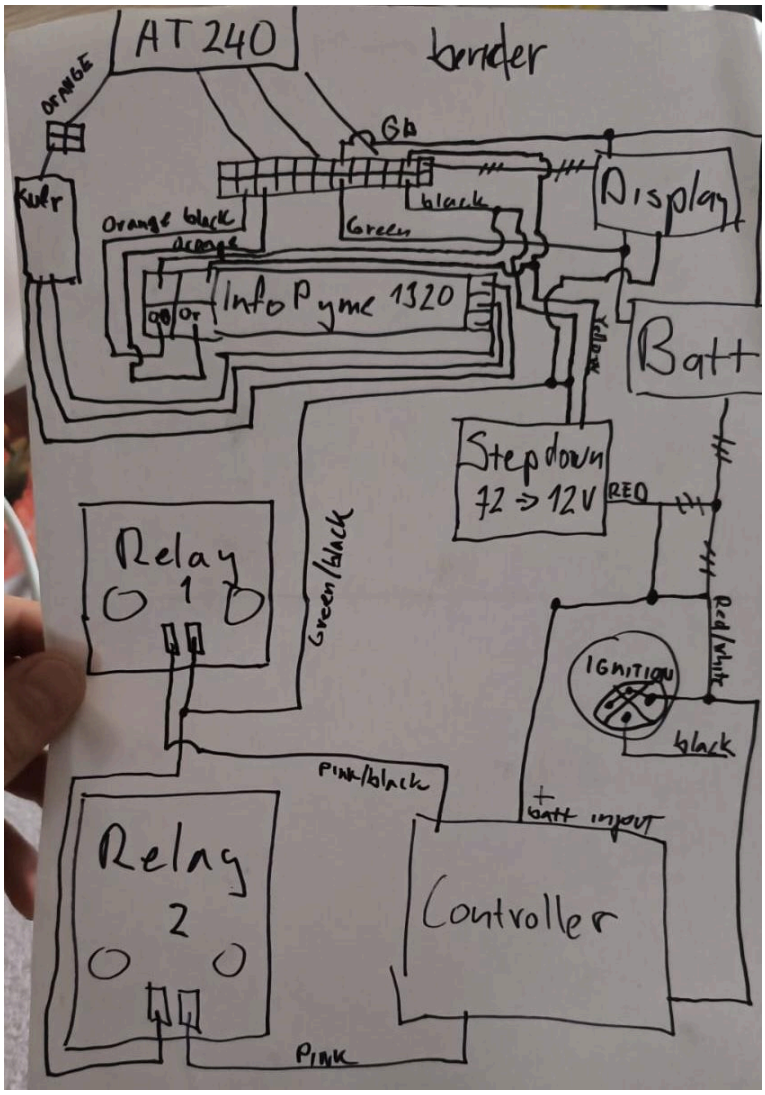
Torrot Muvi Controller Replacement

Selected Controller:
Kellys – sine wave BLDC controller KLS7215N
🔗 **https://kellycontroller.com/shop/brushless/**

