Doc for the pellet dispenser's control box

Please plug the USB on a powered computer before pluging the power/data cable

Neurospin

In order to get a pellet dispenser to work we had to build the control box for it, here's its documentation from the wiring layers to the programming stuff. Hopefully if you read this it's just for fun, but if something is broken be reassured, the design is simple enough and fixing it shouldn't be too too hard.

Contents

Introduction

This box is designed to control the ENV-203IR Pellet Dispenser from Med Associates Inc., and it's not official and relies on guesses about the internal work flow of the dispenser. That being said, it works so far and is relatively simple to build, repair or understand. It should also control the non "IR" one provided that you remove the feedback wire.

Thanks Edouard, Jeremy and Timo for their help in doing this. And many thanks to Julien, a friend of mine who was patient enough to answer my numerous naïve questions, never failing to provide an understandable answer!

The pellet dispenser internals

Apparently, Med Associates Inc decided that everything was +28V, go to hell if you're not happy with that[1]... For the small electrical motor, why not, for the control though? Anyway, you have a coupe of wires for the motor power supply, one for the operate and one for the feedback.

Here's how it works : you have to maintain the operate cable at +28V all the time, and when you want to dispense something you have to send a tension fall to 0 which is interpreted as *GOGOGO, DELIVER A PELLET!*. The duration of this fall, or how long it should stay like this is nowhere specified, note however that the feedback wire operates the same way : to warn you of an error (such as "no pellet were dispensed", and this is actually the only possible case), it sends a $+28 \rightarrow 0 \rightarrow +28$ where 0 lasts for 50ms, and keeps the wire at +28 all the

[1] My theory is that the actual reason for this is to make it "harder" to make it yourself so that you have to keep buying stuff from them.

time otherwise. So maybe that what we should do on the operate cable? Guess what, that's what I did! What a hero.

NB : the wires name is incoherent, as per the manufacturer's will: the output wire brings the power in and the input wires gives the feedback. Deal with it.

The box internals

Attached to this text should be an electrical circuit, let's break its logic apart in a human readable way:

- The power is supplied through an AC/DC converter, easy to replace, pretty standard, and is not used in the arduino part.
- Both the operate and the feedback cable are operated through the arduino, in the folowing way:

1. The feedback is the simplest one : most of the time is +28V, and we want to read 0V to know that something went wrong. But the arduino only accepts 5V max on the analog input pins, so we go through a simple voltage divider (good to know : if you want to compute the total equivalent resistor, know that the arduino has a 100MΩ internal resistor and is thus neglected in comparison with the 100KΩ that runs in parallel). The resistances were chosen high as the output current is very small and because Ohm's law, meaning that with too small resistors the voltage was falling dramatically and thus thing became unreadable from an arduino's point of view.

   It is connected to the analog pin **A05**.

2. The operate is slightly more tricky : because we can output max 5V, 40mA, a reed relay was used (ref. W172DIP-1 from Magnecraft/Schneider Electric) that is commanded with 5V, has an internal resistor of 200Ω (thus the 50Ω additional resistor to get a 20mA current as recommanded in the arduino doc), but handles up to 60V 1A for the relay part. This is an SPDT relay as we need at will to set the output value either to +28V or to 0V. A diode was added to force users to put it in a given way (why you may ask? Because I fancy people using the schematics the way I drew them. Arguably you can remove it and it becomes more general.). Another **important** diode was added as a flyback diode to prevent locally infinite current and thus to avoid damaging the relay or the arduino. When the arduino goes $+5V \rightarrow 0 \rightarrow +5V$ in 50ms, the output thus goes $+28V \rightarrow 0 \rightarrow +28V$, which is what we want. Note that it is possible to remove the diode and replace the relay with its brother the W172DIP-5 that has an integrated one.

   It is connected to the digital pin **D13**.

The cable internals

The data cable. Two wires are used for the power, one for the feedback and one for the operate. The ground is on pin 8, the +28V on pin 2, the operate on pin 1 and the feedback on pin 3 (see attached picture). Internally, in the box, the ground is black, the power is red, the feedback is green and the operate is yellow. Inside the extension wire are two pairs of wire, the first one is red and transparent respectively for +28V and ground, and the other is brown and transparent respectively for operate and feedback.

Some programmation notes

The logical part

If you can pilot the arduino from a computer, what you want to do is to set the digital pin D13 to HIGH all the time, and whenever you want to deliver a pellet you can go for a

```
Set_Arduino_Pin(A13,LOW)
Wait(50ms)
Set_Arduino_Pin(A13HIGH)
```

Then you can wait for about a second, check the analog pin as quickly as possible for half a second and if you read a surprisingly low value, typically 0V, then it means that no food was dispensed. In this scenario it is up to you to retry or not, manually or not, etc.

The MATLAB part (aka the illogical part)

I wrote a MATLAB function that does this. Here's its prototype:

```
function didItWork = give_reward(a)
```

It takes an arduino — that you should get separately with `a = arduino();` as it takes time to initiate the connection — returns a boolean that is set to true is food was delivered and false otherwise. Note that the pins are hardcoded but easy to change.

