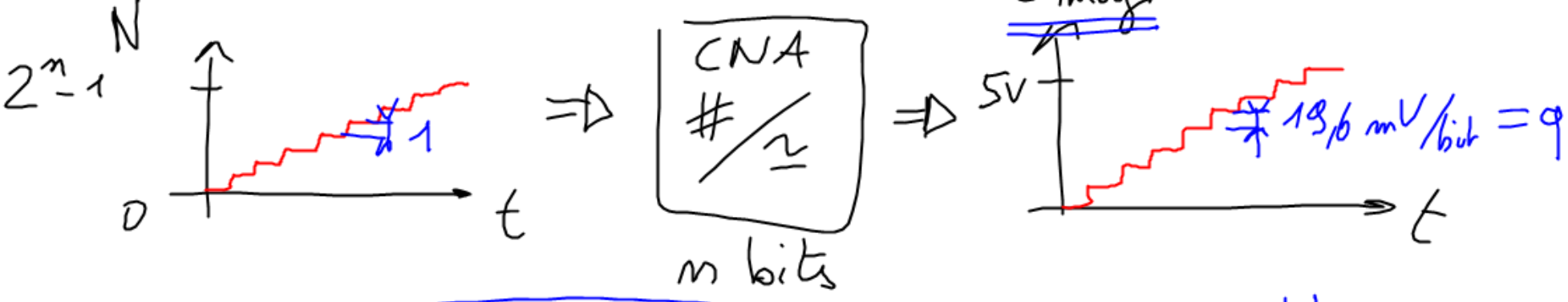
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

$2^3 = 8$

$7 = 2^3 - 1$

1 0 0 0 8

II CNA



Quantum : $q = \frac{U_{pleine\ echelle}}{2^n - 1}$

$U_{moy} = N \cdot q$

ex : n = 8 bits $U_{pleine\ echelle} = 5\,V$

pour N = 100 calculer U_{moy}

$q = \frac{5}{2^8 - 1} = \frac{5}{255} \simeq 1{,}96 \cdot 10^{-2}\ V/bit$

$\simeq 19{,}6\ mV/bit$

$U_{moy} = 1{,}96 \cdot 10^{-2} \cdot 100 = 1{,}96\ V$

sketch_mar13a | Arduino 1.8.12

Fichier Édition Croquis Outils Aide

sketch_mar13a §

3/7

```
int pinCNA = 3;
int pinCAN = A0;
int valCAN = 0; //10bits
int valCNA = 0; //8bits

void setup() {
pinMode(pinCNA, OUTPUT);
}

void loop() {
valCAN = analogRead(pinCAN); // de 0 à 1023
valCNA = valCAN / 4; // de 0 à 255
analogWrite(pinCNA, valCNA);
}
```