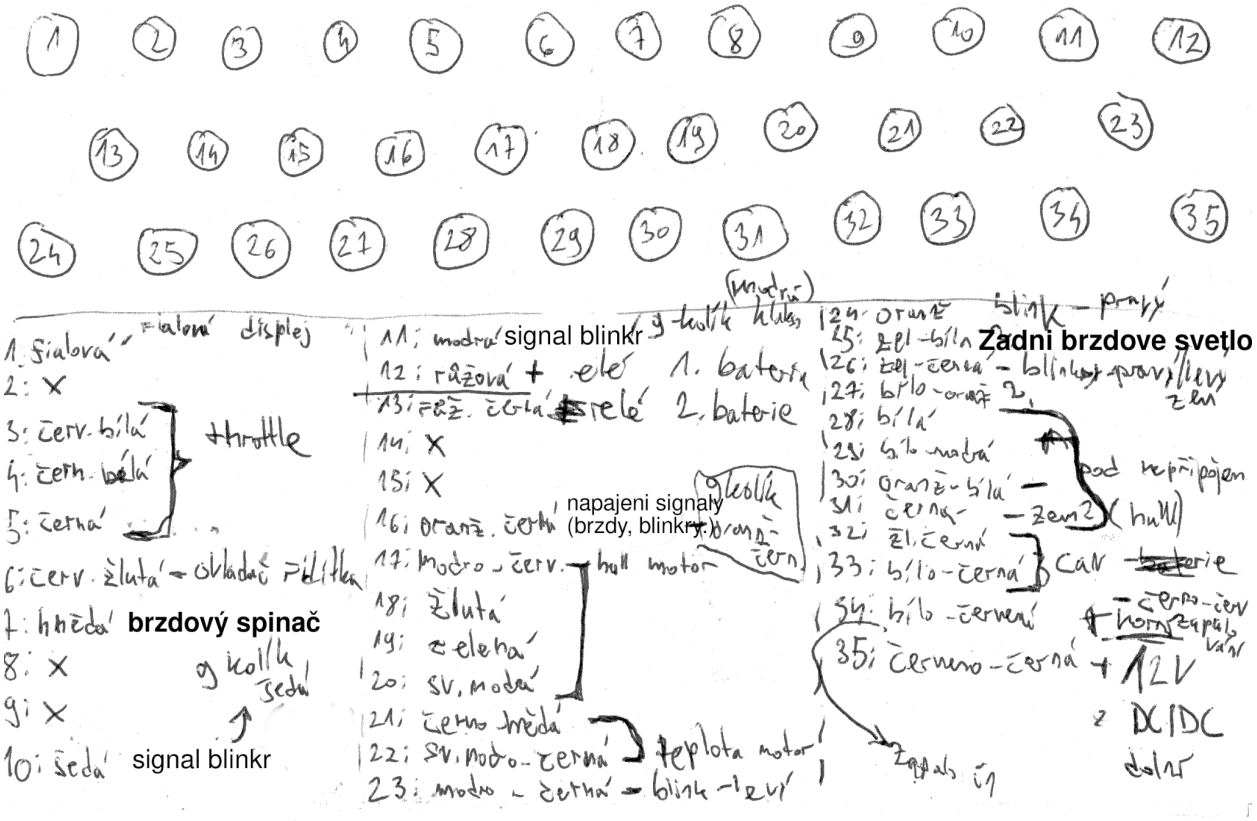
Torrot Architecture:

- The original Torrot controller, display, and batteries communicate over a CAN-bus — *this is not used in the new setup*.
- The scooter contains two DC-DC step-down converters (72V → 12V):
    - One is powered immediately after the battery is connected.
    - The second one (which powers lights, etc.) is only active after the scooter is started.
- Battery 1 is connected via a thin wire to the controller (pin 34) and to one of the DC-DC converters, along with protection components (like GPS, alarm, etc.) — *see the security wiring diagram*. After startup, contactors are closed (controlled by the controller via pins 12 and 13), and the battery is then connected via a thick cable.
- Startup likely works as follows:
  When you turn the key, power is supplied to the display, which most likely sends a signal to the controller. In response, the controller outputs voltage (pin 35) to power the second DC-DC converter. I wasn't able to locate any other source powering that second DC-DC converter.

