Battery Charger / Discharger Ideas

(Electrical Side)

Main proposal:

The aim of the device is to automatically charge and discharge the batteries safely, regulating current, potential difference.

Safety features will be included such as overvoltage, overcurrent and overheat protection to reduce risk of fire and damage to the batteries.

Key ideas:

Microcontroller controls the circuit, turning on and off the charger and discharger loop. This allows for automatic charging and discharging.

The microcontroller would have equipment connected to monitor the batteries and when the voltage passes a threshold (e.g. 14.4V) or the current falls below a threshold it would turn off the charger.

When discharging, the battery will be connected to a resistor bank or heaters. When the PD drops past 11.9V, the discharger would be disconnected and the charger switched back on. Leaving the batteries in a discharged state for too long can damage them so charging would be done immediately.

Rest periods would have to be allowed for the batteries because constant charge/ discharge cycling will damage batteries.

Safety features:

- Overheat protection: Turns the power for all chargers and dischargers off in case of fire. System will have to be manually reset before restarting to check for damage.
- Overcurrent protection: If current becomes too high in one of the wires, the power will be turned off for a single battery so that it can be manually checked before resetting.
- Emergency stop button: A safety system that you can use to turn everything off manually instead of relying on automatic systems.
- Overvoltage/undervoltage protection: Will turn off power to the single battery if the PD gets too low or too high.

- Power cutoff protection: MOFSET gate with strong pull-down resistor so that batteries don't continue discharging (as this could be dangerous). Small 5-12V UPS module for microcontroller so that it can log final data and cleanly shut down.
- Manual reset button: this must be pressed to reset the system after a fault has occurred (such as overheat) so that someone can manually check everything is in order.
- Microcontroller crash protection: 'watchdog' protection. Checks that everything is working, if not it will shut down the microcontroller and resets everything so that no damage happens.

Other ideas:

SD card log: logs events such as times each battery switches state, max voltage, discharge time, reason for faults. This lets you see the data for each individual battery so you can find any faulty ones and see why the system tripped.

Email system: lets teachers know when a fault has occurred so that they are aware and can check and reset it when they are free. Means that it doesn't need to be constantly checked on.

Displays for battery monitoring: displays their current state - charging, resting, discharging or fault.

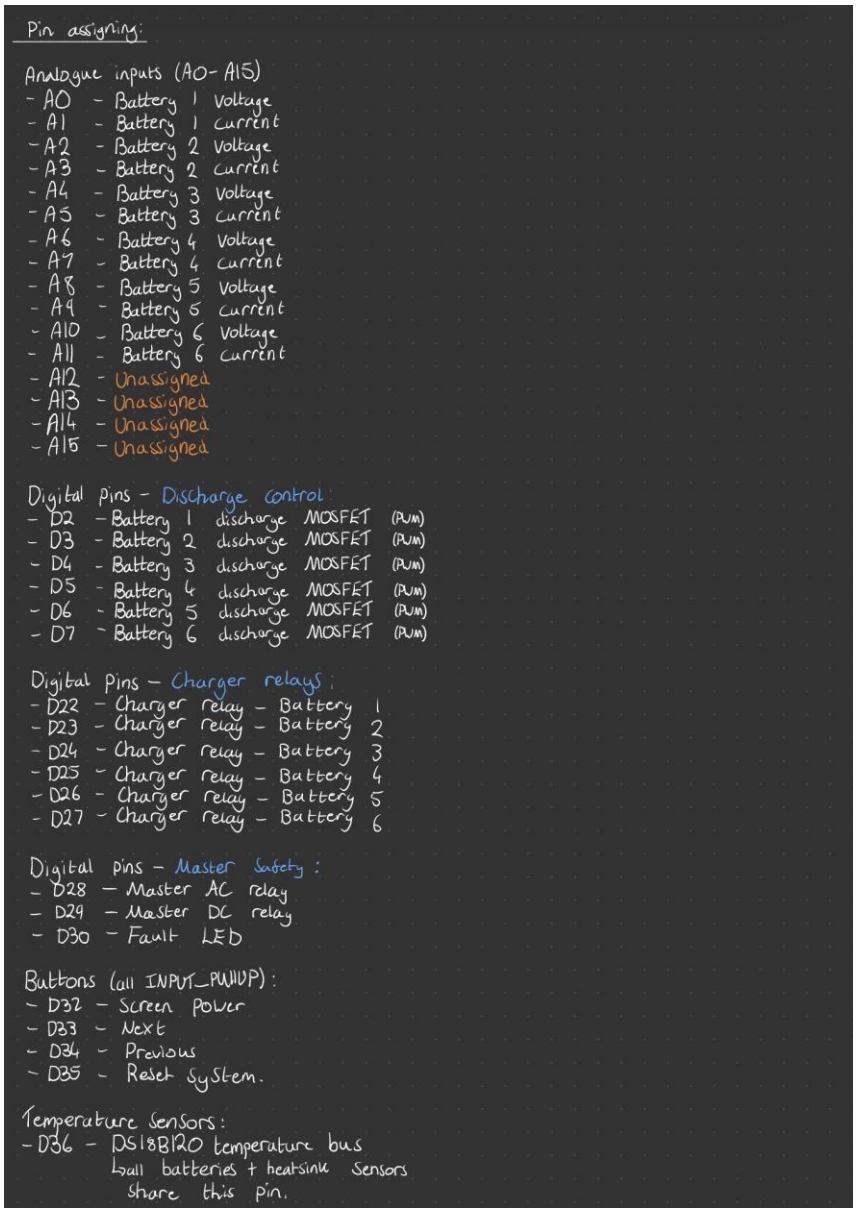
Fans: this reduces the chance of overheating inside the container

