Ecole Polytechnique de Bruxelles

University of Leeds

Johnston Lab Internship

# Technical description and User guide: Lick-Sensor device for behavioral rodent experiments

*Author:*
Maxime Verstraeten

*Supervisor:*
Jamie Johnston

September 16, 2019

# Contents

## List of Figures

# 1 Introduction

This user guide intends to provide the user with the information required to use and understand the lick-sensor device developed at JohnstonLab as well as giving some clues regarding troubleshooting.

This device was developed as part of my internship at JohnstonLab (Leeds University, UK) during my master's years as a student engineer from the Université Libre de Bruxelles, Belgium.

This device is part of the rig that allows the lab to perform in-vivo measurements and experiments on mice. This device has been designed to be able to perform behavioral experiments with a reward system in relation to different stimuli such as odors.

Note: The workflow, thoughts, issues, ... for this project are discussed in my OneNote.

The latest software and documentation updates are available on my Github: https://github.com/maverstr/Lick-Sensor

# 2 Objective

The lick-sensor device is intended to be use for mice behavioral studies in a kind of reward system with sugared water. The device allows a mouse to lick a needle which shall deliver some water through a valve. The amount delivered can be setup logically with an Arduino code as well as the conditions required for delivery (e.g. the minimum delay between two deliveries, in case of specific odors delivered, ...).

# 3 Hardware

## 3.1 Water Tank

The water should be stored higher than the delivery as to provide pressure for the water to flow through the valve.

## 3.2 Valve

The chosen valve is a solenoid valve VDW22LA for its size, price and speed of actuation. It is working with 24V DC and as it is a solenoid valve, it will create peak voltage and current when switching it off/on so care must be taken to cope with this issue both on the hardware and software sides.

## 3.3 Enclosure

The device comes with its enclosure which implements a switch, a push-button as well as a BNC input for valve activation, 4 BNC connectors (24V valve, GND valve, sensor+, sensor -), a 9V output for the fan and 2 BNC outputs for the licking signal and the valve activation signal. Note: Sensor+ is connected to 9V (hence "+") while Sensor- is connected to ground through a series of resistors. It is NOT a differential signals. Just a denomination for two wires coming onto two different pads on the circuit.

## 3.4 Microcontroller

The $\mu$C used is an Arduino Uno. The PCB is designed to be stacked on top of it so that each PCB pin is correctly connected to the Arduino.

## 3.5 Fan

Because of the use of linear voltage regulators (at least in early designs), a fan had to be added to the enclosure. It is a simple fan MC002111 working on max 12V DC. The fan is no longer needed with the latest hardware release as it uses switching DC/DC converter.

# 4 Software

The Arduino has been programmed using Arduino IDE.

It works around interrupts setting flags which are being polled every loop of the main loop. The touch sensor circuit, once powered, provides a 3.3V TLL active low signal connected to pin 2 of the Arduino. Once a falling edge is detected, an interrupt is triggered which sets a flag to True. In the loop, the flag will be reset and, if the conditions are met, an appropriate duration pulse is output to pin 6. This 3.3V signal will then be sent to another reed relay functioning with 24V. This will therefore activate the valve for the specific duration.

It works the same way for the switch, push-button and BNC input (i.e. interrupt, flag, pulse output).

# 5 How to use?

The device is simple to use.

It should be powered through the DC socket with a 24V DC supply. The bullet connectors shall be connected to the crocodile clamps of the sensor and the bullet connectors of the valve. Obviously, correct polarity must be observed for the valve (Red cable is 24V, black cable is ground for both the wires coming from the PCB and the valve). If the device is overheating (which it should not with the new DC switching regulator), you can power the fan by connecting the two battery plugs.

The valve is activated when both sensors are touched at the same time or when the push-button is pressed or the switch is in up position.

The button can be used to manually set a droplet on the needle, the switch for a long time activation (clearing the water tank, filling the void in the pipes, ...) and the BNC input can be triggered by another script such as a Python script that would randomize odours and associate the odours with a droplet.

# 6 Expected Results

The valve should open briefly when the two sensors are touched or when the push-button is pressed.

# 7 Electronic Circuits

## 7.1 Sensor circuit

### 7.1.1 Original circuit

The basic sensor circuit is based of a paper (Slotnick, A simple 2-transistor touch or lick detector circuit, 2009) and allows to send a TTL output (by closing a reed relay) whenever an animal touches two wires at the same time.

By analysing the paper and the circuit we note that

- since the animal subject must survive the experiment and ideally not notice the lick sensor, the current flowing through the rodent should be minimal..
- A battery-operated circuit makes it independent from a DC power supply and an AC line and thus does not require isolation transformer (or optocoupler, ...) to protect the rodent.
- The circuit proposed here above is using a 9V battery, the current flowing is less than 1$\mu$A (according to the paper) and the output operates a relay that can then act as a switch to operate any microcontroller, whatever the voltage required for the logic inputs and therefore not requires level shifting (directly outputs TTL-compatible pulses)
- According to the paper, a 9V alkaline battery is sufficient for a couple of months of experiment. 5.5V is the minimum battery voltage for reliable operation. A 9V lithium battery could have an extended lifetime.