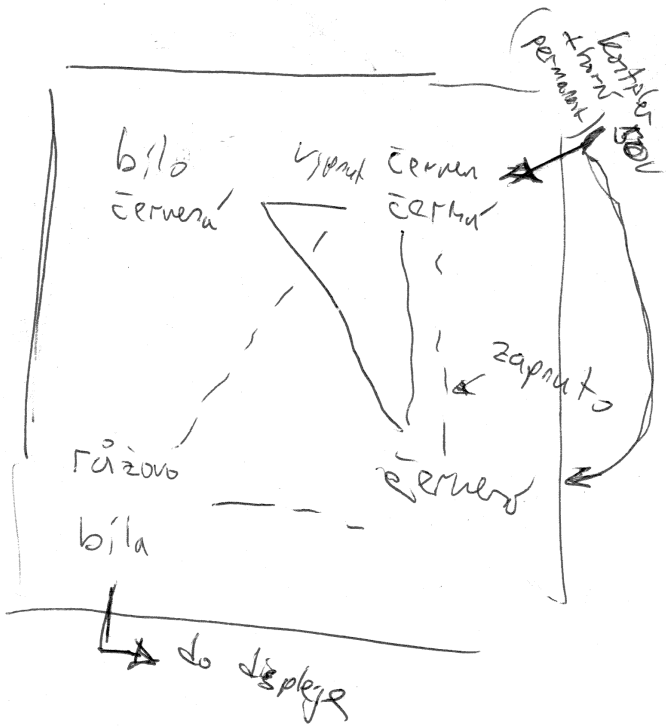
security wiring diagram



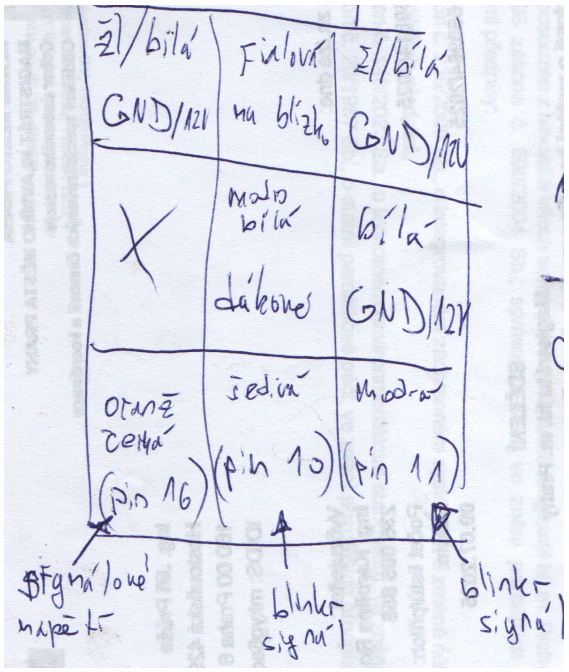
Torrot pin identification:

