1 octet = 8 bits = nombre Hex à 2 chiffres

ASCII décimal

$$E3_{(16)} = 14 \times 16^1 + 3 \times 16^0$$

Rang 2 1 0

$7AE_{(16)} = 7 \times 16^2 + 10 \times 16^1 + 14 \times 16^0$

Rang 7 6 5 4 3 2 1 0

$1011\ 1011_{(2)} = 1.2^7 + 1.2^5 + 1.2^4 + 1.2^3 + 1.2^1 + 1.2^0$

128 64 32 16 8 4 2 1

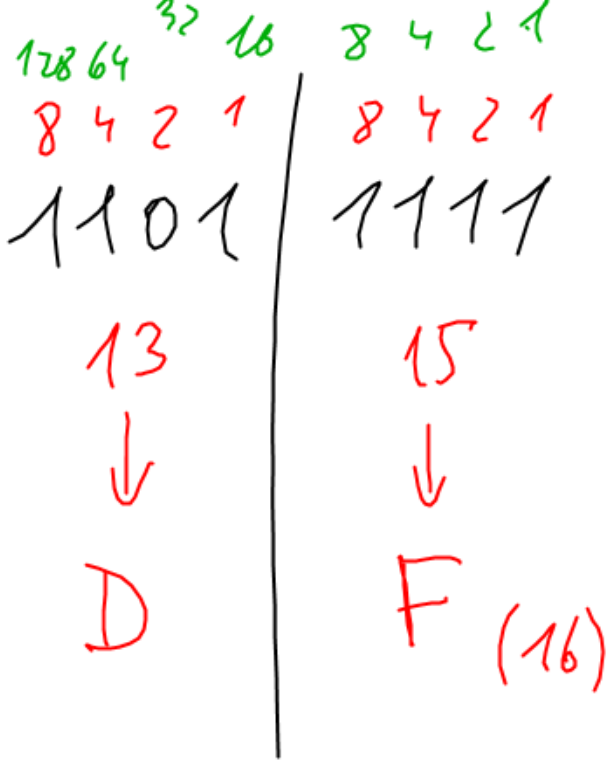
$= 128 + 32 + 16 + 8 + 2 + 1$

$1432_{(10)}$

1432	16	
8	89	16
9	5	16
5	0	

$= 598_{(16)}$

$(= 5 \cdot 16^2 + 9 \cdot 16 + 8)$



128 + 64 + 16 + 8 + 4 + 2 + 1

= (10)

(16)	(10)
0	0
⋮	⋮
9	9
A	10
⋮	
F	15
10	16

suivantes :

Code d'accès 72 bits	En-tête 56 bits	Données 240 bits max

Code d'accès	72 bits 9 octets	Ces bits permettent de configurer la liaison afin de synchroniser les différents composants Bluetooth. (Fréquence, code canal, code de recherche……
En tête	56 bits 7 octets	Ce champ contient dans l'ordre l'adresse de l'esclave (codée sur 3 bits), le type de paquets et des bits de contrôle (erreurs…)
Données	240 bits max	La taille de la donnée est variable et peut aller jusqu'à 240 bits maximum.

Dans une trame **c'est le LSB (bit de poids faible) qui est transmis en premier**.

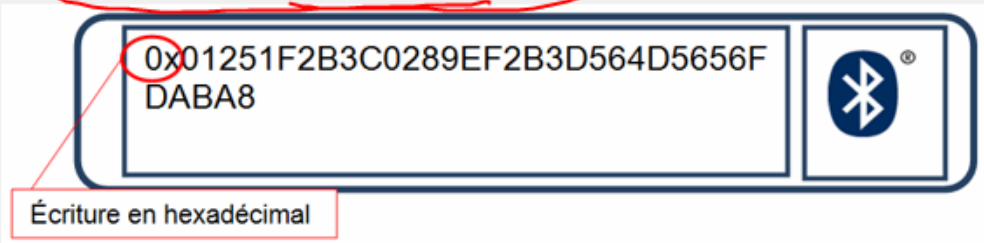


Figure 23 : relevé de la trame Bluetooth de commande d'inclinaison de la tablette.

