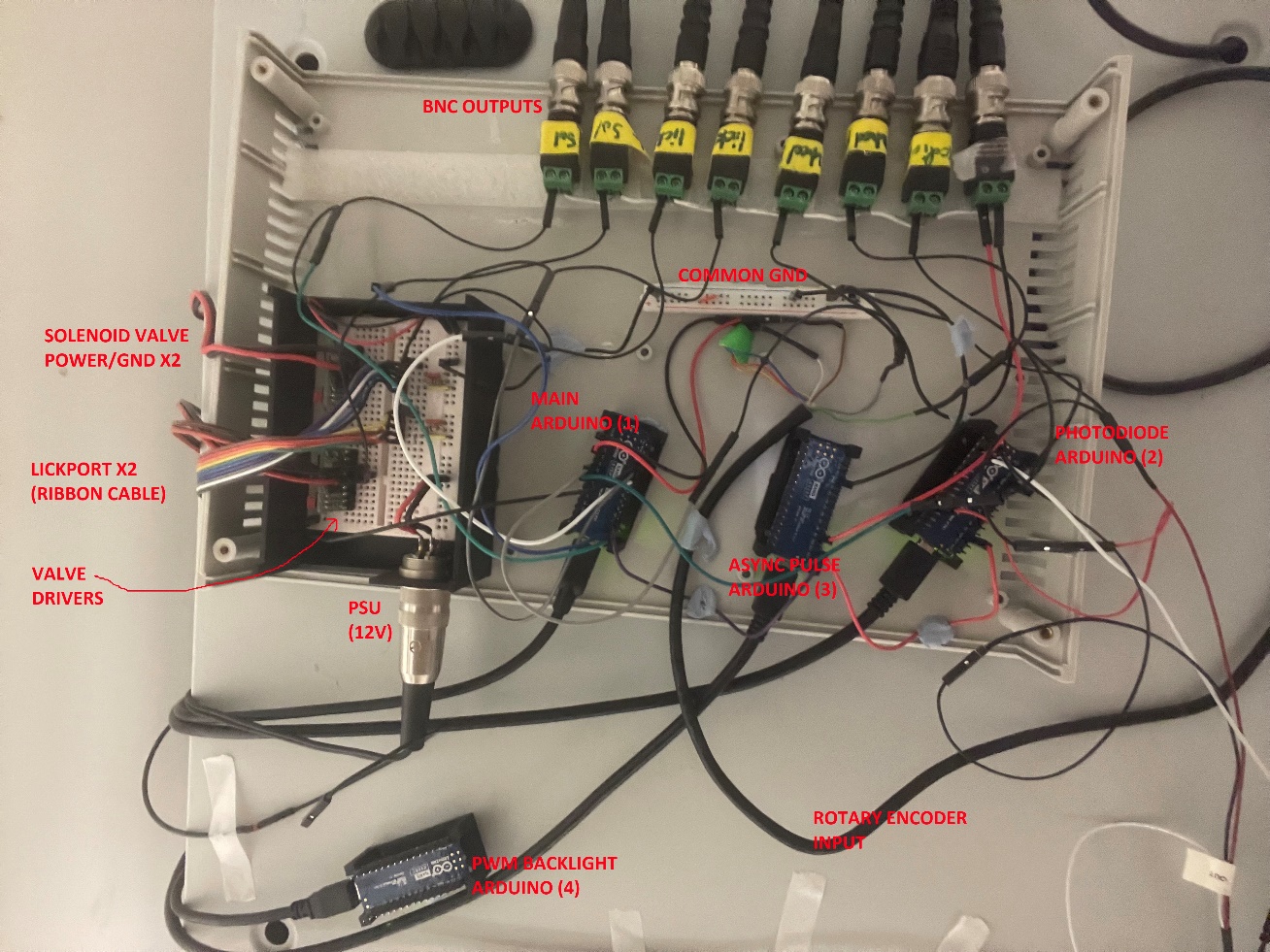
Matrix rigs overview



* We have a number of peripheral componenets which we generally control using Arduino Nano Every’s (w/ male pin headers). We interface with these components using standard 2.54mm pins. We need to be able to interchange these components to clean/replace them.
* If at all possible(?), it would be great if there was some flexibility in terms of connecting new components. In general, other components would probably just need 5V/12V power and have their output split to an Arduino pin and BNC.
* One specific example we’d like to include on the PCB is a simple thermistor circuit – I’ve added this as a component below.