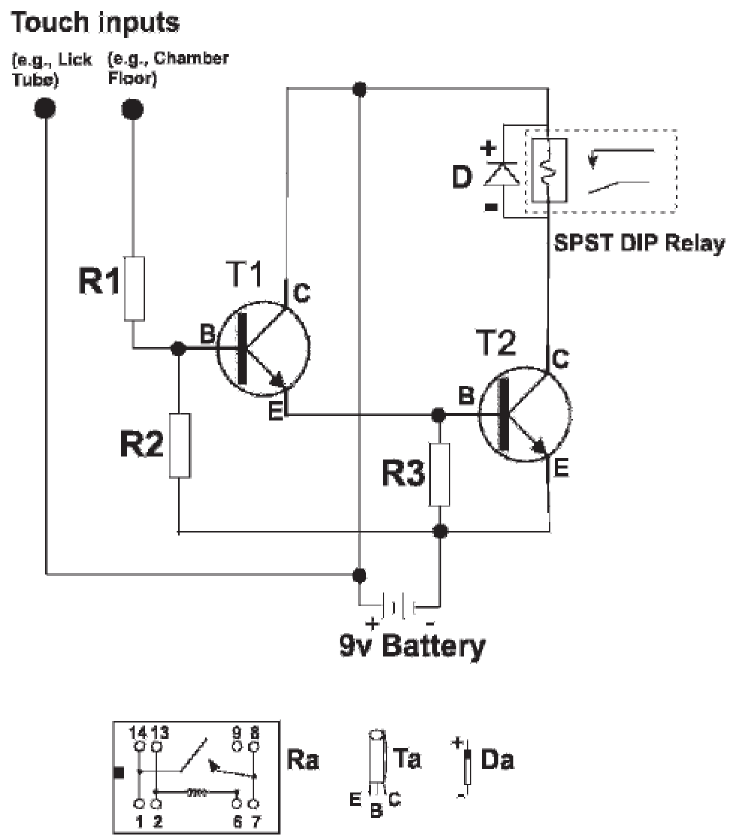



Figure 1: Sensor circuit

- The battery could also be replaced by a Zener in 6.3V from a 12 or 24V DC power supply which allows unlimited lifetime and controlled voltage or from a voltage regulator or 9V DC supply.

#### 7.1.2 Actual implementation: First design

The sensor circuit is only a small part of the whole circuit shown below. Moreover, I decided to add a pull-down at the output of the TTL relay and add some decoupling capacitors on the circuit. The BJTs used are PN2222A and the relays are COTO 9007-05-01 SPST.

#### 7.1.3 Actual implementation: Second design

After testing, I decided the relay wasn't reliable enough and expensive. I designed a new circuit that can switch faster, is cheaper and more reliable.

It has been implemented in the V3 and the circuit can be seen on Figure 4.

### 7.2 DC/DC Converter

The valves require 24V DC and the Arduino is powered by 9V on Vin pin. It can then be used to get 5V and 3.3V.

To avoid getting a supply for 24V and 9V, a single DC Supply 24V is used and a TSR 1-2490 DC/DC switching regulator is used to provide 9V. Circuit is shown on Figure 5.

Early design implemented linear voltage regulators but they tended to overheat and required the use of a fan.

### 7.3 Logical part, $\mu$C

The logical part is handled by an Arduino Uno.