Téléversement terminé

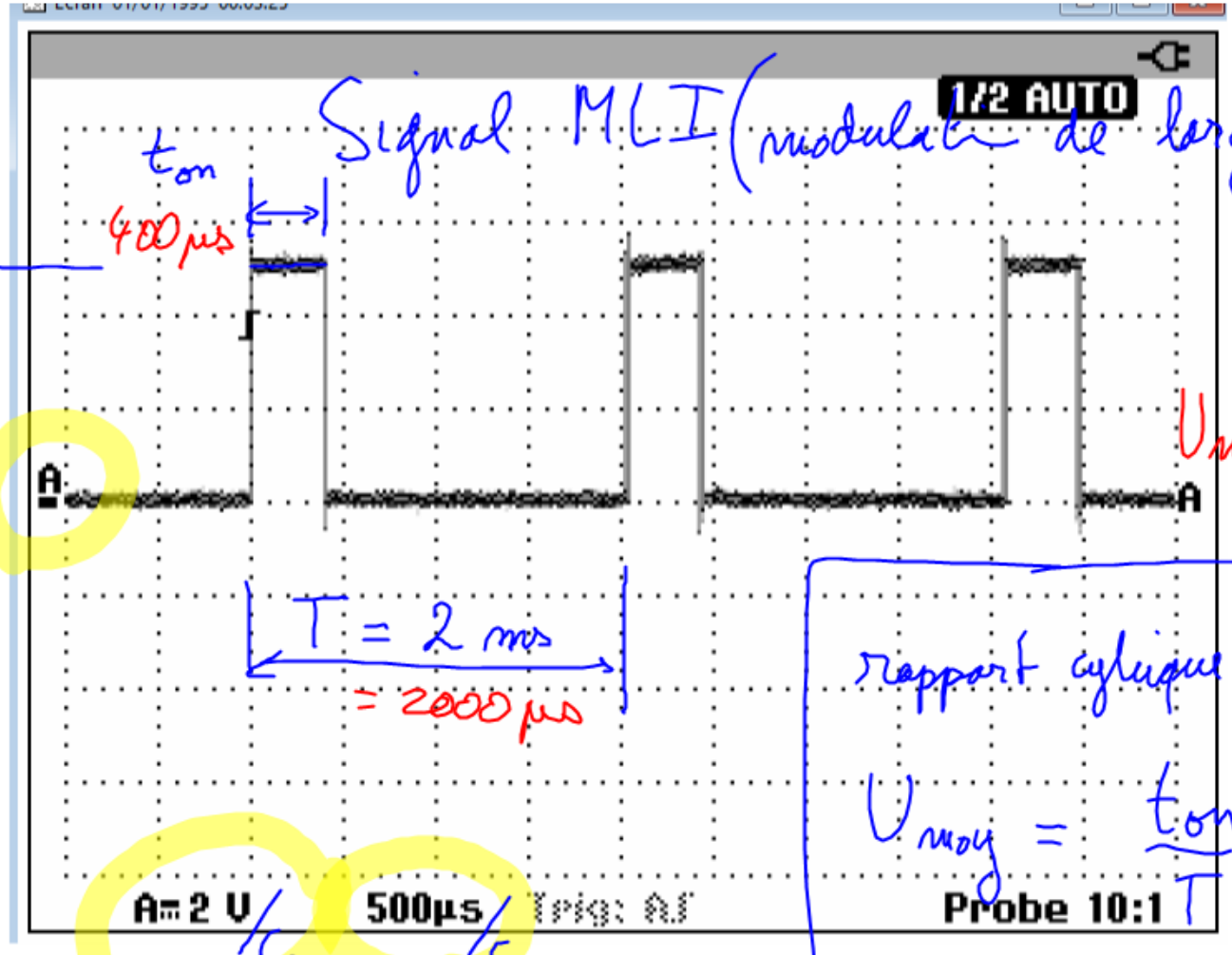
```
Le croquis utilise 962 octets (2%) de l'espace de stockage de programmes. Le maximum est de
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2039 oct
```

3

Arduino Uno sur COM19

FRA 14:13 13/03/2025

Signal MLI (modulation de largeur d'impulsion) (=PMW)



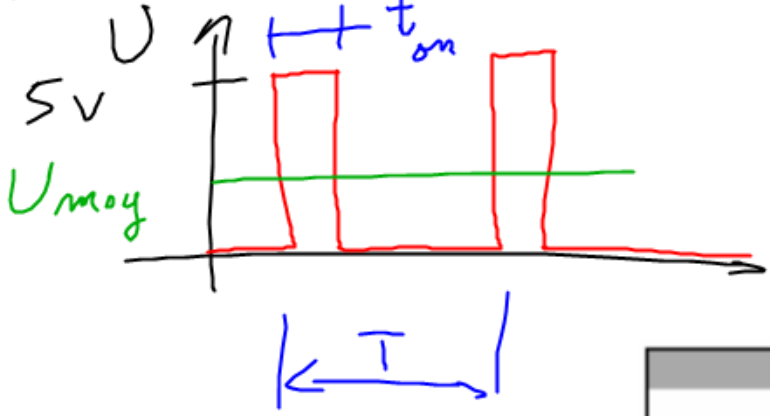
rapport cyl = $\frac{400}{2000} = 20\%$

$U_{moy} = \frac{1}{5} \cdot 5 = 1\ V$

rapport cyclique = $\frac{t_{on}}{T}$

$U_{moy} = \frac{t_{on}}{T} \cdot U_{pleine\ échelle}$

Signal sortie CNA :



rapport cyclique = $\frac{t_{on}}{T}$ (en %)

