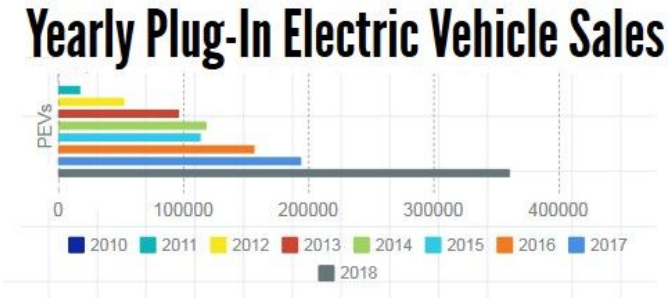
Cheap uBMS

a cheap
micro-battery management system (BMS)
for anomaly detection

Motivation: 11,000 MWh

Plug-In Sales by Year - 2010* thru January 2019	
Year	**Total**
2010	345
2011	17,735
2012	52,835
2013	96,702
2014	118,773
2015	114,022
2016	157,112
2017	194,479
2018	361,307
2019 YTD	17,040
Total	1,130,350
	*mass-market introduction

Photo Credits:
“ Electric Drive Sales Dashboard”
by Electric Drive Transportation Association
https://electricdrive.org/index.php?ht=d/sp/i/20952/pid/20952



Quick Power Math:

1,130,350 AEVs + PHEVs

10 kWh batteries (on avg)

11,303,500 kWh “rolling” capacity

~11,000 MWh

How much is 11,000 MWh?

Boeing 777-300ER (Extended Range)

2x 70MW Engines for **6 days non stop**

Photo Credit:
"The Boeing 777-300ER: On Your Mark. Get Lighter. Go!"
by Boeing
https://www.youtube.com/watch?v=sN90SXMM1q4

Enrico Fermi Nuclear Station

1098 MW Reactor for **10 hours**

Photo Credit:
"The Fermi Station (NRC image)"
by Nuclear Regulatory Commission
https://en.wikipedia.org/wiki/Enrico_Fermi_Nuclear_Generating_Station#/media/File:Fermi_NPP.jpg

Lithium Ion can be dangerous...

Hoverboards with Lithium Ion Batteries

Photo Credit:
AltRiders
https://altriders.com/are-hoverboards-safe/

Samsung Galaxy Note 7

Photo Credit:
wbrz.com
https://thebottomline.as.ucsb.edu/2016/09/catching-fire-samsung-galaxy-note-7-batteries

… and cars can be compromised.

Jeep hacked by Charlie Miller and Chris Valasek

Photo Credit:
“Hackers Remotely Kill a Jeep on the Highway”
by Wired
https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

Jeep Cherokee

Remotely hacked to:

1. Change HVAC settings
2. Controlling digital display
3. Transmission (powertrain!)
4. Kill engine (powertrain!)
5. Disable braking (powertrain!)

Putting these all together?

Photo Credit:
“Chevrolet Volt”
by GM
https://www.chevrolet.com/electric/volt-plug-in-hybrid

Photo Credit:
Core77
https://www.core77.com/posts/74535/Understanding-Why-Electric-Car-Fires-Pose-a-Unique-Hazard

11,000 MW of rolling electricity,

that can catch fire,

and be controlled by someone else.

So what do we do?

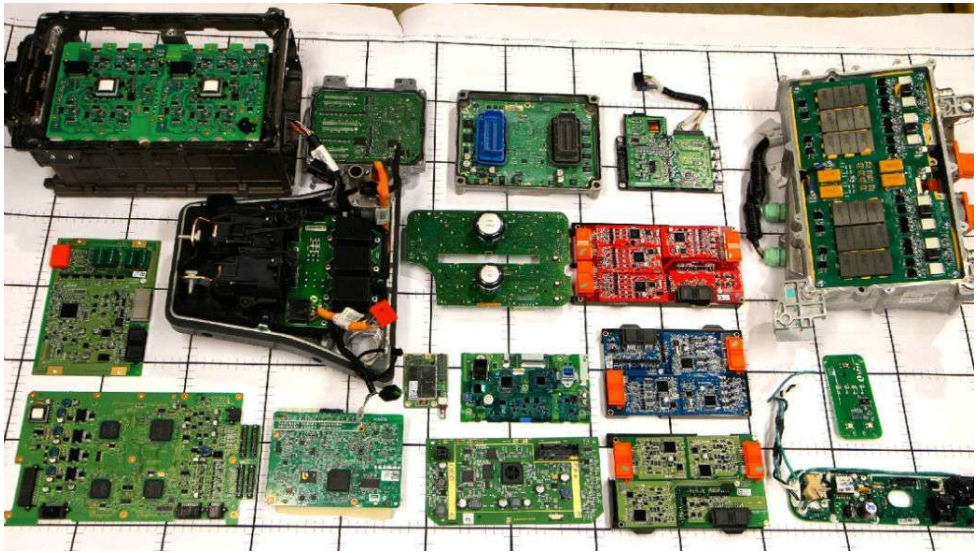


Photo Credit:
UBM Technologies
https://www.edn.com/design/systems-design/4391497/Teardown--High-voltage-Li-ion-battery-stack-management---the-drive-for-safe-power

Learn about the state of the art!