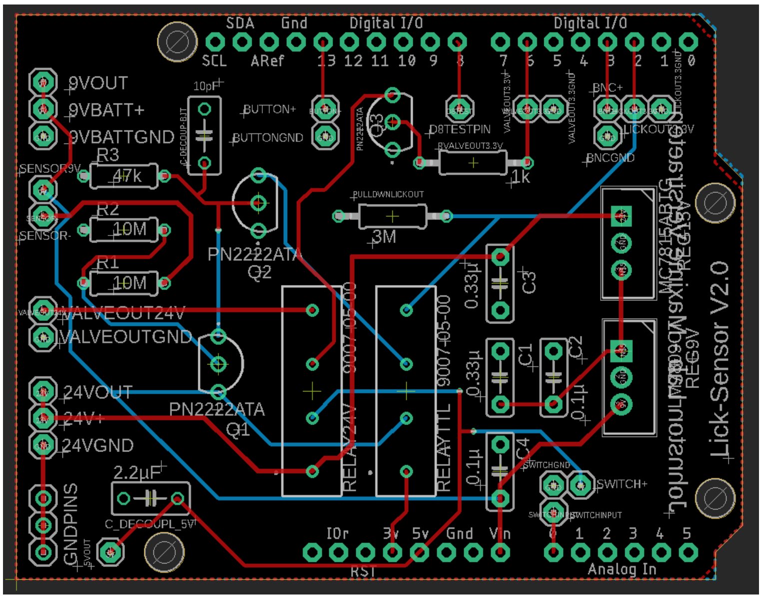



Figure 2: PCB View

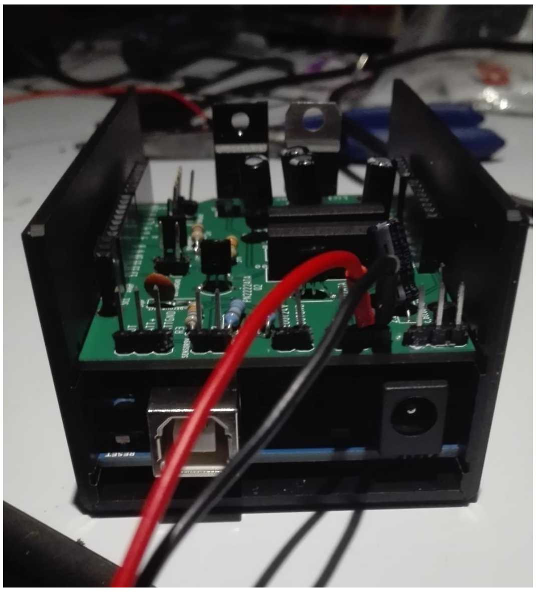


Figure 3: View of the PCB on top of the Arduino

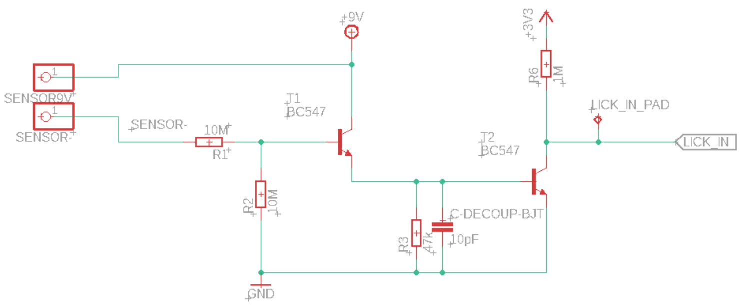



Figure 4: Improved sensor circuit

The pinout is shown in Figure 6.

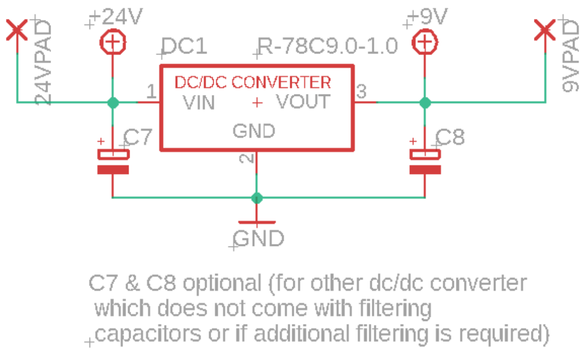



Figure 5: DC/DC Converter circuit

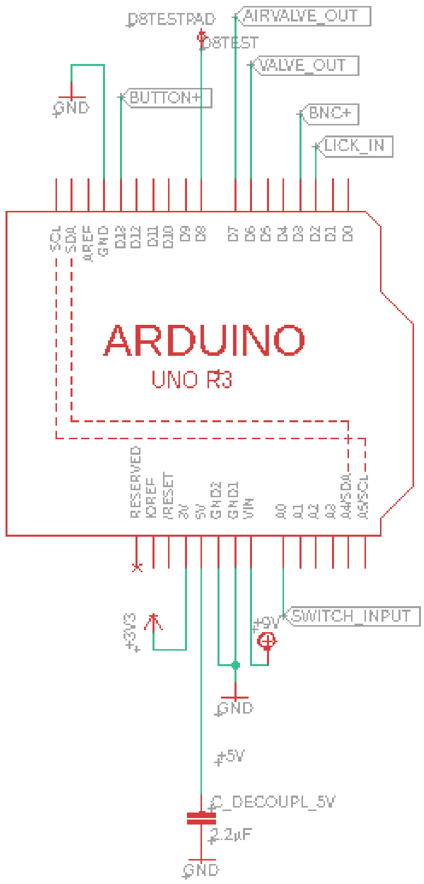



Figure 6: Arduino pinout

## 7.4 Valve Actuator circuit

The valve actuator circuit is composed of a digital input (3.3V, coming from the Arduino digital output pin) which is the Arduino command for opening the valve. R5 and R4 protect the FET gate and provide a way for the gate capacitance to discharge rapidly.

This part is shown on Figure 7.

When 3.3V signal is given, the FET opens and connects the solenoid valve ground pin to actual ground. Current can then flow through the solenoid and activate the valve. This is basically sinking-current activation.

A fly-back diode is extremely important to protect the circuit against peak voltage when switching the valve off.