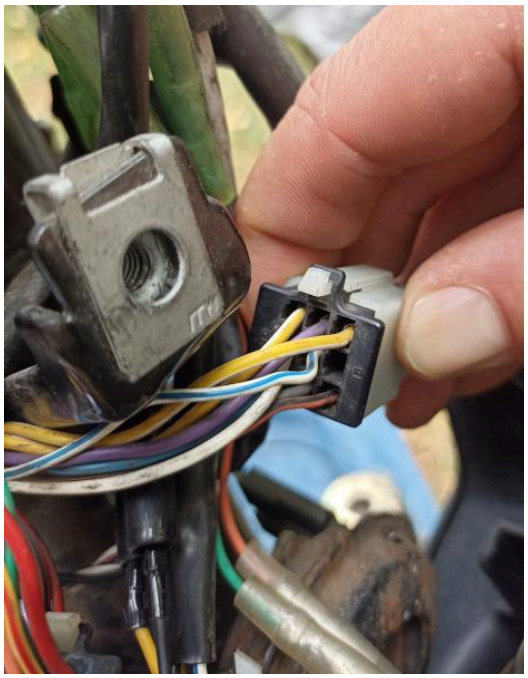


“Ignition” diagram

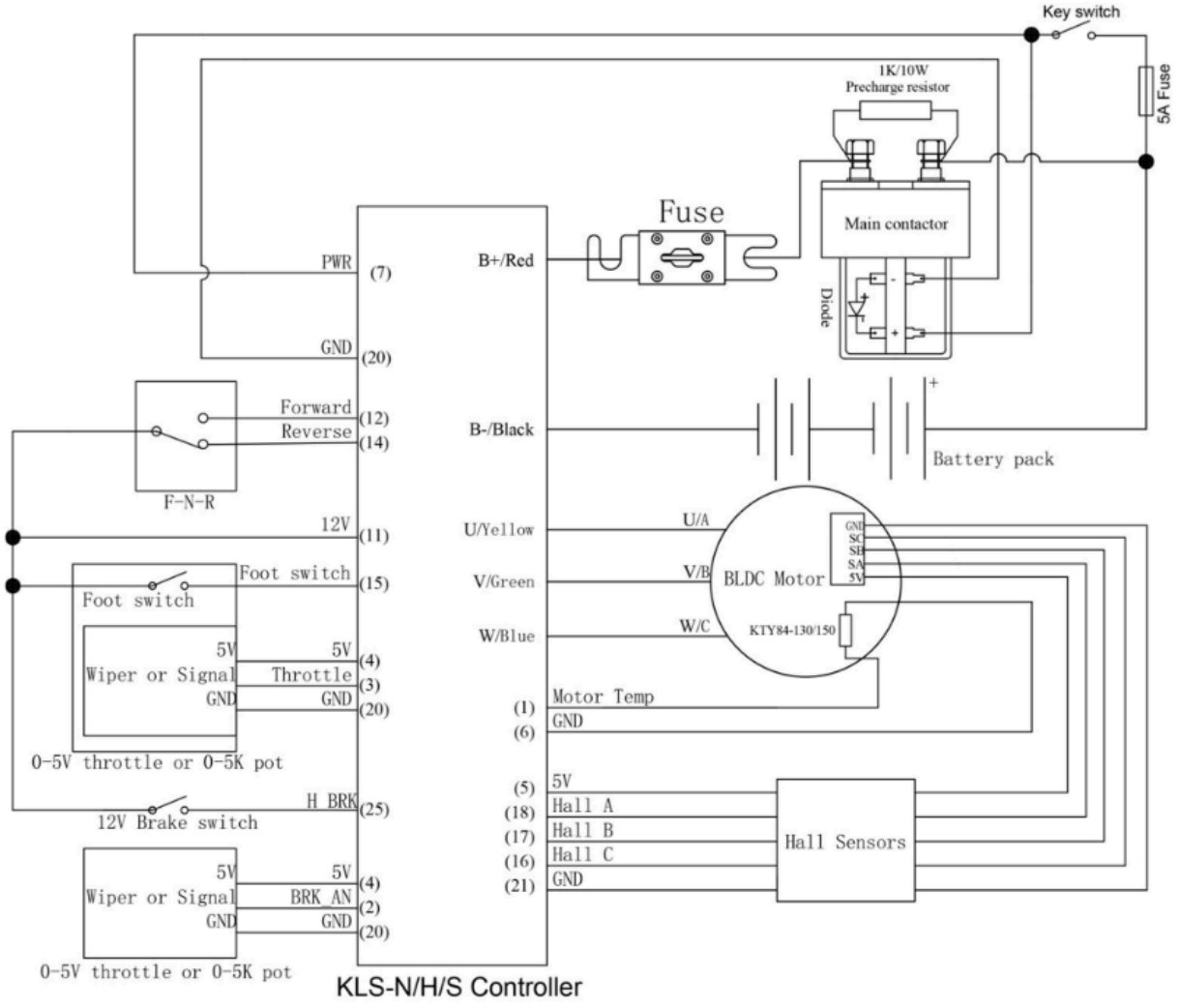


“9pin diagram (blinkers)”



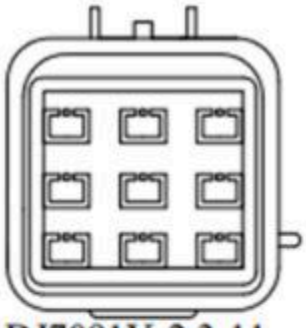


Kellys - KLS7215N
Connection diagram:



Pinout:

A



DJ7091Y-2.3-11
See from output side

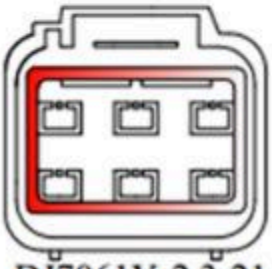
Orange REV-SW (14)	Black GND (6)	White FWD (12)
Red 12V (11)	Yellowish 12V Brake (25)	Blue ECO (22)
Greenish CAN_H (33)	Pink PWR (7)	Brownish CAN_L (34)

B

DJ7091Y-2.3-21
See from output side

Gray Foot_SW (15)	Green Throttle (3)	
Black GND (20)	D-Gray Meter (8)	
Purple 5V (4)	Brown Brake_AN (2)	Red 12V (11)

C



DJ7061Y-2.3-21
See from output side

Black GND (21)	Raddle Temp (1)	Purple 5V (5)
Yellow Hall A (18)	D-Green Hall B (17)	D-Blue Hall C (16)

Conversion:

Connecting the **main cables** (battery power input and motor phase wires) is straightforward — the **terminal sizes and motor cable markings match** well.