This is a **blocking** function and it takes about 1.5s in total, it's up to you to either write a parallelized version or to deal with it any way you can.

The circuit

As naïvely design by Mathias

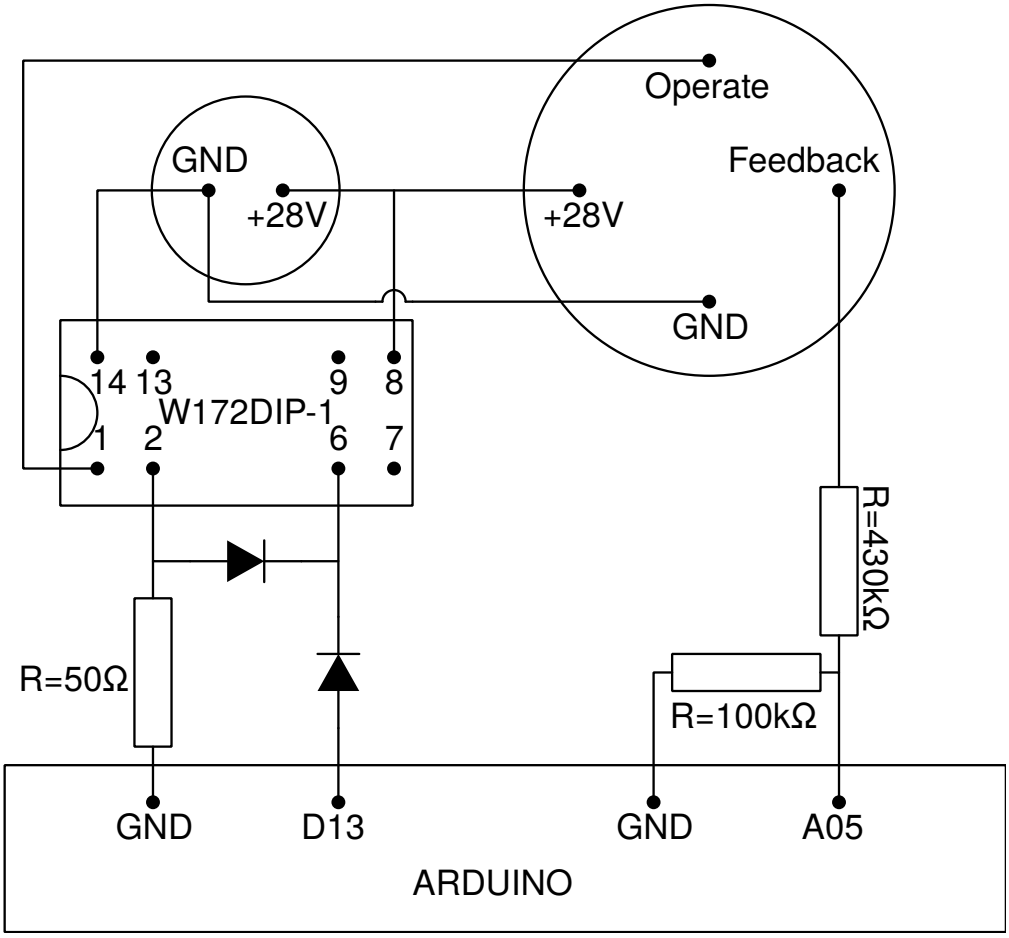

Figure 1: The circuit's conceptual design — the circles are the power and the data plugs