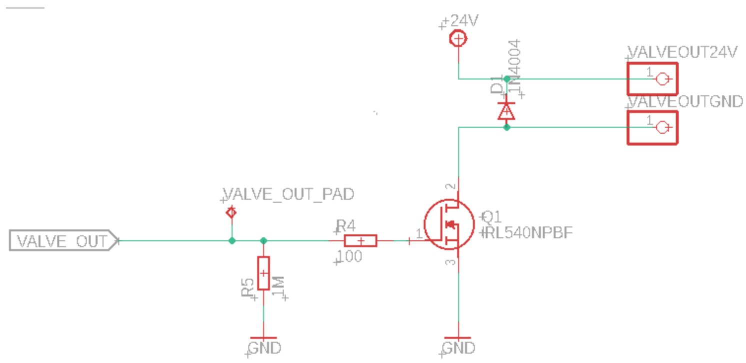



Figure 7: Valve actuator circuit

### 7.5 Adding a second valve for droplet removal

It has been discussed later that to be more flexible in behavioral experiments, it would be interesting to add a new valve to drain the droplet that may stay on the needle if the mouse doesn't lick it before a certain time. Indeed, if the droplet is triggered by an odor and we want to train the mouse to associate this odor with food (or reward, i.e. sugary water) the droplet can't stay (if it isn't licked) and another odor is sent or the mouse will not correctly associate the odors with the correct reward.

To drain the droplet, we must simply open the circuit to air to stop the capillarity effect. The water pipe is therefore connected to a second pipe with the other valve. This valve is the same as the first one (VDW22LA) as water will flow through to be drained and it reduces the number of different components (Plumbing drawing can be seen on Figure 8).

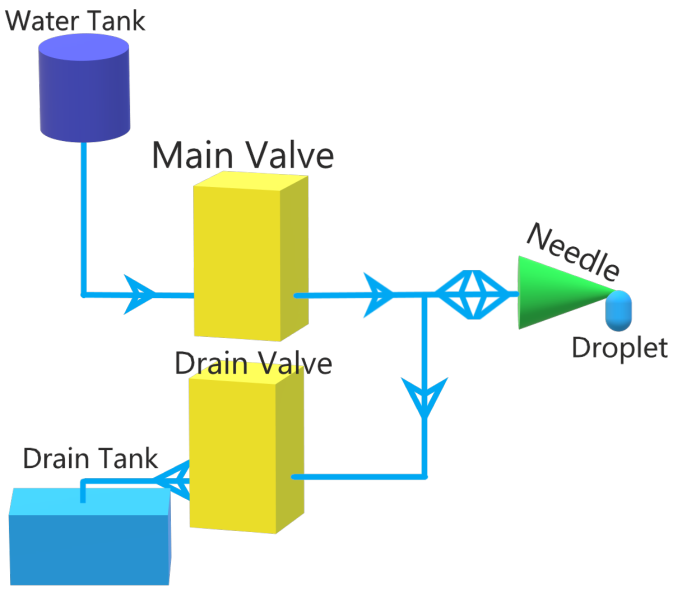



Figure 8: Plumbing Map

This requires to design a second valve circuit actuator. (the same as shown in section 7.4. It was the opportunity to implement some of the improvements discussed.

This new design is shown in section 7.8.

### 7.6 Full design V1

The full circuit has shown some good results but the relays (especially the one driving the valve) is not reliable enough and needs to be replaced every thousand actuations. Moreover, this V1 design used linear voltage regulators which tended to overheat. This design also didn't implemented a direct way to avoid peak voltage on the load. (a snubber RC circuit has been added later on when testing the PCB).

This circuit is shown in appendix 2.

### 7.7 Full design V2

A second version of the circuit has been designed but not produced. It implemented solutions to some of the problems presented here above.

The 24V relay is replaced by a FET and a flyback diode is added on the load. The schematic is shown in section C.

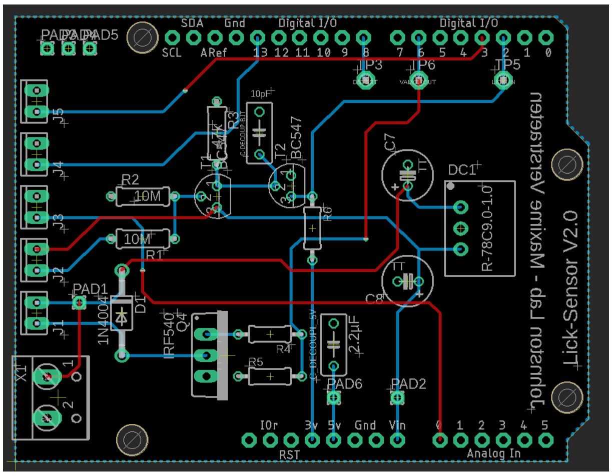



Figure 9: PCB V2

### 7.8 Full design V3

To add the new valve and bring some of the improvements discussed here-above, a third version of the PCB has been designed and produced.

