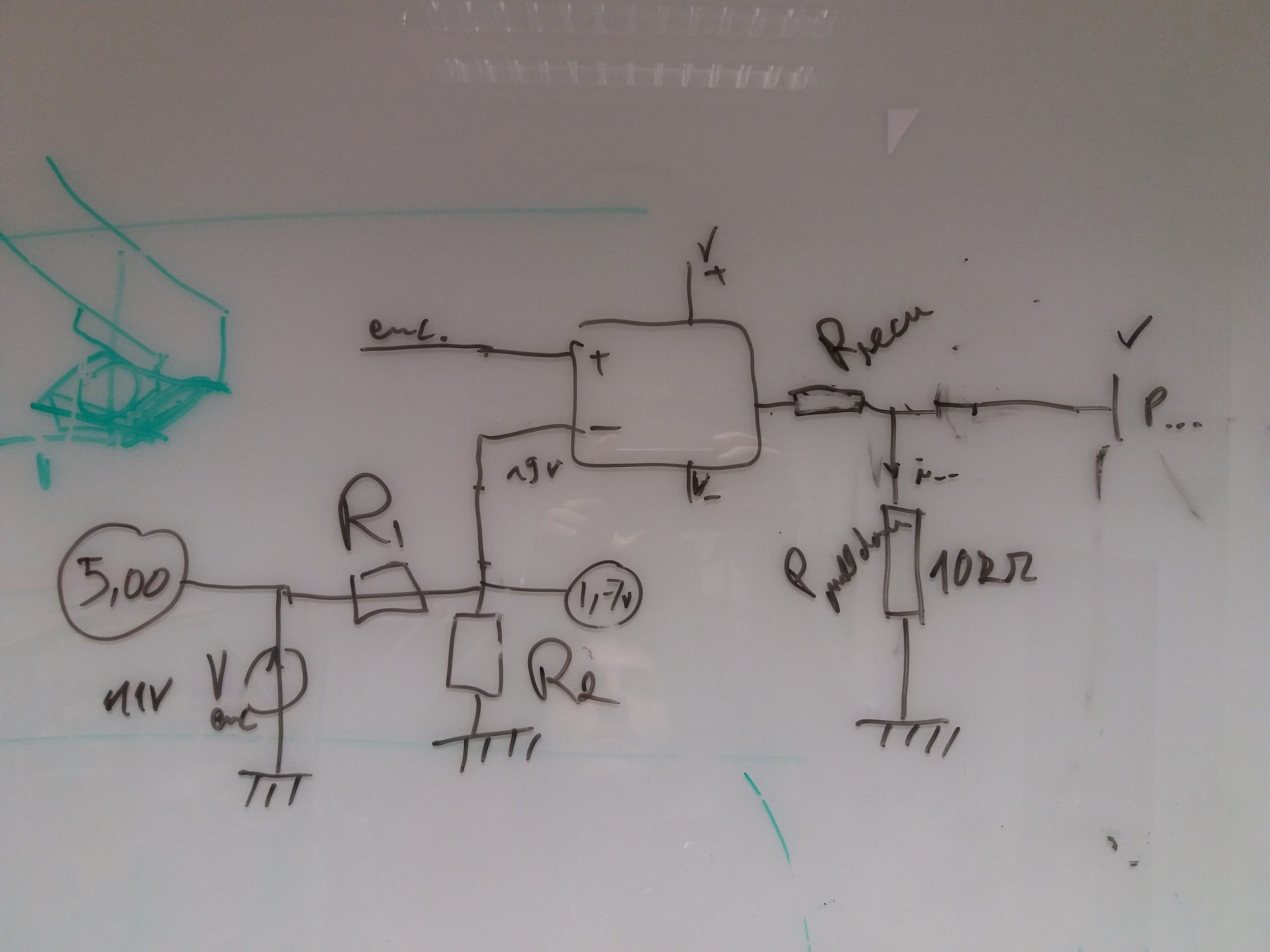
**Découverte encodeur rotatif MUIJE**

Plein de problèmes qu’on a pas réussi à résoudre :

Tension de sortie non compatible Arduino (Pour pouvoir détecter facilement un changement avec Arduino, on voudrait : LOW = 0-1,5V, HIGH = 3-5V. Ce sont les valeurs qui permettent de faire un *attachInterrupt* avec un pin *digital* Arduino). Or, et contrairement à ce qui est affiché sur la Datasheet, l’encodeur (quand alimenté par du 5V) donne LOW = 1,5V et HIGH = 1,8V.

On pourrait essayer de faire des interrupts avec un pin analogique, mais ça a l’air compliqué.

Autre idée : faire un montage avec un ampli op pour transformer 1,5V => 0V et 1,8V => 5V. On a essayé, ça marche pas tout le temps, et quand ça marche ça donne 1,5V => 0V et 1,8V => 3,7V, ce qui n’est pas ouf.



R1 = 630kOhms environ (on l’a trouvée en tâtonnant) et R2 = 330kOhms

Un autre problème se pose, celui de la fréquence des clicks qui risque d’être trop élevée. Il semblerait (testé avec oscilloscope) que le temps de passage de l’encodeur de HIGH à LOW soit trèèès grand (plusieurs dizaines de µs, alors que la Datasheet le signale < 10µs). Or, quand notre robot ira à pleine vitesse (1 m/s, avec des roues de périmètre 12cm, et 5000 clicks par tour), on aura une fréquence de clicks de 42kHz. Donc l’encodeur n’a pas le temps de passer de HIGH à LOW ! En pratique, il remonte plus vite qu’il ne descend, ce qui fait que la tension observée à l’oscillo (et il peut y avoir erreur, mais c’est ce que l’on a cru observer) est à peu près moyennée à la tension HIGH (1,8V).

Enfin, dans l’hypothèse où l’on arrivait à avoir les clicks qui vont bien, détectables par l’Arduino, est-ce qu’elle pourra gérer tous ces *interrupts* au rythme de plusieurs dizaines de milliers par seconde, tout en faisant les autres calculs en parallèle (asservissement) ? La fréquence du processeur est de 16MHz, mais on ne sait pas trop comment mettre en relation cette fréquence et le nombre maximal de clicks que l’on pourra détecter. Et si Arduino ne marche pas, peut-être faudra-t-il passer sous Raspberry Pi, comme l’avait conseillé Exotec et contrairement à notre idée initiale.

En définitive, on ne sait pas trop quoi faire : est-ce qu’il faut continuer à se battre avec cet encodeur pour le faire fonctionner, ou est-ce qu’il ne serait pas plus simple de se rabattre sur un encodeur plus classique, peut-être moins performant, mais qui donnera LOW = 0, HIGH = 5V, qui passera vite de l’une à l’autre, et qui ne tuera pas Arduino à cause d’un trop grand nombre d’interrupts.