The state of the art:

Research

+ Covers cell reconfig, SoH estimation, battery balancing, real-time scheduling
+ Offers predictive models and offline anomaly detection
- **Few real-world implementations**
- **No online anomaly detection**

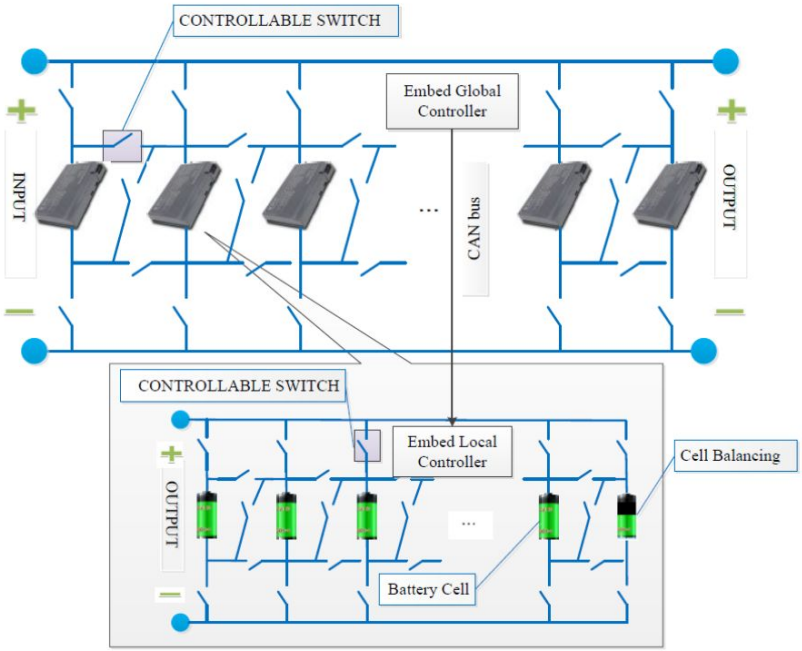


FIGURE 1. A diagram of typical large-scale reconfigurable battery pack design.

Image Credit:
Reconfigurable Battery Pack
By Ci et al.
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7442763

The state of the art:

BMS1040BT

Back to: BMS10x

40V, 100 Amps Management System for 6 to 10 Cells Lithium Ion Batteries. Bluetooth

BMS10x0 Datasheet
BMS10x0 CANOpen Manual

Price $425.00

Tax amount

Stock: Contact us for leadtime

1 Add to Cart

Image Credit:
BMS1040BT
By RoboteQ
https://www.roboteq.com/index.php/component/virtuemart/419/43/404/battery-management-io-extenders/bms10x0-battery-management-systems/bms1040-400-detail?Itemid=0

Industry

\+ **Many real-world implementations**
\- **Expensive**
\- **Not transparent**
\- **No built-in, online anomaly detection**

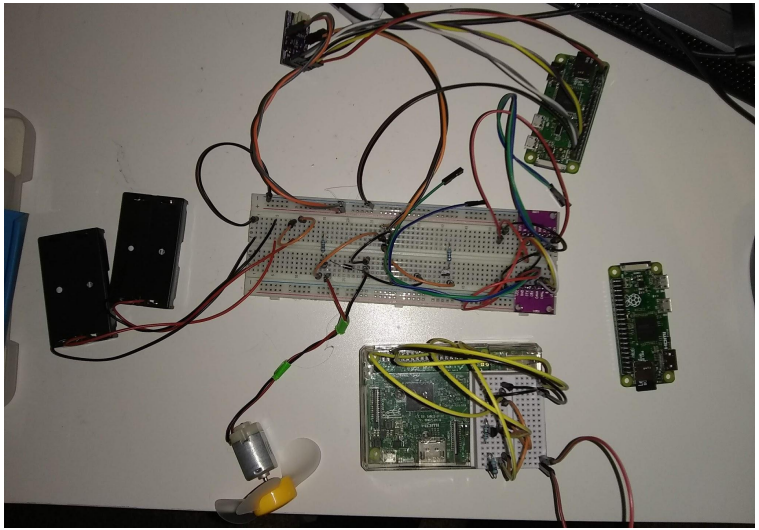
Problem Statement

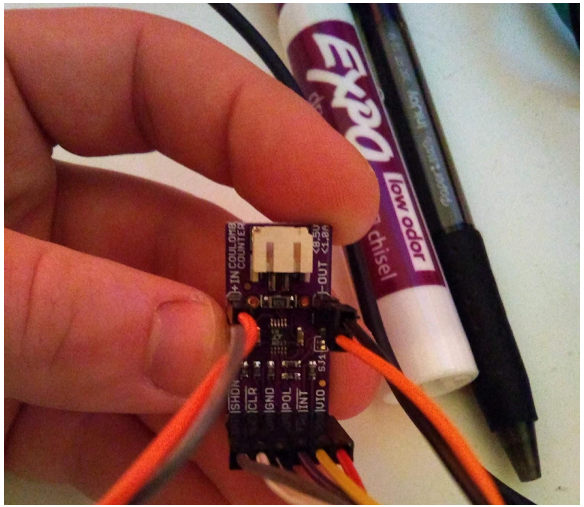
To the best of our knowledge,

existing battery management systems and research thereon:

1. are **expensive** to implement,
2. do not have online **anomaly detection**, and
3. do not use **request-based communication.**