Hardware

- Arduino mega 2560
- IEC C14 inlet
- MCB Fuse
- AC Relay
- DC Relays (x6)
- DIN Sockets (x6)
- Transistor (x6) (for relays)
- 60A fuse (x6)
- Current sensor (ACS785) (x6)
- Temperature sensors (DS18B20)
- LCD screen (20x4)

- SD card Module
- Wi-Fi module (ESP8266)
- Momentary Switches (buttons) (x3)
- Many different Resistors
- Resistor load (100W, 3 ohm) (x3 per battery)
- Emergency stop button
- Fans (x?)
- Reset Button (momentary switch)
- Wire (need to work out gauge and different sizes)

Test Pin Assignment & Some Wiring Diagrams

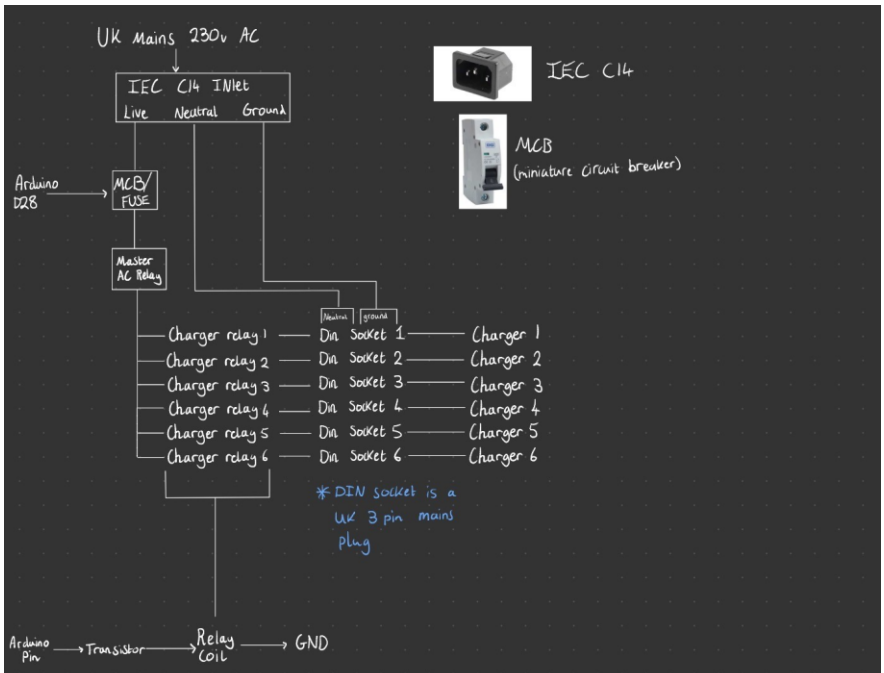


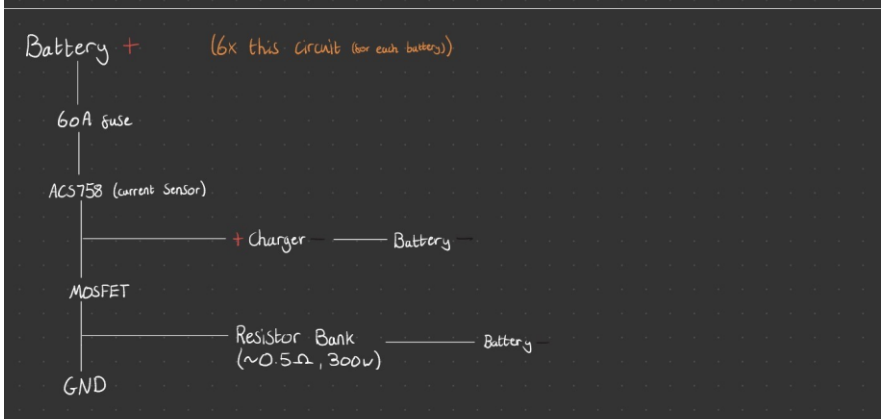
SPI bus (Shared) (for SD card module):
- D50 - MISO
- D51 - MOSI
- D52 - SCK
- D53 - SD Card CS.

Optional networking:
- D18 - RX
- D19 - TX

LCD:
- D20 - SDA
- D21 - SCL

- D37 - LCD backlight control.





SD card: (x1)

SD Vcc	5v
SD GND	GND
SD MOSI	D51
SD MISO	D50
SD SCK	D52
SD CS	D53

5v and GND: + 100nF capacitor

ESP8266 (wifi): (x1)

5v → AMS1117 ~3.3v
- → ESP VCC
- → ESP EN

Decoupling:
- ESP VCC — 10µF — GND
- ESP VCC — 100µF — GND

D18 — 1.8kΩ — ESP RX
- 3.3kΩ — GND

Boot Pins:
- GPIO 0 — 10kΩ — 3.3v
- GPIO 2 — 10kΩ — 3.3v
- EN — 10kΩ — 3.3v

Buttons (x4)

Arduino Pin (D32-35) — Button — GND

Voltage Sensor: (X6)

Battery
- 100kΩ resistor
  - → Arduino pins (A0-11)
  - 33kΩ resistor
    - GND

Master DC Kill (failSafe): (x1)

Arduino D29 — Transistor — DC relay coil