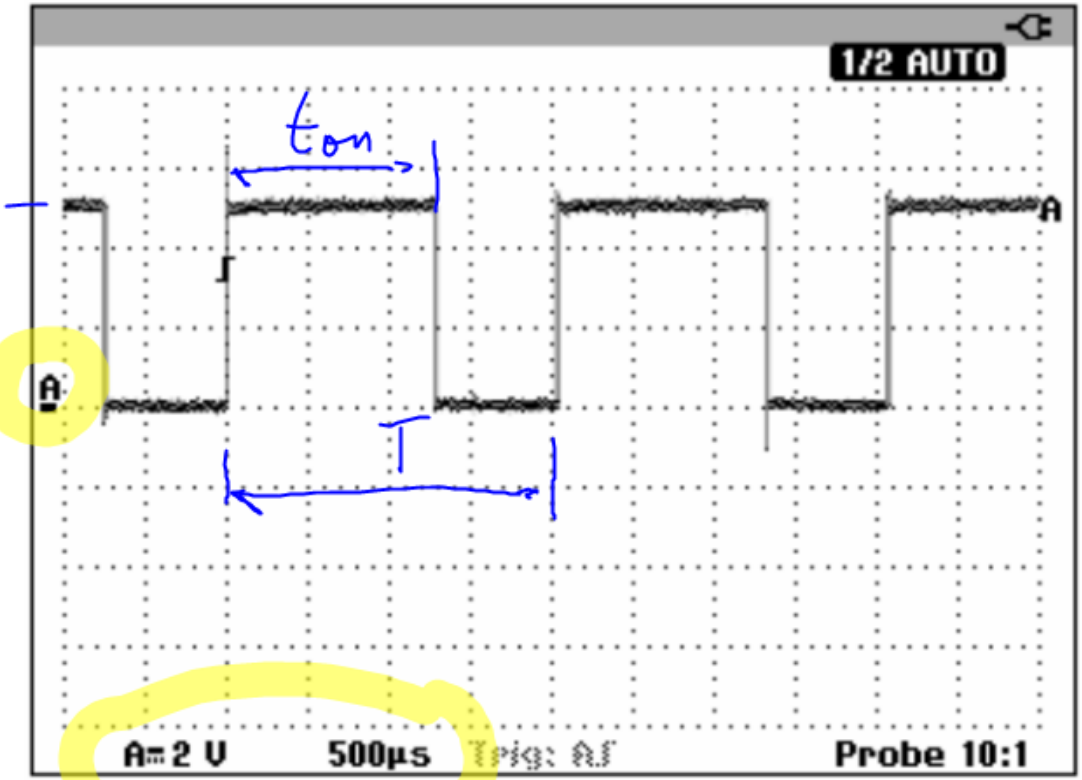
$U_{moy} = \frac{t_{on}}{T} \cdot U_{pleine\ échelle}$