Approach: Feasibility Study + Formal Model

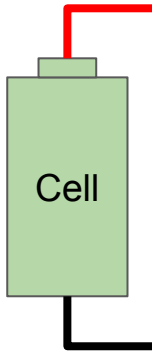




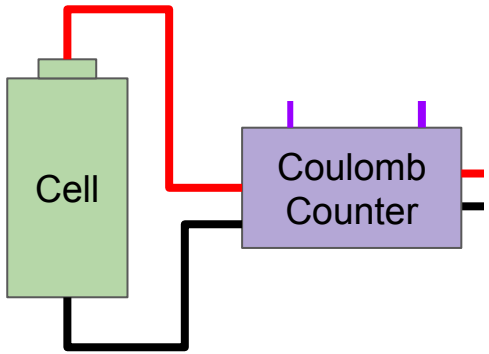
Create a small-scale, low-cost BMS
to demonstrate online anomaly detection
via request-based communication.

uBMS Design

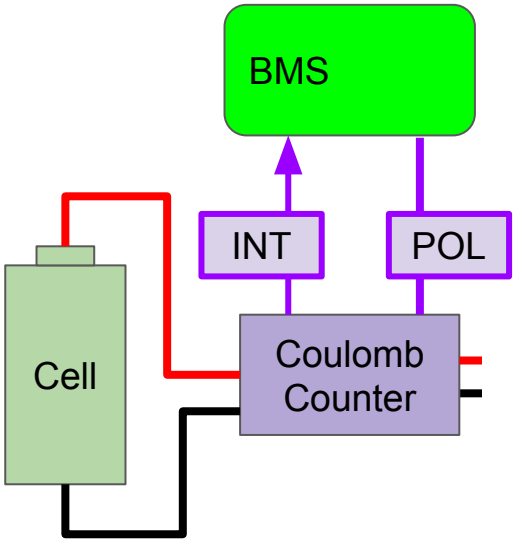
uBMS Model



