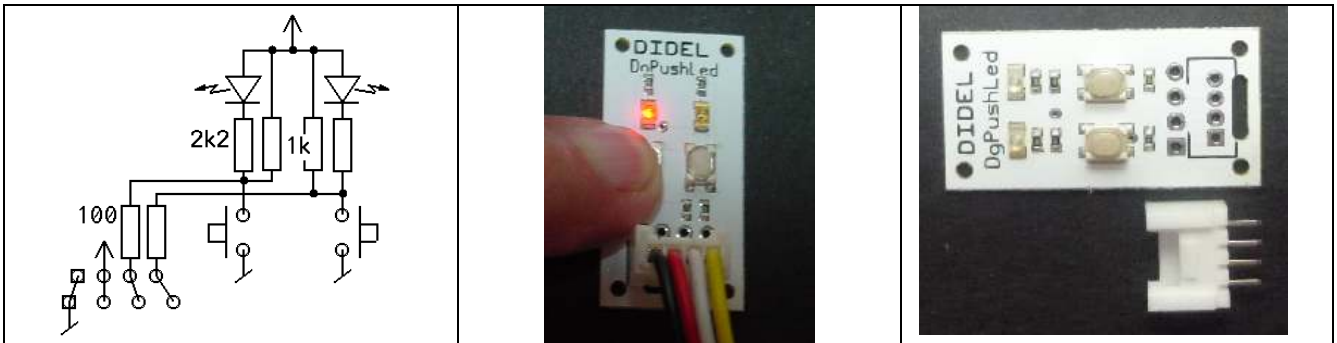


DgPushLed

Two pushbutton and/or 2 leds



Le même module est utilisé comme poussoirs ou comme leds. Dans les deux cas les signaux sont actifs à zéro.

Si le microcontrôleur est programmé en **entrée**, il lit:

touche pressée 0V état 0 (LOW)

touche relâchée >3V état 1 (HIGH).

Programmer une pull-up en entrée du microcontrôleur n'est pas nécessaire, mais elle ne gêne pas.

Si le microcontrôleur est programmé en **sortie**, il écrit:

état 0 (LOW) la Led est allumée. Le processeur ou l'interface doit avoir une résistance interne vers le + (pull-up) supérieure à 470 Ohm pour que la tension soit inférieure à 1.5 - 2V. . Le courant absorbé est de $(5-1.5)/3.2 \approx 1$ mA.

Dans cet état 0, si on presse sur le poussoir, la Led est un peu plus lumineuse et l'état zéro est renforcé..

état 1 (HIGH) la Led est éteinte. Si on presse elle s'allume, un courant de 5mA est fourni par le microcontrôleurs à travers la résistance de protection de 100 Ohm, ce qui est sans conséquence.

Programmer un poussoir est beaucoup plus riche que l'on croit. Il a une action statique et dynamique, des rebonds, et si une led est associée, on peut donner une quittance à chaque action sur le poussoir. Voir www.didel.com/digrove/PousLed.pdf

Cela devient un peu plus compliqué pour choisir un programme selon le nombre d'impulsions sur un poussoir: Voir www.didel.com/diduino/Poussoir.pdf

Le LearnCbot avec ses poussoirs et leds a été développé pour appuyer le MOOC EPFL "Microcontrôleurs". De nombreux exemples et exercices sont donnés dans www.didel.com/coursera/LC.pdf



The same module can be used as an input for 2 switches, or as 2 leds. You can even make both at the same time, the software can read the switch while the led is blinking. Depressing the switch will always light the led, but protection resistors avoid excessive current in case of collision.

Understand the schematic. You may be surprised to see a resistor in parallel to the the Led. This is not required at 5V, but is safer at 3V: the red led induces a voltage drop of 1.5V, hence the high level is 1.5V and not 3V.

If the microcontroller is programmed in **input** mode:

button depressed, one read a "0" (LOW)

button released, one read a "1" (HIGH)

No need to program a pull-up, but it is permitted.

If the microcontroller is programmed in **output** mode

if a "0" (LOW) is written, the led is ON. Current is ~1mA

if a "1" (HIGH) is written, the led is OFF, and no current is asked. But if the button is depressed, the led lights, and a "short circuit" current of 5mA must be given by the pin, not a problem.

Programming a button offer many possibilities. The static operation is evident, but detecting dynamic actions (transition off-on, durations, counting bursts, successive actions like morse code) are excellent (and useful) exercises to understand C programming.

See www.didel.com/diduino/ReadButton.pdf (to be done, ask for it if you do not read French!)