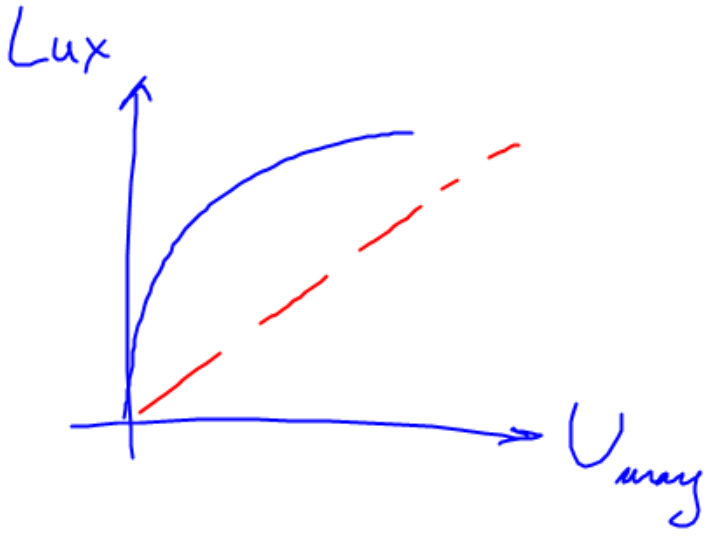
ex : $T = 4 \times 500 = 2000\ \mu s$

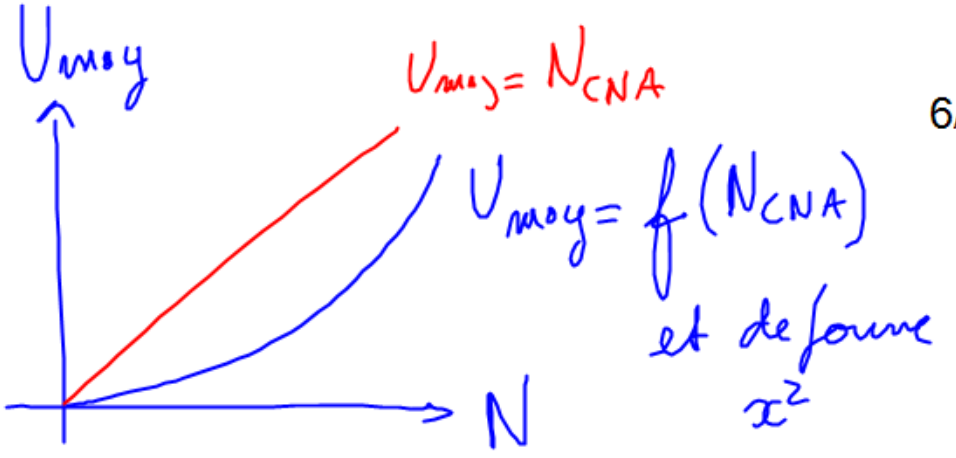
$t_{on} = 1300\ \mu s$

$U_{moy} = \frac{1300}{2000} \cdot 5 \simeq 3{,}25\ V$

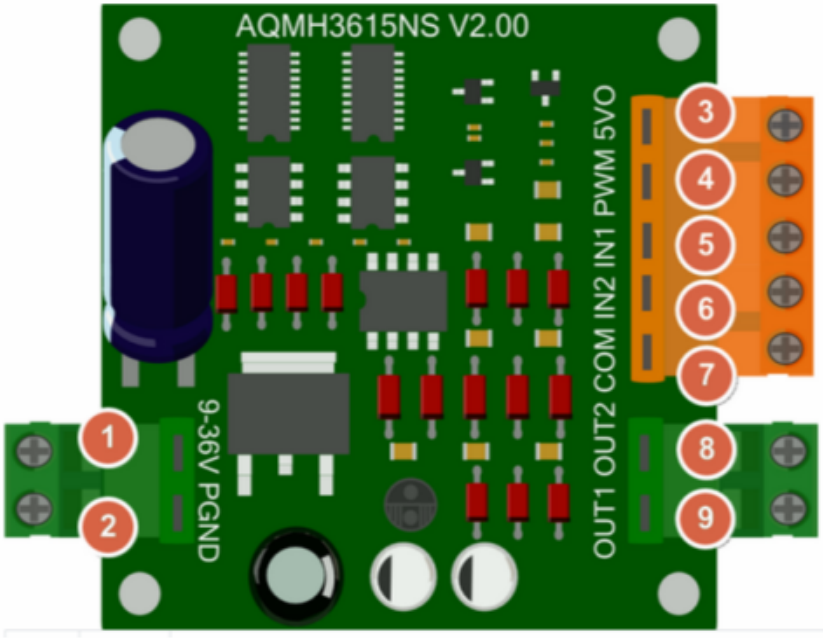
avec $\frac{t_{on}}{T} = \frac{1300}{2000} \simeq 65\%$







$$N_{CNA\_L} = \frac{N_{CNA}^{2}}{255}$$



IN1	IN2	PWM	OUT1, OUT2 Motor Behavior
0	0	x	Stop
1	1	x	Vacant
1	0	1	Forward 100%
0	1	1	Reverse 100%
1	0	PWM	Forward at PWM speed
0	1	PWM	Reverse at PWM speed

In this table

- "0": TTL_Low
- "1": TTL_High
- "PWM": PWM speed setting
- "x": Any TTL, and it is default TTL_Low while no PWM signal.

Note1: IN1 & IN2 To protect your motor, before switching the motor steering direction, make sure firstly to BRAKE motor by setting IN when the PWM was set as 100%, full speed. And the suggestion time to brake is >0.1S, depending on your motor.

Num	Label	Description
1	9 - 36V	Power Supply,
2	PGND	Power Supply, GND/ -
3	5VO	5V Output, you can use this power to feed Arduino through Arduino's 5V port, NOT Vin which requires 7-12V input
4	PWM	Speed control signal input
5	IN1	Motor steering control signal input 1, **Note1**
6	IN2	Motor steering control signal input 2
7	COM	GND
8	OUT2	Motor_
9	OUT1	Motor_-