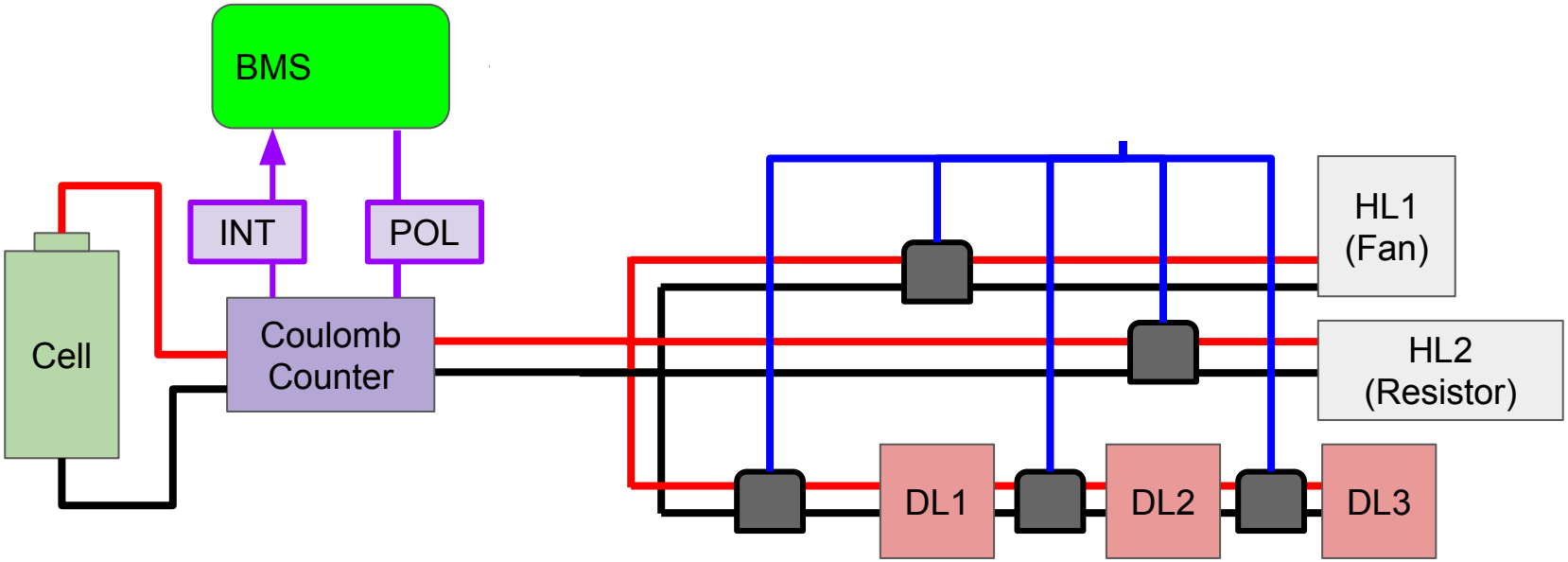
uBMS Model



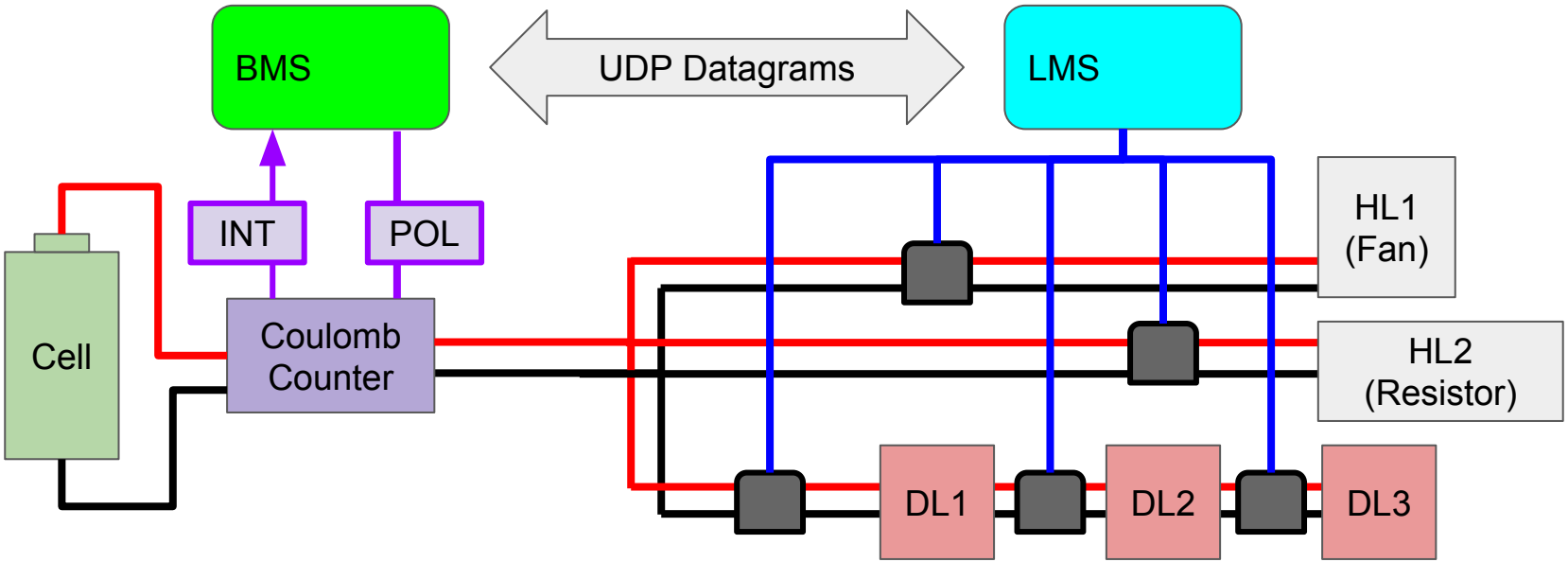
uBMS Model



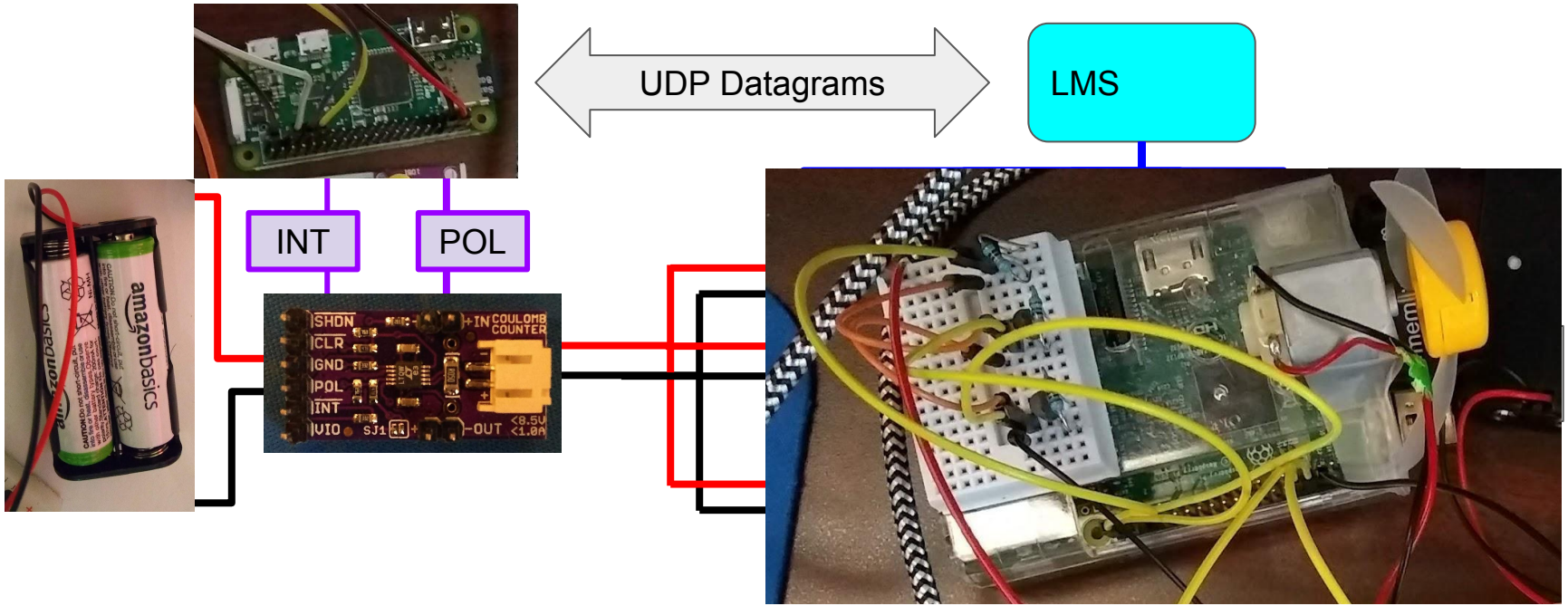
uBMS Model


BMS
INT
POL
Cell
Coulomb
Counter
HL1
(Fan)
HL2
(Resistor)
DL1
DL2
DL3

uBMS Model


BMS
UDP Datagrams
LMS
INT
POL
Cell
Coulomb
Counter
HL1
(Fan)
HL2
(Resistor)
DL1
DL2
DL3