To get the controller working, you need to connect at minimum:

- **A – Battery voltage** to **pin 7**
- **B – Throttle control**: connect to **pins 4 (signal), 20 (GND), and 3 (+5V)**
- **C – Hall sensors from the motor**: connect to **pins 5 (power), 21 (GND), and 16, 17, 18 (signal lines)**

Optional but recommended:

- **Brake lever signal** → connect to **pin 25**
- **Motor temperature sensor** → connect to **pin 1**

Additional optional features:

- **Reverse mode switching** → pin **14**
- **ECO mode toggle** → pin **22**

Connecting the **throttle** is simple: just wire up +5V, Signal, GND

These must match the voltage and pin expectations of both the throttle and the controller.

	Torrot	Kellys
+5V	3	**4**
signal	4	**3**
GND	5	**20**

Wiring Hall Sensors and Power Supply

Wiring the **motor hall sensors** is also straightforward — it involves 3 signal wires and power lines.

- On the **Torrot**, the **+5V supply** is on **pin 17** → connect this to **Kellys pin 5**
- Ground is on **Torrot pin 31** → connect to **Kellys pin 21**
- The **signal wires** (Torrot pins 18, 19, 20) should be connected **according to their wire colors** to **Kellys pins 16, 17, 18**

Power Supply Considerations

Power for the controller comes from the **battery**, connected to **Kellys pin 7**. However, **do not use Torrot pin 34** for this. That pin is live **as soon as the battery is connected**, meaning the scooter would consume standby power even when parked — possibly not much, but it's a concern. Instead, I chose a method that follows the **Kellys wiring scheme** - The controller only receives power **after the key switch is turned on**.

Pin 35 is also **not usable**, as it seems it originally powered the lower DC-DC converter. But after the key is turned, **this pin does not have voltage** — so it can't be used to power the controller either.

The Problem: Original Torrot Switching Logic

The Torrot's original power switching setup was **partially handled by the controller**. It controlled power to the **secondary DC-DC converter.** It also controlled the **contactors** that connect the main battery.This logic also seems to involve the **display**, since the pink-white wire in the ignition harness goes to the display.

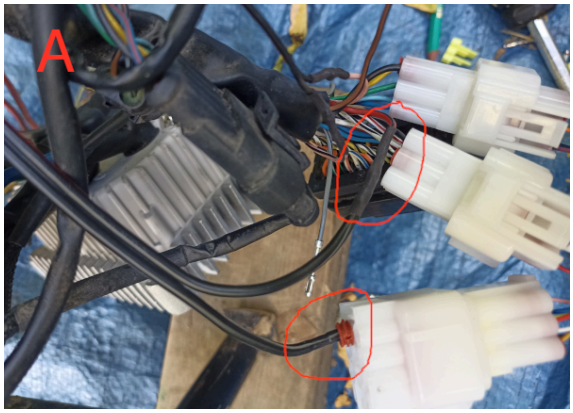
The Solution: Simplified Ignition Switching