L'information concernant la commande du sens de déplacement correspond au bit (bit_{sens}) de poids faible du premier octet de données (si bit_{sens} = 0 diminution de l'angle (déplacement négatif), si bit_{sens} = 1 augmentation de l'angle (déplacement positif)).

		3 bits), le type de paquets et des bits de contrôle (erreurs…)
Données	240 bits max	La taille de la donnée est variable et peut aller jusqu'à 240 bits maximum.

Dans une trame **c'est le LSB (bit de poids faible) qui est transmis en premier**.

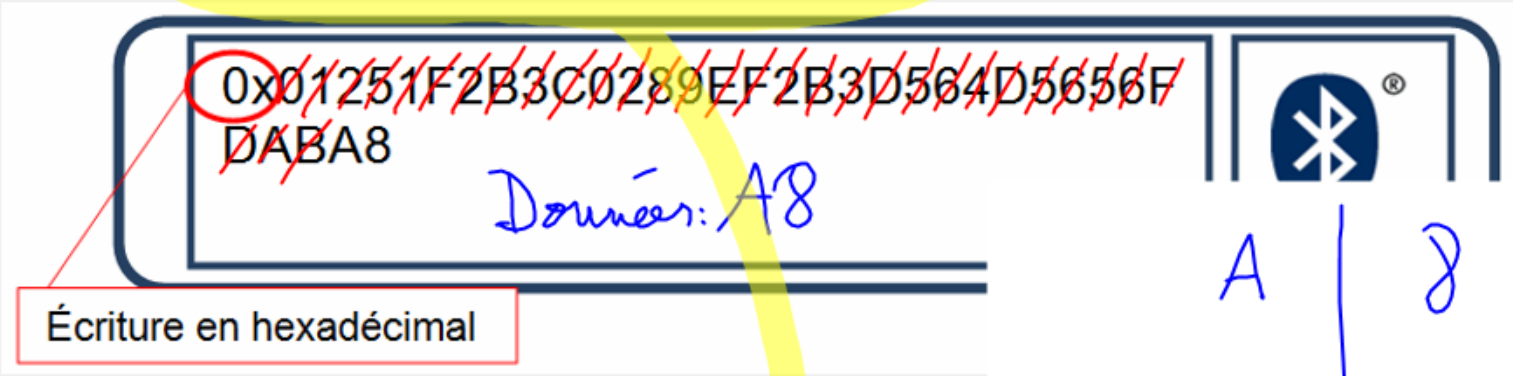


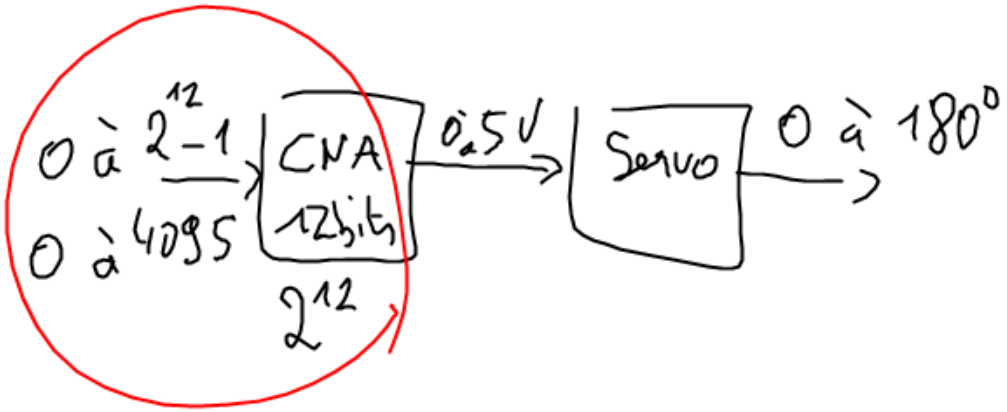
Figure 23 : relevé de la trame Bluetooth de comma

formation concernant la commande du sens de déplace
le du premier octet de données (si bit_{sens} = 0 diminut
$_{ns}$ = 1 augmentation de l'angle (déplacement positif)).