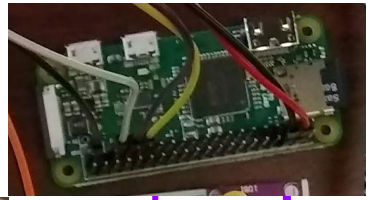
uBMS Model


UDP Datagrams
LMS
INT
POL
SHDN
CLR
GND
POL
INT
VIO
SJ1
+IN
COULOMB COUNTER
-OUT
<8.5V
<1.0A
amazonbasics

uBMS Model

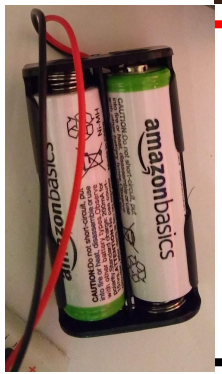
Total: $95

RPi Zero W - $5



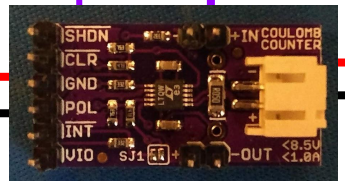
UDP Datagrams

LMS



INT

POL



LTC4150 Module
$12

Battery
Case - $8

Basic Electronics Kit - $30

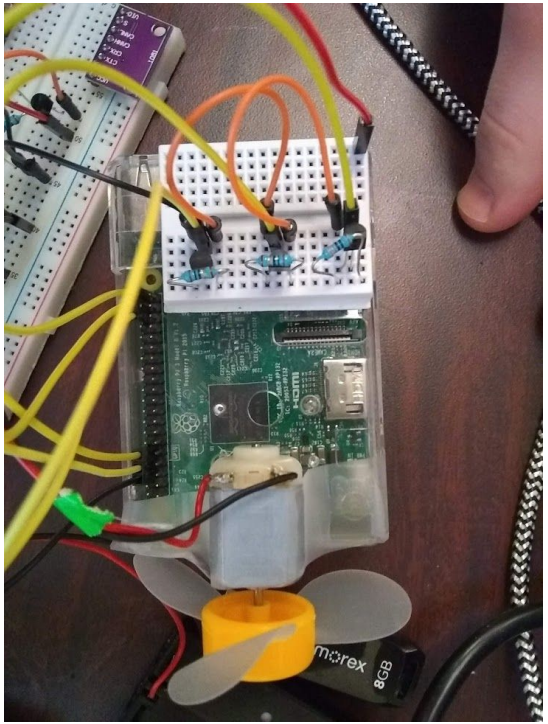
RPi 3B - $40

Request-Based Communication

Load Management System (LMS)

Responsible for:

1. Identifying and modeling attached loads.
2. Creating load requests identifying when and how loads are to be powered.
3. Activating and deactivating loads based on accepted/rejected load requests.



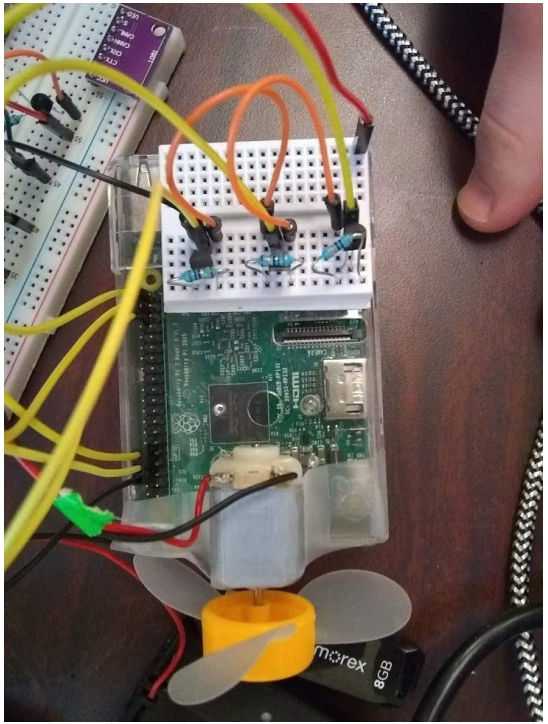
Example Request: Turning on a Fan

1. Create a model for the fan:
   (Voltage range, current range, power, energy, start and end time, unique ID)

2. Send a request to the BMS

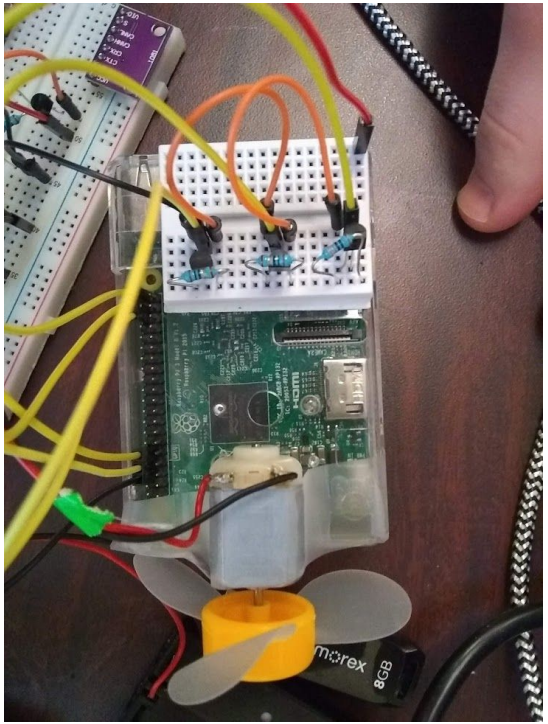
3. Wait for acceptance/rejection

4. If accepted, turn on fan at the start time.
   If rejected, adjust model based on error.



Example Request: Turning on a Fan

```
(0V,        #Min voltage
6V,         #Max voltage
0A,         #Min current
0.200A,     #Max current
0W,         #Min Power
1.2W,             #Max Power
10s,        #Release time (seconds)
10s,        #Duration (seconds)
0J,         #Min Energy (Joules)
12J,        #Max Energy (Joules)
120s,       #Deadline
0x0217)     #Unique Token
```



Anomaly Detection

What would be anomalous?