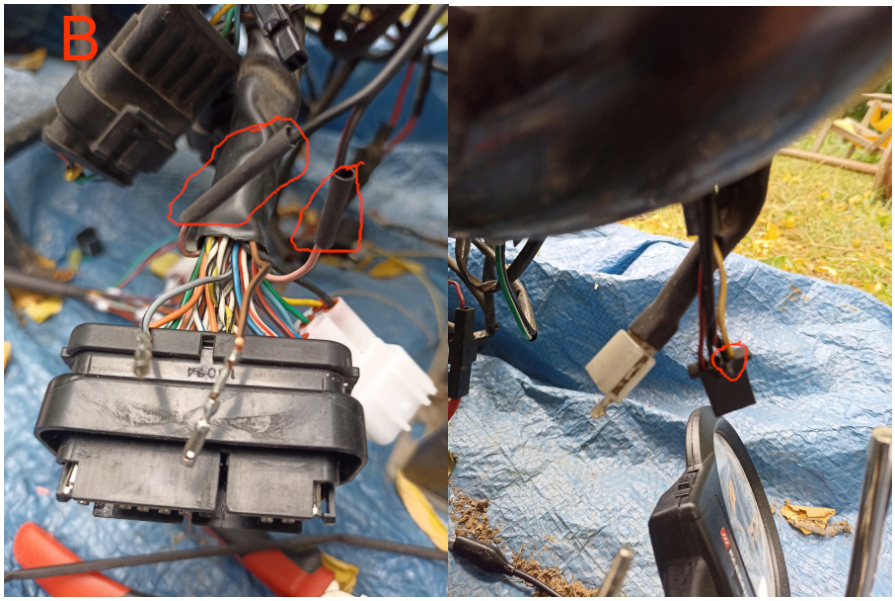
- Replace the **pink-white wire** going to the original display with a wire that sends **voltage from the key switch** to:
    - **(a)** The Kellys controller
    - **(b)** The **secondary DC-DC converter**

- Then, wire the **contactors** to be switched by **12V output from the secondary DC-DC**

This way, turning the key:

- Powers the controller
- Powers the DC-DC converter
- And through it, switches on the contactors — safely connecting the battery





A) Connect **power from the ignition switch** (key switch) to the **Kellys controller**
→ and also use this to **power the secondary DC-DC converter** (this replaces the original function of pin 35).

B) Take the **12V output from the secondary DC-DC converter** (available on the cable near its connector)
→ and route it to the **relay controlling the contactors** (which originally used pins 12 and 13 on the Torrot controller).

Blinkers

The **turn signal switch** on the handlebars sends 12V to a signal wire (orange-black) that goes to the “9-pin” connector — specifically to the **gray and blue wires**, which originally went to **pins 11 and 11** of the controller.

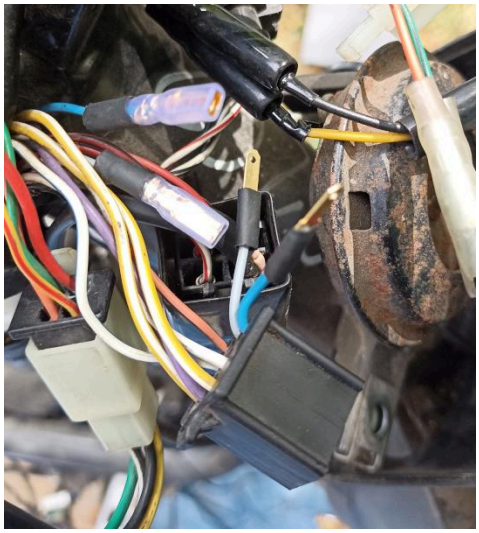
We bring **12V from the Kellys controller (e.g., pin 11 or 11)** to the **signal pin**, and then use a **voltage divider** to step it down from **12V → 5V**, which feeds into an **Arduino input**.

