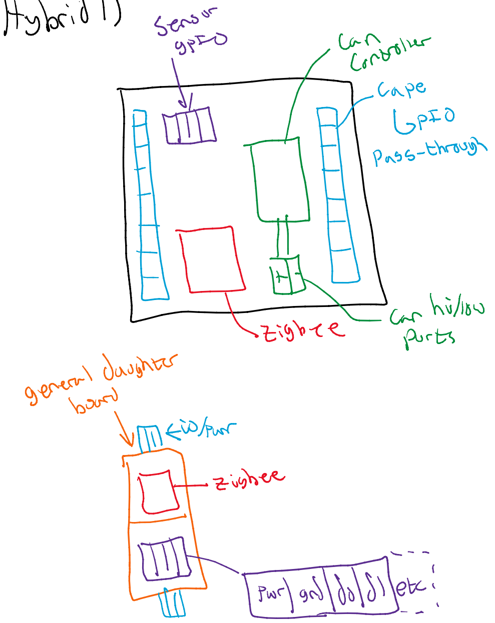
Idea is to develop the cape with a soldered CAN OBD2 port, a dedicated Zigbee MCU, and an additional sensor on-board. Additional daughterboards would be developed with the same or similar Zigbee MCU, and an attached sensor or control device (relay, temp, switch, etc) would specific to each daughter board.

A thought was that it would be easier to develop separate Zigbee PCB designs for the daughter boards and cape in parallel rather than developing one general connection and an interconnect layer (that mostly exists in firmware) for the zigbee MCUs. Also, by "hard-coding" what each daughter board's "role" is (what sensor or control device it has), pin-to-pin connectors wouldn't need to be devoped to elegantly connect said sensors to the daughter boards.

Also, in this design the cape directly supports the Zigbee protocol which we know does satisify a client design requirement. However, it's assumed that we would still be designing daughter boards for a final demo

Note:
In either design 1 or 2, additional daugher boards would need to be deveoped (the amout of different PCBs for boards would depend on which design is chosen). In either case, the cape would still have an OBD2 connection, and would be given a zigbee MCU (how that MCU is attached depends on what design is chosen). Firmware for the daughterboars would largely be the same between each design. In the case of design 1, an api would be created to allow an end-user to program what sensor is attached (like an arduino, for example), and in the case of design 2, that api would still be created to generalize the firmware for each board, but what sensor is attached would be hard coded and would not be changeable to the end-user.

Hybrid 1)

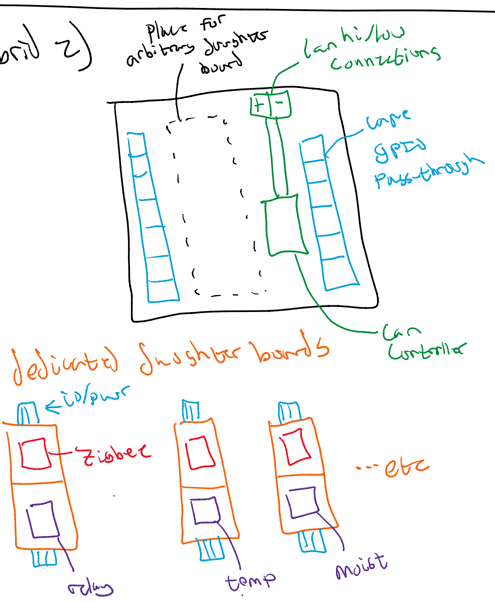


- Cape
    - Dedicated CAN bus tranciever (OBD2)
    - Pass-through beaglebone GPIO breakout
    - Dedicated Zigbee antenna/microcontroller
    - Generalized I/O breakout
        - User can plug in arbitary sensor (temp, moisture, power relay, etc)
- Daughter board
    - Dedicated Zigbee antenna/microcontroller (common across all daughter boards)
    - Generalized I/O breakout
        - End-user can plug in arbitary sensor (temp, moisture, power relay, etc)
    - No reason this couldn't be any arbitary sensor with zigbee connectivity, but we would create our own daughterboard PCB design

Takes the cape design of number 2 (with the change that the on-board sensor would be general I/O instead), and the daughter board design of number 1. Daughter boards are not specific to a sensor, and have break-out pins instead.