Battery Management Systems can monitor:

1. State of Charge (SoC)
2. Depth of Discharge (DoD)
3. State of Health (SoH)
4. Configuration (Series, Parallel, Mix, etc)
5. Min/Max Voltage
6. Min/Max Current
7. Temperature
8. Number of cycles
9. Total power delivered
10. Total power received
11. Open-Circuit Voltage (OCV)
12. Relaxation Time
13. Communications (CAN, Eth, etc.)

Any of these parameters can deviate from expectations.

How do we know what the expectations **are**?

What would be anomalous?

Battery Management Systems can monitor:

1. State of Charge (SoC)
2. Depth of Discharge (DoD)
3. State of Health (SoH)
4. Configuration (Series, Parallel, Mix, etc)
5. Min/Max Voltage
6. Min/Max Current
7. Temperature
8. Number of cycles
9. Total power delivered
10. Total power received
11. Open-Circuit Voltage (OCV)
12. Relaxation Time
13. Communications (CAN, Eth, etc.)

Any of these parameters can deviate from expectations.

How do we know what the expectations **are**?

Use the load requests!

+

Use a model for the battery!

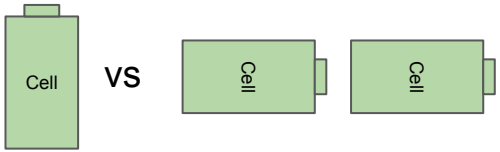
What do we detect?

Battery Management Systems can monitor:

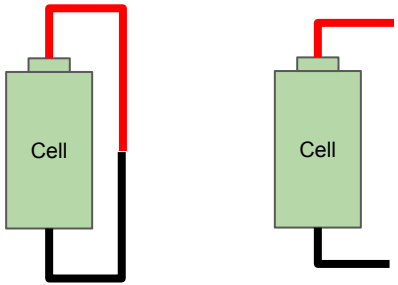
1. State of Charge (SoC)
2. Depth of Discharge (DoD)
3. State of Health (SoH)
4. **Configuration (Series, Parallel, Mix, etc)**
5. **Min/Max Voltage**
6. **Min/Max Current**
7. Temperature
8. Number of cycles
9. **Total power delivered**
10. Total power received
11. **Open-Circuit Voltage (OCV)**
12. Relaxation Time
13. Communications (CAN, Eth, etc.)

Detectable Anomalies

- **Over/under-voltage**:
  Supply is too high/low



- **Over/under-current**:
  Resistance too small/large



- **Overdraw**: Too much energy expended
- **Overtime***: Exceeded maximum duration

Results + Conclusion

Results

Nothing has caught fire,

No one has died,

…

to the best of our knowledge

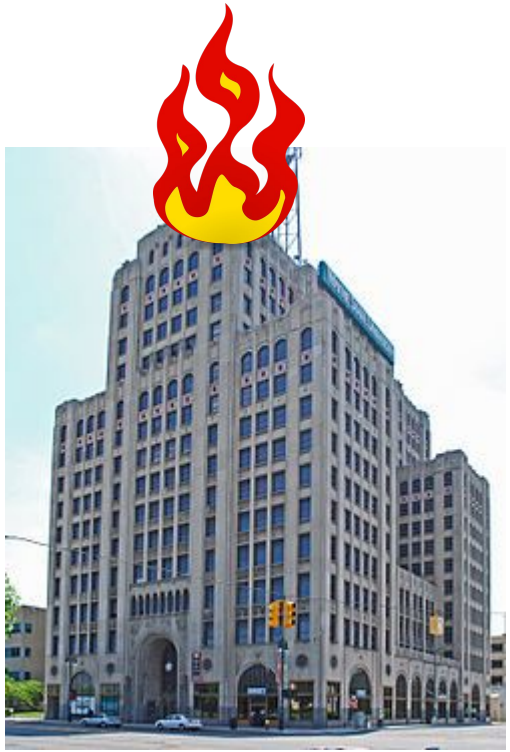


Image Credit:
Macabees Building Detroit
By Andrew Jameson
https://en.wikipedia.org/wiki/Maccabees_Building#/media/File:MacabeesBuilding2010.jpg