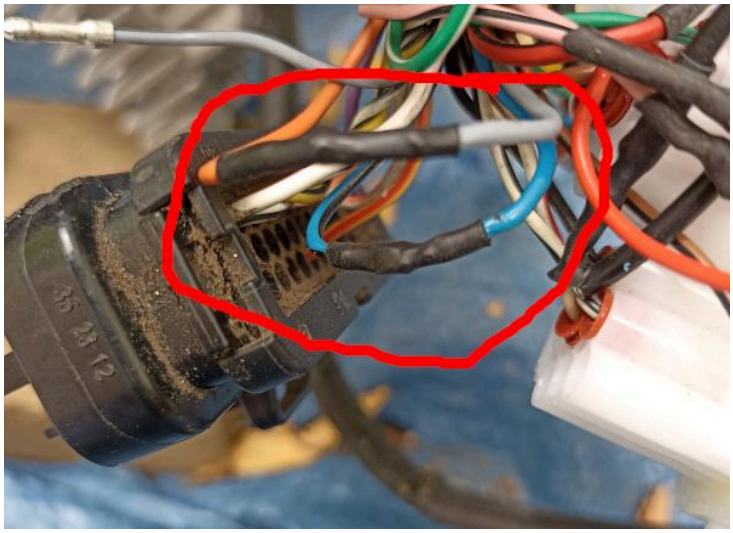
The **blinkers (bulbs)** are grounded through the **original controller’s pin 26**. This ground should now be connected to **Kellys pin 6 or 20**.

Blinkers are switched via the **Arduino output**, which drives **MOSFETs or relays** to supply **12V from the DC-DC converter** to the blinkers.
The control signals go through the **original wires that led to pins 23 and 24**. See photo.

I found it best to **cut the signal wire** from the turn signal switch near the “9-pin” connector on the handlebars, so that I could access **both sides of the wire**.

- The **signal from the switch** goes to an **Arduino input**.
- The **Arduino output** then drives a **relay**, which sends **12V** to the original blinker signal wires.
- These wires lead down to the controller area, where I connected them to the **original signal wires** (10 and 11), which now drive the **relay/MOSFET switching lines** (23 and 24). See photo.





Brake Switch (Lever Brakes)

The **brake switch** on the handlebars is **normally open (OFF)** by default.
When pressed, it sends a **12V signal** (from the same power branch used for the blinkers) to **pin 7**.

This **signal should be connected to Kellys controller pin 25**.

Additionally, the signal should be **split** so it can also be used to **switch the brake lights** (see next section).

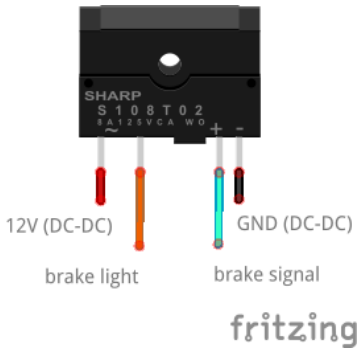
Rear Lights, Brake Lights, and Blinkers

- **Blinkers** do not require separate wiring — the **rear blinkers share the same circuit as the front**.
- The **rear lights** share a **common ground** with the blinkers (see previous "Blinkers" section) — just wire them up and they work.
- The **brake lights** were originally switched by the Torrot controller.

To replace this functionality:

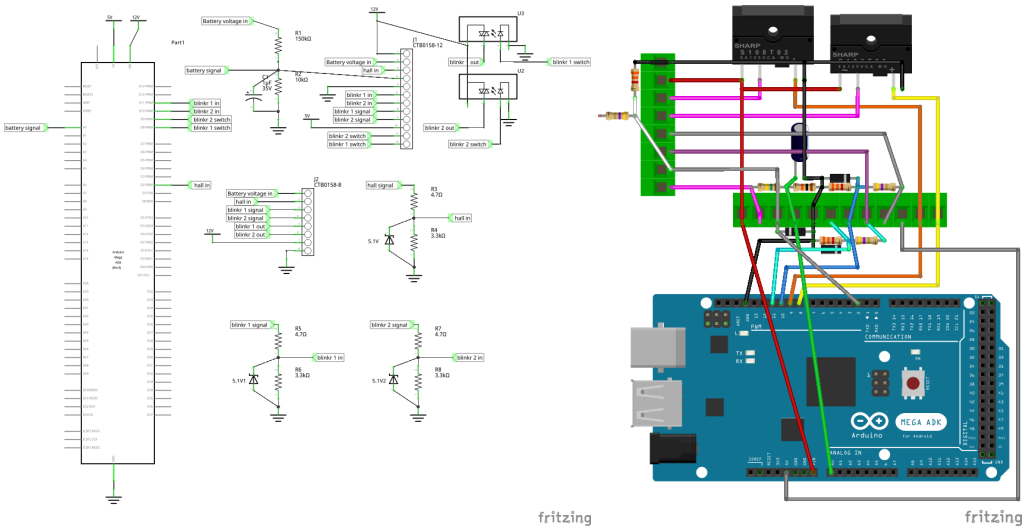
- Either control them via the **Arduino**
- Or (preferably) **split the signal from the brake switch** (brown wire, pin 7 at the connector), and use it to **control a MOSFET or relay**, which then switches **12V from the DC-DC converter** to the brake lights.