Relay Contacts — Discharge ground path

Temperature Sensors: (x11)

Arduino Pin: D36 — One-wire bus — DS18B20 Sensors — Battery, load bank

Current Sensor: (X6)

Arduino	Sensor
5v	Vcc
GND	GND
Input Pin	OUT

LCD wiring: (x1)

Arduino	LCD
5v	VCC
GND	GND
D20	SDA
D21	SCL

For backlight toggle:

D37 — 4.7kΩ resistor — Base

NPN transistor (2N2222)
- Emitter — GND
- Collector — LCD backlight −
- LCD Backlight + — 5v

LED wiring (x1)

D31 — 1kΩ — +LED− — GND

Approximate Pricing:

Arduino mega 2560 - £45

AC Relay Module (x7) - £10 each

DC Master Relay (24v 150A) (x1) - £20

20x4 LCD - £10

LCD control buttons (x3) - £5 for 3

Hall-effect current sensor (x6) - £8 each

MOSFET - (IRF3205 MOSFET) (x6) - £2 each

NPN transistor drivers (for relays) (x8) - £1 each

Flyback diodes (1N4007) (x8) - £3 for all

SD card module - £5

Wifi module (ESP8226) - £5

Emergency stop button - £5

Reset Button - £5

60A fuses (x6) - £1 each

Temperature sensors - £

33kΩ resistor (precision) (x6) - £

100kΩ resistor (precision) (x6) - £

Gate resistors 100Ω (x6) - £

Load resistors (x36) (6x 3Ω, 100W per loop (6 loops)) - £

Cooling fans (12V) (x6) - £

PC817 (lets arduino check mains) - £4

https://www.amazon.co.uk/gp/product/B00DW6RHN2/ref=ewc_pr_img_1?smid=A3P5ROKL5A1OLE&th=1 - £26.39

Other bits (hard to cost):

Wires, screw terminals, resistors, LED...

Total: approximately £300-£400 (not including the enclosure)