The new circuit implements, among other things:

- An open collector output signal for the motion sensor. This increases the reliability of this part of the circuit as well as its lifetime expectancy and reduce the cost because the Reed relay is replaced by a BJT.
- A MOSFET driven actuation of the valve (IRL540) which greatly increases the reliability (the reed relay was especially sensitive at 24V with peak voltage due to the solenoid valve). This does not reduce the cost as logic-level FET is expensive ( £2 for IRL540 on RS and 1.15 for a relay on Farnell).
- Additional fly-back diode instead of RC circuit for the valve recovery. Should be more effective.
- Capacitors C7 and C8 are optional. Indeed, the TSR 1-2490 DC/DC converter comes with integrated filtering capacitors. However, if it is replaced by another one that does not comes with capacitors, the pad is present on the PCB. Moreover, they can still be added if additional filtering is required.

The Eagle files are available on request and the schematic is shown in appendix D. The PCB is shown on Figure 10.

## 8 Improvements

### 8.1 Zener Diode

A Zener diode can be used to control the voltage and make sure that no change in sensitivity appears as voltage decreases in case of a battery-powered application. It could also be used as a voltage regulator for the DC Supply.

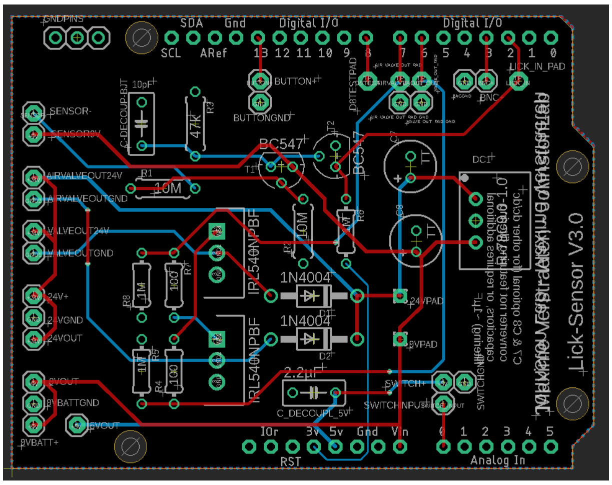



Figure 10: PCB V3

Another Zener diode coud also be added at the MOSFET gate to protect it from high voltage. In normal operations though, the gate is connected to the Arduino pin which will not output more than 3.3V and 150mA max. Still, ESD could happen so it would be good practice to add it, even though the Arduino also probably features ESD-protection diodes.

## 8.2 Reduce the current flowing through the mouse

R1 and R2 could be increased (20M ohms) to reduce current flowing through the subject but reduces reliability. TODO MEASURE CURRENT FLOWING

## 8.3 FET instead of BJT ?

Transistors used in the sensor design are BJT wich are okay in a low current application but FET could probably do, even if the voltage-controlled is less convenient than BJT's current-controlled. The MOSFETs present high input impedance, less fabrication dispersion, very small, easy to scale. BJTs present input capacitance (bad for HF), higher output resistance (not suited for low-impedance load), lower gain/stage so SNR quickly degrades, current-operated -> higher consumption

In the end, we could use MOSFET to reduce the consumption but this is not necessary. Additionaly, FETs are harder to use at logic gate signal (need to use FETs like IRL540 which are more expensive).

## 8.4 Avoid shorts with stackable design

One of the issues I ran into, was that my power pins were spatially close (on top) of the Arduino USB Frame (connected to ground). This may have cause a short-circuit between 24V or 9V and Ground. One solution would be to spatially move those pins in another place. The other solution which I did was to add one more layer of stackable header to further separate the PCB and the $\mu$C. This also allows more air to cool both the PCB and the Arduino down.

## 8.5 MTA connectors

Once the design is tested and validated, we can also implement some MTA connectors instead of the pin headers. This will however increase the cost of the design but it might be interesting if the device is shared to other people.

## 8.6 Photobeam detection

By implementing a photobeam, the animal can signal it is ready for stimulation by approaching the sensor without even licking it. This data can then be used to logically control the water tank valve.

## 8.7 Shielded lines

The solenoid valve may create some noise for other devices in the lab as well as pick up some noise. Shielded lines can be included if needed.

# 9 Troubleshooting

Note: several ground pins are available on the PCB for easy measurements.

## 9.1 The valve doesn't activate

Make sure that the device is powered by 24V DC. Check if the valve is opened or closed by putting water in the water tank above. If it is open check the subsection 9.2. If it stays closed, check with an oscilloscope (or voltmeter) if 24V is sent to the valve when push-button is pressed. If it is the case and no water is flowing through, the valve might be blocked due to residues (sugars, ions deposit, ... in the valve). Simply try to apply more pressure to the water (blowing hard on the water syringe might do the trick) to try and unclog it.

## 9.2 The valve stays open

If the valve is always open, check that the switch on the box is on low position. Indeed, if it is up, the valve will stay open.

If the valve still stays open, check with an oscilloscope if a 24V signal is sent to the valve (you can check the BNC output). If the switch is low, and no button is pressed, the signal should be ground If it stays to 24V no matter what, the reed relay might be broken (fused in closed position). In such cases, the reed relay might need replacement. Note: ideally, a socket should be placed on the PCB to avoid the need to (un)solder the relay everytime which tends to damage the PCB. Note2: a new design with another BJT instead of a relay is discussed in section **??**.