**We have 4 arduinos in total:**

**1.**The main arduino receives the rotary encoder signal, lickport IR sensor outputs (x2) and controls solenoid valves (x2)

**2.** Photodiode arduino which receives the photodiode signal

**3.** Asynchronous sync pulse arduino - this sends a psuedo-random TTL pulse to the main arduino (1) and photodiode arduino (2) and also as a BNC output for aligning signals during analysis. We also use this asynchronous pulse to blink a number of simple LEDs inside the rig to synchronise video frame times. To do this we are just setting multiple digital outputs pins HIGH and LOW simultaneously and using these to control the LEDs.

**4.** Backlight PWM arduino which sends a PWM signal to the displays. (It may not be necessary to incorporate this into the PCB since there is just 1 cable between the displays and the Arduino).

We record the various signals outlined below by sending serial messages from the main Arduino (1) and photodiode Arduino (2) to a PC.

We also send these signals to a National Instruments BNC-2110 via BNC cables for more precise timing alongside our electrophysiology data.

(Current) **BNC outputs:**

1. Asynchronous sync pulse signal

1. Photodiode output
2. Rotary encoder output 1
3. Rotary encoder output 2
4. Left lickport IR sensor
5. Right lickport IR sensor
6. Left solenoid valve trigger copy
7. Right solenoid valve trigger copy
8. (future) Thermistor circuit output

Overview of components

**Power supply**: 12V input

**Lickports**

Quantity: 2 (left/right lick detection)  
each lickport consists of an infrared sensor/LED pair.   
5 pins (male 2.54mm)

Pin 1 gnd PSU (LED)

Pin 2 gnd PSU (sensor)

Pin 3 12V - > 560R -> LED power

Pin 4 12V -> IR sensor power  
Pin 5 IR sensor output (5v)\*  
  
\* we send the IR sensor output to the main Arduino (1) (1 pin for each lickport)  
\* we also send the sensor output to a BNC outputs (1 BNC output for each lickport)

\* we also have an LED connected to the arduino triggers so we can visually see when licks are detected

**Solenoid valve drivers**   
Quantity: 2 (for left/right reward delivery)

9 pins (male, 2.54mm) [we only use pins 1-5 at the moment, but the valve drivers have 9 pins].

1. PSU voltage (12V)

2. PSU ground

3. valve output high

4. Arduino trigger (5V)\*

5. Valve gnd

6. Arduino trigger (5V)

7. Valve output high

8. PSU gnd

9. PSU line voltage  
  
\* each valve Arduino trigger is controlled by the main Arduino (1) (1 pin per valve driver)  
\* we also send a copy of the trigger signals to a BNC output (1 BNC per valve driver)

**Rotary encoder**

Quantity: 1 (measures wheel speed)

4 pins (male, 2.54mm)

Pin 1: 5V (we currently use the main arduino (1) 5V pin to power the rotary encoder)  
Pin 2: gnd

Pins 3 + 4: output (TTL pulses)\*  
  
\* we output these 2 pins to the main Arduino (1)  
\* we output these 2 pins to 2 BNC outputs

**Photodiode**

Quantity: 1 (to measure events on displays)

3 pins (male, 2.54mm)

Pin 1: 5V power (we power using the photodiode arduino 5V at the moment)

Pin 2: analogue output (0-5V)\*  
Pin 3: ground  
  
\* we output this to the Photodiode Arduino (2)   
\* we also send this output to a BNC

**Infrared lights**

Quantity: 2+ (we currently use 2 but may want increase this at a later time)  
male pins

Pin 1: 12V

Pin 2: Gnd

**Thermistor circuit**

Quantity: 1 (to measure breathing)

We have a 22kOhm thermistor (<https://www.digikey.co.uk/en/products/detail/te-connectivity-measurement-specialties/GAG22K7MCD419/5277253>) that we plan to use to measure the breathing.

We plan to use a circuit like the one listed here (<https://learn.adafruit.com/thermistor/using-a-thermistor>) using the Arduino 3.3V/AREF as VCC and split the output to an analogue input of the main Arduino (1) and to a BNC.   
I will test this works ASAP in case we need to make any small changes

**Display backlight PWM control**

Direct from an arduino nano every PWM pin (Backlight Arduino)

**Extras**

It would be useful to have some spare 12V/gnd and 5V/gnd pins to connect to in case we want to introduce any additional components