As redesigned by Edouard

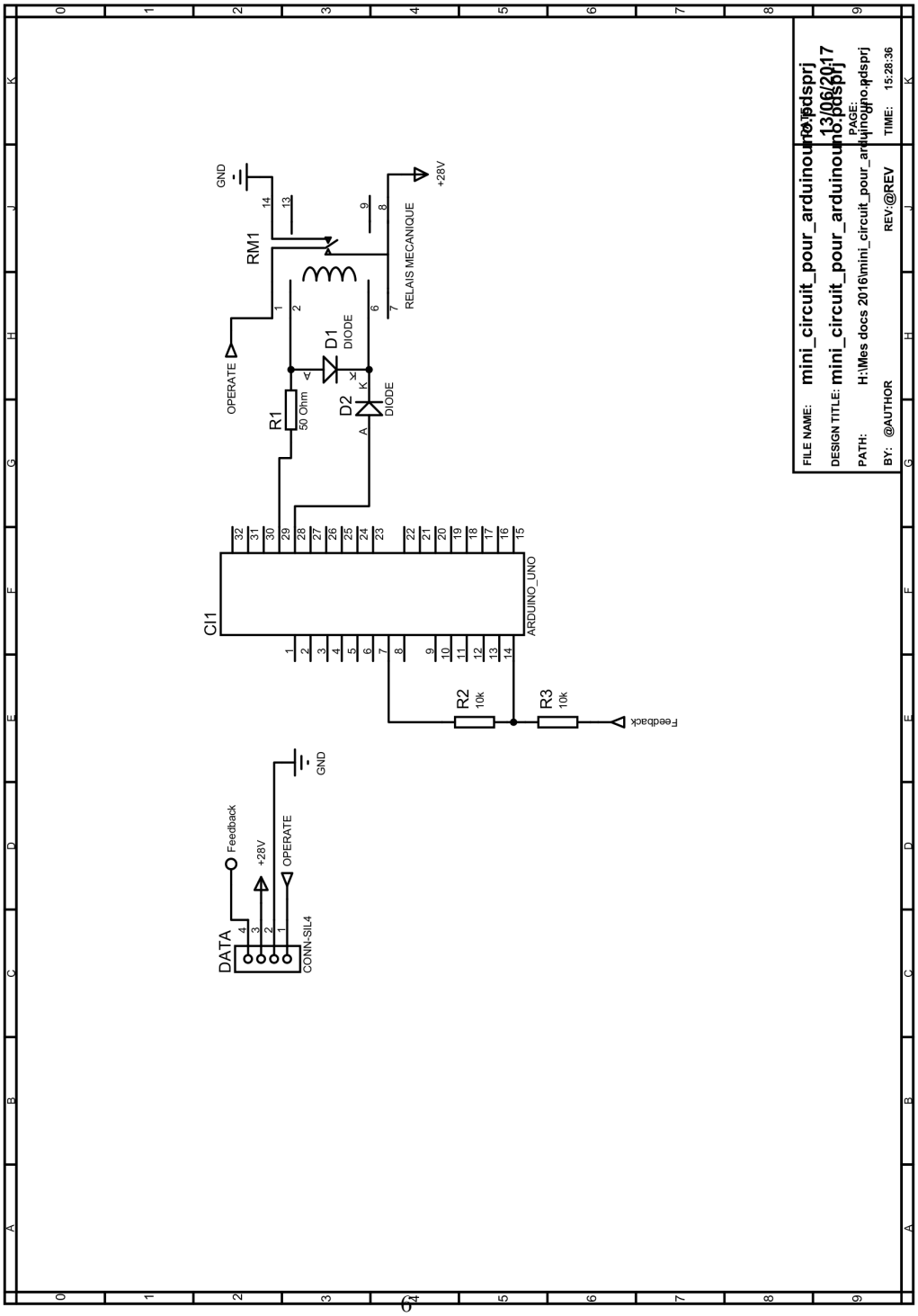

Figure 2: A more professional circuit, designed by Edouard

The PCB design

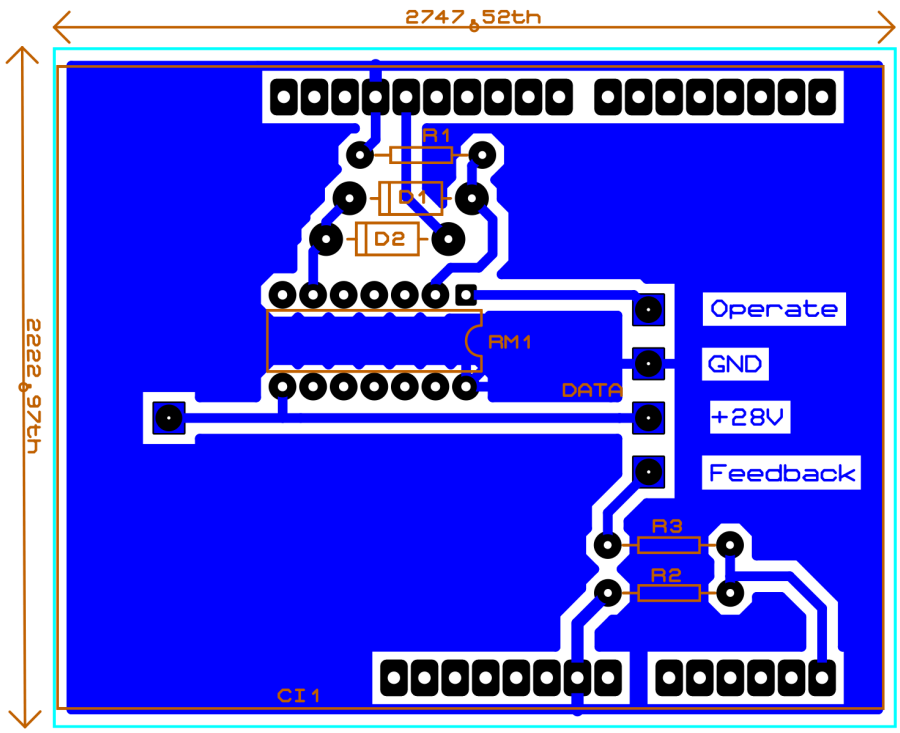

Figure 3: The PCB design from Edouard