## 9.3 Push-button/switch doesn't work

Unscrew the box and check that the wires (2 for the push-button (digital signal pulled up and ground); 3 for the switch (5V, ground, signal)) are still in place both on the button itself and on the PCB pins. If so, plug the USB into the Arduino and open the Serial monitor. When a button is activated, a corresponding message should appear in the Serial monitor (Note: the Serial should be set to 500000 bauds).

## 9.4 Is there a polarity for the sensor?

One of the sensor is connected to 9V and therefore should not be touching any ground. The other sensor is connected to ground through a series of resistors.

# A Power dissipation by linear voltage regulators

The first design used linear voltage regulators to transform the 24V to 9V able to power the Arduino. They transform excessive voltage in heat. Therefore the power dissipated = ((24V-15V)+(15V-9V))*Imax Imax = 1.095V/20ohms (current measurement with voltage meter) = 55mA.

Safety measure : 100mA.

So we get power dissipated = 1.5W Or 900mW for the first regulator and 600mW for the second one. The regulators comes within a TO-220 package which is fine for under 1W of dissipation (rule of thumb). However, in a closed box, there will be no airflow.

The datasheet of the regulators specify 60°C/W as thermal resistance for junction/ambient.

**Table 2: Thermal data**

| Symbol | Parameter | D²PAK | DPAK | TO-220 | TO-220FP | Unit |
|---|---|---|---|---|---|---|
| $R_{thJC}$ | Thermal resistance junction-case | 3 | 8 | 5 | 5 | °C/W |
| $R_{thJA}$ | Thermal resistance junction-ambient | 62.5 | 100 | 50 | 60 | °C/W |

Figure 11: Thermal Data

So at 900mW it's 54°C above ambient. Worst case ambient would be 40°C room temp + enclosed box.. Estimates about 60°C. That makes a total of 104°C.

The datasheet mentions that the maximum operation temperature is 125°C. So we're just under.

**Table 1: Absolute maximum ratings**

| Symbol | Parameter | | Value | Unit |
|---|---|---|---|---|
| $V_I$ | DC input voltage | for $V_O$= 5 to 18 V | 35 | V |
| | | for $V_O$= 20, 24 V | 40 | |
| $I_O$ | Output current | | Internally limited | |
| $P_D$ | Power dissipation | | Internally limited | |
| $T_{STG}$ | Storage temperature range | | -65 to 150 | °C |
| $T_{OP}$ | Operating junction temperature range | for L78xxC, L78xxAC | 0 to 125 | °C |
| | | for L78xxAB | -40 to 125 | |

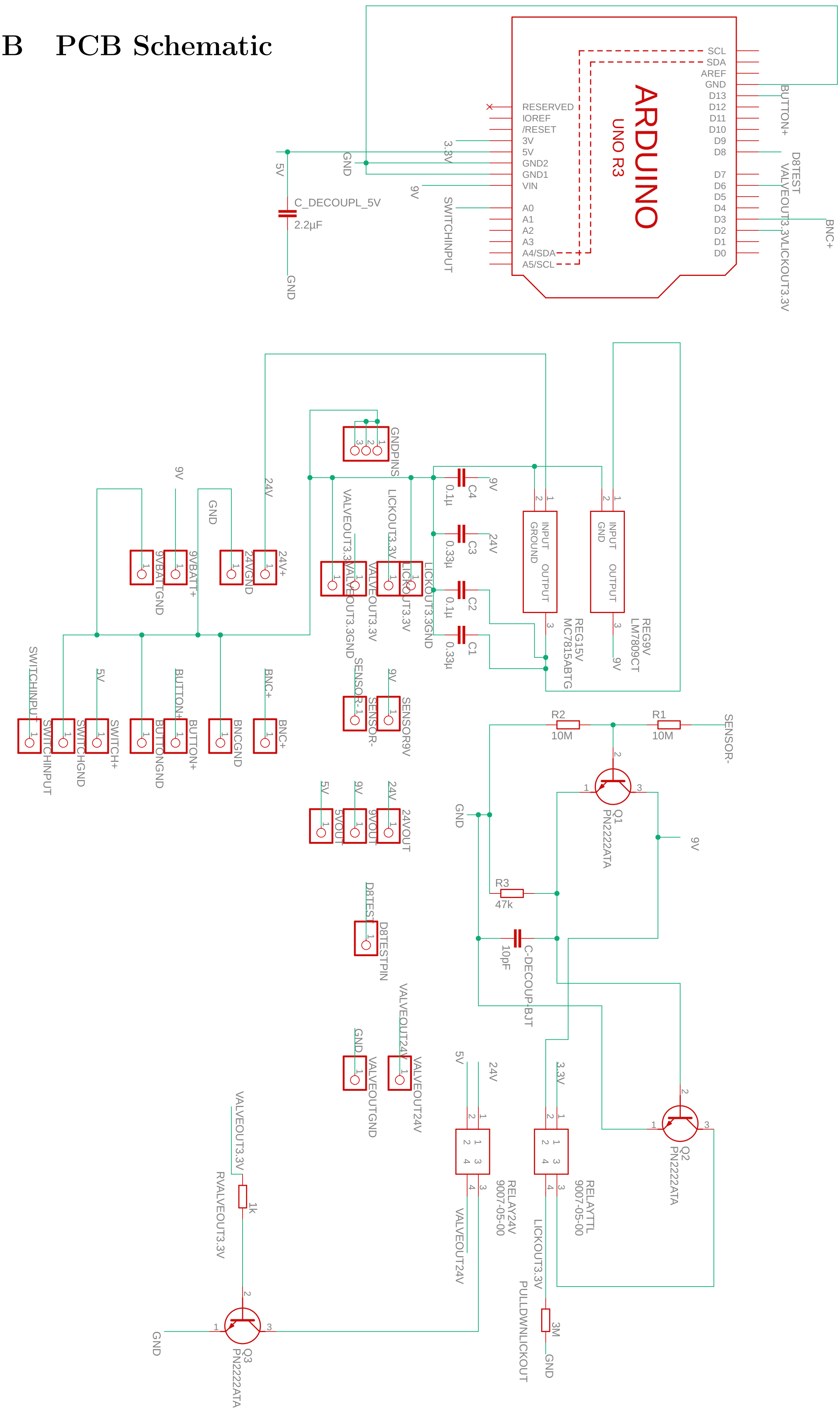
Figure 12: Maximum operating temperature