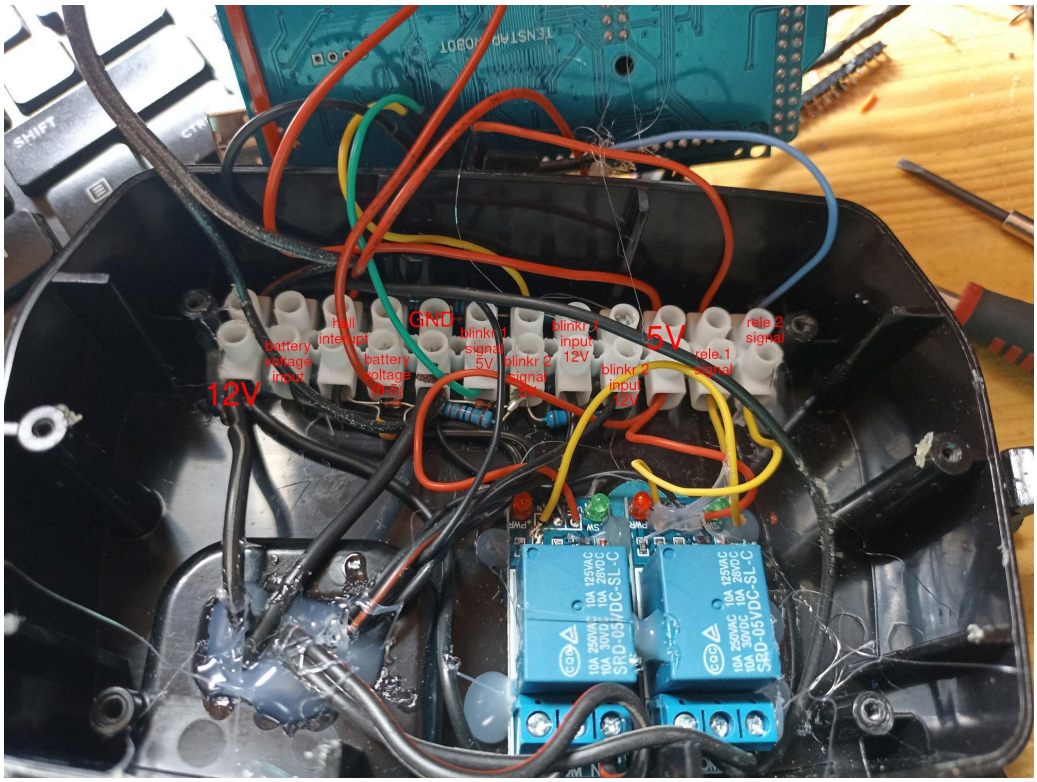
A photo is missing, but a wiring diagram is included instead.



Internal Display Wiring

The original display has been replaced with an **Arduino MEGA**, paired with a display shield - [3.5" 320x480 TFT displej, ILI9486, Arduino Mega shield]
This custom setup serves as the new dashboard display in place of the factory CAN-bus display.





Notes

It may be worth considering **not controlling the blinkers via the Arduino**, but instead using a **dedicated hardware blinker relay** (flasher unit), which would simplify the logic and reduce the load on the microcontroller.

Kellys Controller Configuration

At minimum, the Kellys controller requires two configuration steps:

1. **Automatic identification of Hall sensors**
   → Refer to the Kellys manual, pages 23–24.

2. **Disable regenerative braking**
   → To avoid overvoltage issues, set the parameter **BRK_SW Brk %** to **0** (this can be found on the second page of the configuration software).