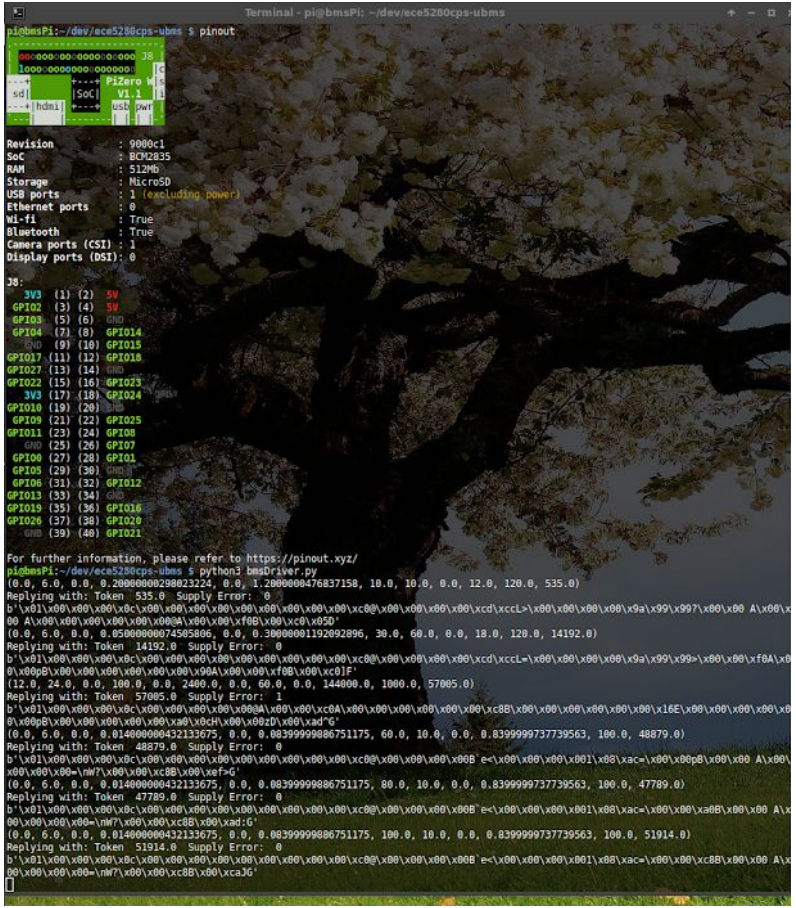
Results

Demonstrated:

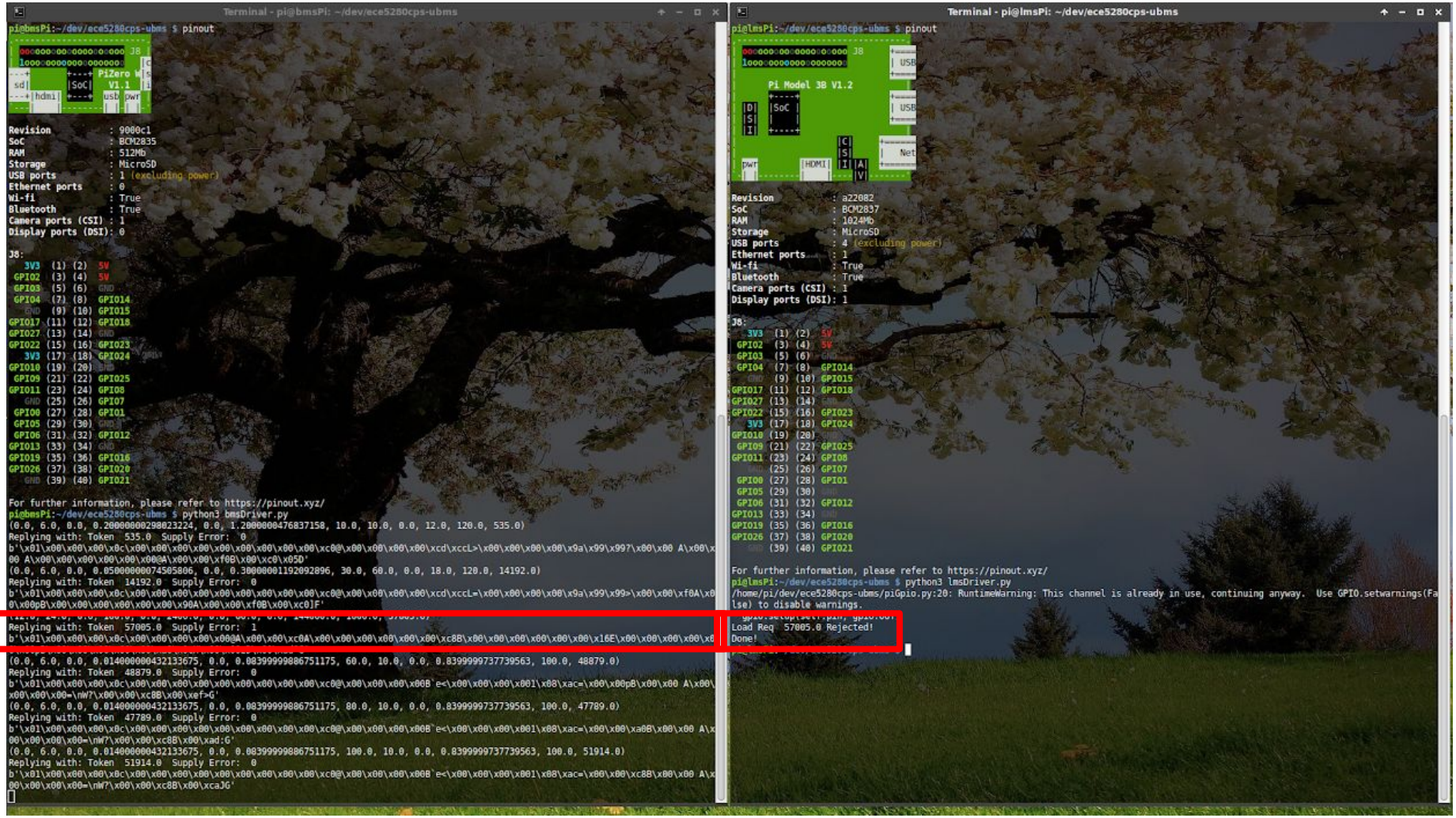
1. Rejecting over- and under-voltage loads
2. Rejecting over- and under-current loads

Left to demonstrate:

1. Rejecting loads exceeding energy schedule
2. Rejecting loads exceeding time schedule



Output trace of several load requests of which one over-voltage load is rejected.