The idea was therefore to drill some holes for the air to flow through the box and monitor the temperature. If it rose too high, a fan was to be added (required 9V and 5V pins were already present on the PCB).

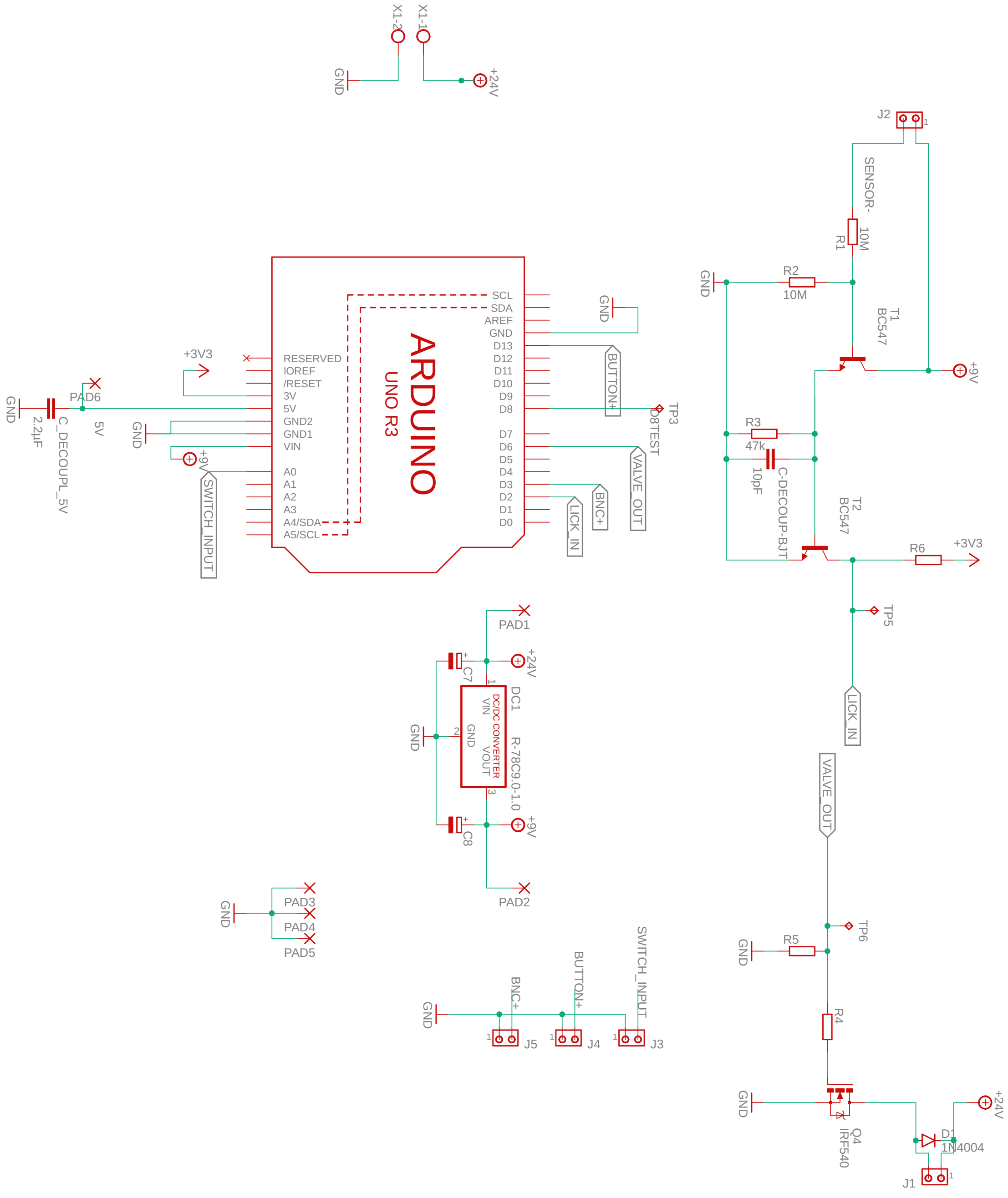
On the PCB: I obtained 1.130V/20ohms = 57mA, as expected.

