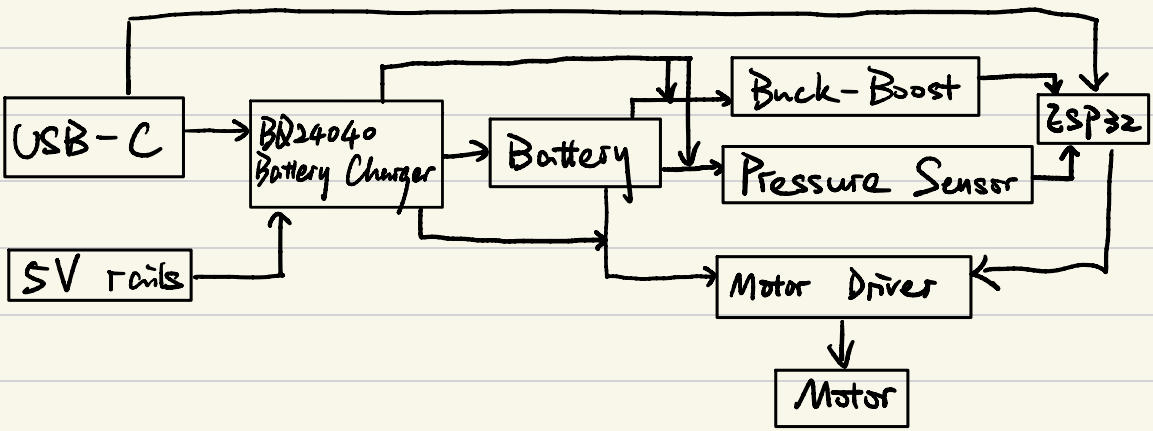
2024/01/27

# Updated Block Diagram & Bring-up Plan




Testing: while USB-C un-soldered:

1. Checking for shorts between:

   5V, Battery Voltage (↓ 3V ~ 4.2V), 3.3V, GND

2. Isolate buck-boost converter:

   disconnect battery, no power into battery charger

   use power supply to provide:

   1) input 3V ~ 4.2V, max current 1.25A

   measure:

   output 3.3V (2A)

   2) UVLO testing

   a) input falling gradually to 3V. TP4(EN) ≈ 1V. output 0V when input below 3V ↑

b) Input below 3V, gradually rises to 3V,
TP4 (EN) ≈ 1V as Input = 3V, output 0V when
Input below 3V & 3.3V when input rises to 3V.

3. Testing Battery Charger

(I realized that I didn't separate charger from the rest of the ckt & am a little worried that it might burn everything)

Assuming battery not fully charged:

Power supply 5V 1A to charger Input

1) Yellow LED (plugged in) should be ON ($\overline{PG}$ low).
Orange LED (charging) ON ($\overline{CHG}$ low)
Output voltage between 3V ~ 4.2V

2) Charge current limit:
Might actually need to GND ISET2 to enable programmable current limit.
(ISET2 floating right now)

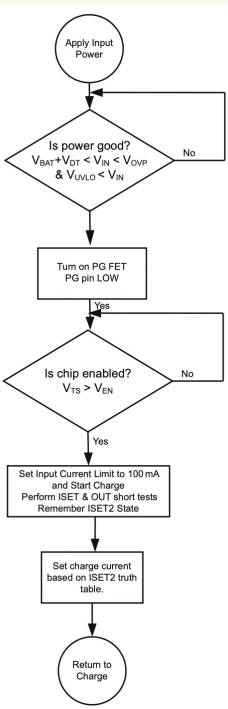


Figure 8-3. BQ2404x Power-Up Flow Diagram

ISET2 is a 3-state input and programs the Input Current Limit/Regulation Threshold. A low will program a regulated fast charge current via the ISET resistor and is the maximum allowed input/output current for any ISET2 setting, Float will program a 100mA Current limit and High will program a 500mA Current limit.

I also can't tell from data sheet (which just says program ISET & ITERM with pins 2 & 4) whether you can also MEASURE with them. If no I don't

know how to test them other than carefully tracking battery charge level & measure I-OUT. If yes TP16 & 15 might be of use.

3) I think most importantly just make sure:
   Battery voltage pins still between 3V & 4.2V (includes buck-boost IN, sensor VDD, driver VDD).
   Buck-boost OUT still 3.3V.

Then we can safely connect USB-C (which powers battery charger) & test digital sections.

Testing with USB-C soldered:

4. ESP32
   1) EN (TP6) 0V when reset pressed & non-zero otherwise
   2) Test ESP32 works with boot button & digital read: IO 0, pin 27. Low when button pressed

5. Motor Driver
   1) Check motor works by directly powering it with 2V~5V
   2) Check 3V~4.2V power supply to VDD
   3) Try some simple digital write with ESP32
      SCL to pin 25, IO48 } on ESP32
      SDA to pin 24, IO47

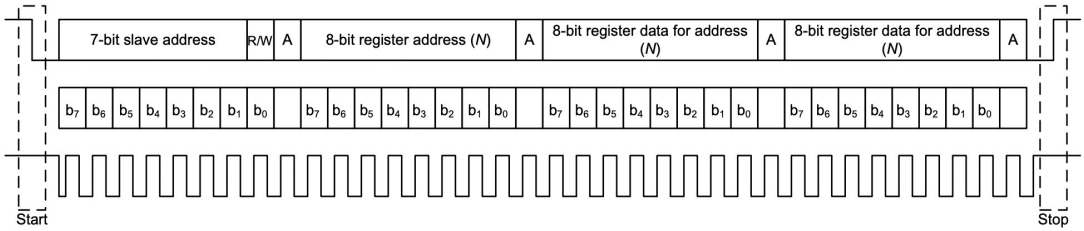



**Figure 20. Typical I²C Sequence**

The DRV2605L device operates as an I²C-slave 1.8-V logic thresholds, but can operate up to the $V_{DD}$ voltage. The device address is 0x5A (7-bit), or 1011010 in binary which is equivalent to 0xB4 (8-bit) for writing and 0xB5 (8-bit) for reading.

Adafruit library linked in schematic for examples.

6. Check pressure sensor
   1) Check 3V ~ 4.2V power suppy to VDD
   2) Connect mechanical parts & try some reading with ESP32. Example code on Adafruit & Xyla's website.

   Pin 29, IO 36 SDA

   Pin 30, IO 37 SCL

   Detailed I²C communication info on datasheet.

7. Check IoT connection.

   Once pressure sensor readings are good, check connecting ESP-32 to demo-LED board's ESP-32 via IoT & try controlling LEDs with sensor readings.