Terminal - pi@bmsPi: ~/dev/ece5280cps-ubms
pi@bmsPi:~/dev/ece5280cps-ubms $ pinout
Revision : 9000c1
SoC : BCM2835
RAM : 512Mb
Storage : MicroSD
USB ports : 1 (excluding power)
Ethernet ports : 0
Wi-fi : True
Bluetooth : True
Camera ports (CSI) : 1
Display ports (DSI): 0
J8:
3V3 (1) (2) 5V
GPIO2 (3) (4) 5V
GPIO3 (5) (6) GND
GPIO4 (7) (8) GPIO14
GND (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
GND (25) (26) GPIO7
GPIO0 (27) (28) GPIO1
GPIO5 (29) (30) GND
GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
GND (39) (40) GPIO21
For further information, please refer to https://pinout.xyz/
pi@bmsPi:~/dev/ece5280cps-ubms $ python3 bmsDriver.py
Replying with: Token 535.0 Supply Error: 0
Replying with: Token 14192.0 Supply Error: 0
Replying with: Token 57005.0 Supply Error: 1
Replying with: Token 48879.0 Supply Error: 0
Replying with: Token 47789.0 Supply Error: 0
Replying with: Token 51914.0 Supply Error: 0
Terminal - pi@lmsPi: ~/dev/ece5280cps-ubms
pi@lmsPi:~/dev/ece5280cps-ubms $ pinout
Pi Model 3B V1.2
Revision : a22082
SoC : BCM2837
RAM : 1024Mb
Storage : MicroSD
USB ports : 4 (excluding power)
Ethernet ports : 1
Wi-fi : True
Bluetooth : True
Camera ports (CSI) : 1
Display ports (DSI): 1
J8:
3V3 (1) (2) 5V
GPIO2 (3) (4) 5V
GPIO3 (5) (6) GND
GPIO4 (7) (8) GPIO14
GND (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
(25) (26) GPIO7
GPIO0 (27) (28) GPIO1
GPIO5 (29) (30) GND
GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
GND (39) (40) GPIO21
For further information, please refer to https://pinout.xyz/
pi@lmsPi:~/dev/ece5280cps-ubms $ python3 lmsDriver.py
/home/pi/dev/ece5280cps-ubms/piGpio.py:20: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
Load Req 57005.0 Rejected!
Done!

https://photos.app.goo.gl/M6xMHJvLDEd4Kmc39

Conclusion: Lessons + Outcomes

1. BMS hardware is not the largest expense
2. uBMS can be reproduced for <$100
3. Online anomaly detection is viable through request-based communication.
4. Code is available online at GitHub

https://github.com/aarontwillcock/ece5280cps-ubms

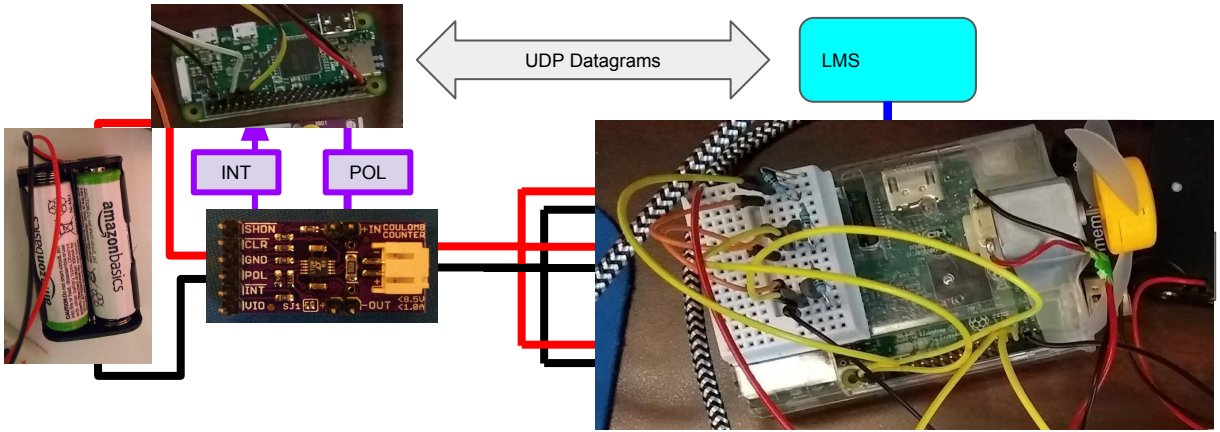
Conclusion: Weaknesses

1. Smaller, parasitic loads can hide in the “wake” of larger load requests
2. Load requests with wide current ranges provide more “hiding space”
3. Assumes BMS has only coarse-grained control of loads (ex. Cutting all power)
4. Does not implement priority
5. Deferred to WiFi from CAN (purchased wrong modules)

Questions?

Public Code

https://github.com/aarontwillcock/ece5280cps-ubms




```
(0V,        #Min voltage
6V,         #Max voltage
0A,         #Min current
0.200A,     #Max current
0W,         #Min Power
1.2W,           #Max Power
10s,        #Release time (seconds)
10s,        #Duration (seconds)
0J,         #Min Energy (Joules)
12J,        #Max Energy (Joules)
120s,       #Deadline
0x0217)     #Unique Token
```