A simpler and more reliable solution (as the device is used with other devices around, for long term use, by people without skills in electronics) was to replace the linear regulators by DC/DC switching regulators, way more efficient but more expensive.

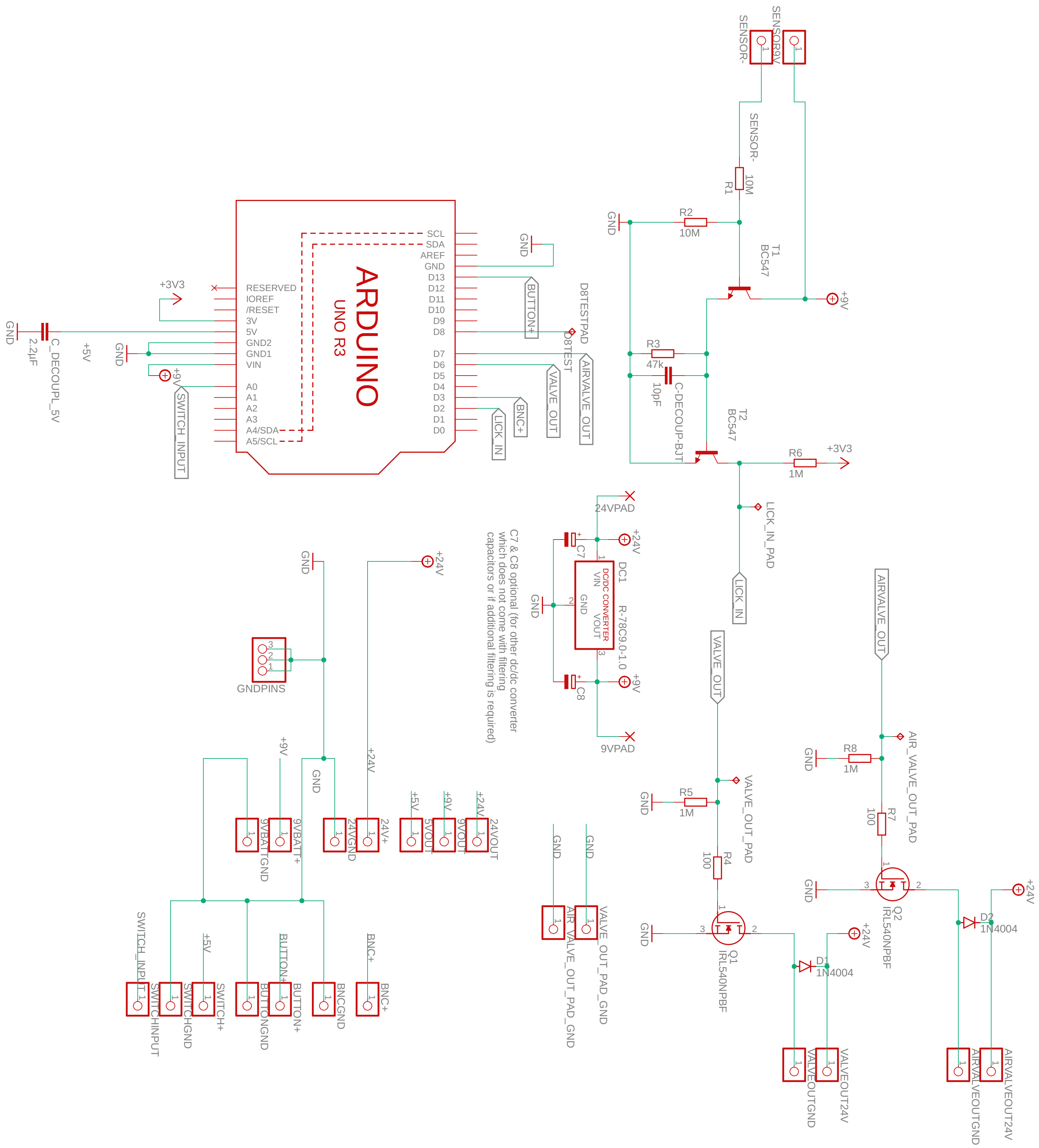
# B PCB Schematic



# C PCB Schematic v2



# D PCB Schematic v3



## References