Con from number 1 with making pin-to-pin PCBs to elegantly connect sensors to daughter boards.
Con from number 2 of developing different Zigbee MCU connections in parallel for the cape and daughter boards.
Pro from number 2 that the cape itself direcly supports the Zigbee protocol without a daughter board attached

Hybrid 2)

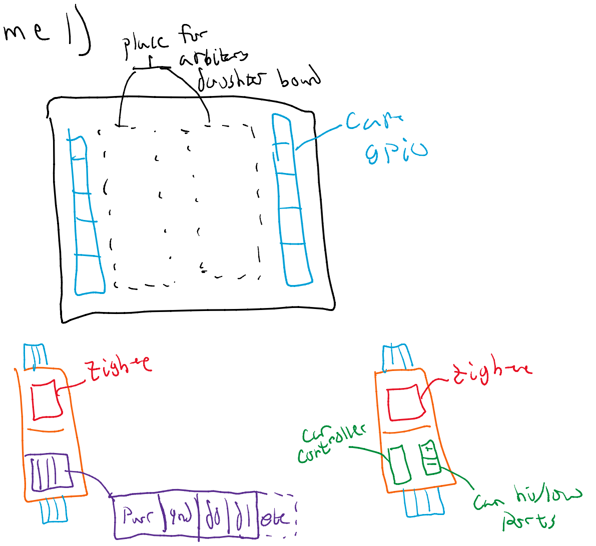


- Cape
    - Dedicated CAN bus tranciever (OBD2)
    - Pass-through beaglebone GPIO breakout
    - Place to plug in daughterboard GPIO/Power
        - Place for daughter board on cape would be specific for a daughterboard we would create a PCB for
- Daughter board
    - Dedicated Zigbee antenna/microcontroller (common across all daughter boards)
    - Single-purpose I/O with sensor soldered
        - Specific to a sensor (temp, moisture, power relay, etc)
    - Requires us to design a daughter board PCB

Takes the cape design of number 1, and the daughter board design of number 2. Daughter boards are not specific to a sensor, and have break-out pins instead.

Con number 2 of developing separate daughter board PCB designs for each sensor we want.
Pro from number 1 of not having to create 2 Zigbee interconnects in parallel (one for the cape, one for the daughter boards).
Con from number 1 of the cape not directly supporting the Zigbee protocol without a daughter board attached

Extreme 1)



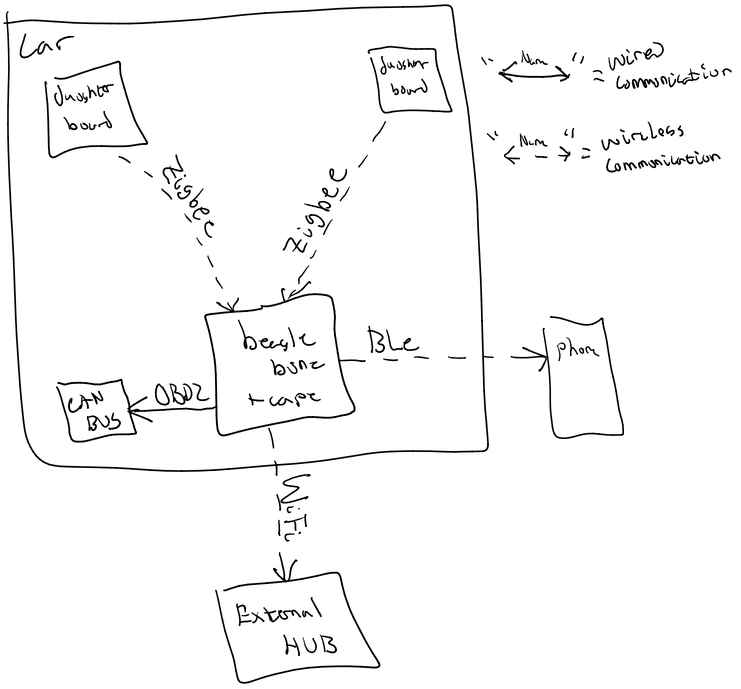
- Cape
    - Pass-through beaglebone GPIO breakout
    - 2 places to plug in daughterboard GPIO/Power
        - Place for daughter board on cape would be specific for a daughterboard we would create a PCB for
- Daughter board
    - Dedicated Zigbee antenna/microcontroller (common across all daughter boards)
    - 1 with generalized I/O breakout
        - End-user can plug in arbitary sensor (temp, moisture, power relay, etc)
    - 1 with CAN OBD2 controller and required bus ports
    - Requires us to design 2 daughter board PCBs (one for general I/O, one specific for CAN)

Takes the design of number 1 to the extreme. Cape just becomes an interconnect layer to connect daughter boards to the BeagleBoard. It would also have software to tell that if two daughter boards are connected, 1 would act as the Zigbee hub, and the other's Zigbee MCU would be disabled.

Cape becomes much easier to develop, because it's essnetially just pin-to-pin mappings between the daughter boards and the BeagleBoard. 2 daughter board designs would be needed, one specific to the CAN controller we're using, and another for the generalized sensor connection.

Con from number 1 with making pin-to-pin PCBs to elegantly connect sensors to daughter boards.
Now making 2 daughter board designs, and now the cape itself doesn't directly support an aditional wireless protocol without having a daughter board attached, which may not be what the client desires.

Communication Layout


Car
daughter board
daughter board
Zigbee
Zigbee
beagle bone + cape
BLE
Phone
OBD2
CAN BUS
WiFi
External HUB
" ←Num→ " = wired communication
" ← - - → " = wireless communication