A | 8 (16)
10 | 8 (10)
8421 | 8421 (2)
1010 | 1000
Données : 00101010
↑ bit sens

0°	0000	0000	0000
	0	0	0
Quantum	0000	0000	0001
180°	1111	1111	1111
	F	F	F

$$q = \frac{\text{Données pour 1 échelon}}{2^n - 1}$$



$$q = \frac{180}{4095} \simeq 0{,}044\ ^\circ/\text{bit}$$

Servomoteur	Référence : FEETECH FT835BL
Masse	72 g
Vitesse de fonctionnement	0,14 sec / 60 degrés (6 V)
Couple de calage	30 kg·cm/416,61 oz·in (6 V)
Tension de fonctionnement	6 V
Angle de fonctionnement	180 degrés
Impulsion requise	500 à 2500 µs
Longueur du fil du connecteur	30 cm

→ 15° en $\frac{0,14}{4}$ s. = 0,035 s

(Attention : les unités peuvent être différentes du système international.)

La largeur de l'impulsion de commande est codée sur 12 bits. La commande d'un angle nul correspond à 12 bits à 0 soit 000 en hexadécimal, la commande d'un angle de 180° correspond à 12 bits à 1 soit FFF en hexadécimal.

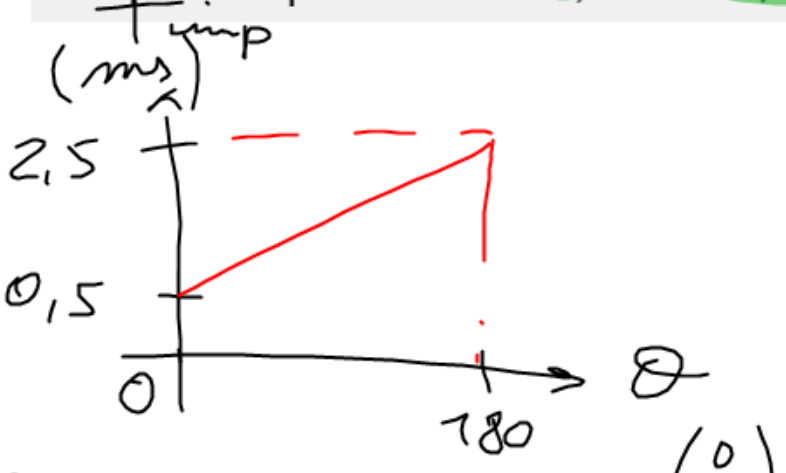
quantum

Question 1.16 ***Déterminer*** *l'angle de déplacement de la tablette le plus petit qu'il est possible de commander, en degrés, puis le temps mis par la tablette pour pivoter de 15°.* ***Commenter*** *cette valeur dans le contexte d'utilisation de la tablette.*

Constitution d'une trame Bluetooth

Les données sont transmises par paquet. Chaque paquet est constitué des informations

Un servomoteur est un système électromécanique, asservi en position, servant à actionner les parties mobiles d'un modèle-réduit, et répondant à une commande externe de type MLI (modulation à largeur d'impulsion) généralement transmise par une radiocommande. C'est la largeur de ces impulsions, générées périodiquement, qui détermine la position angulaire de l'axe de sortie. L'amplitude angulaire du servomoteur varie de 0° à 180° (pour une largeur d'impulsion qui varie entre 0,5 ms et 2,5 ms) pour une période de 20 ms.



$$T_{imp} = \frac{2}{180}\,\Theta + 0{,}5$$

$$T_{imp} = 1{,}61\ ms$$

△

2 → 180

x → 100

x = 1,11

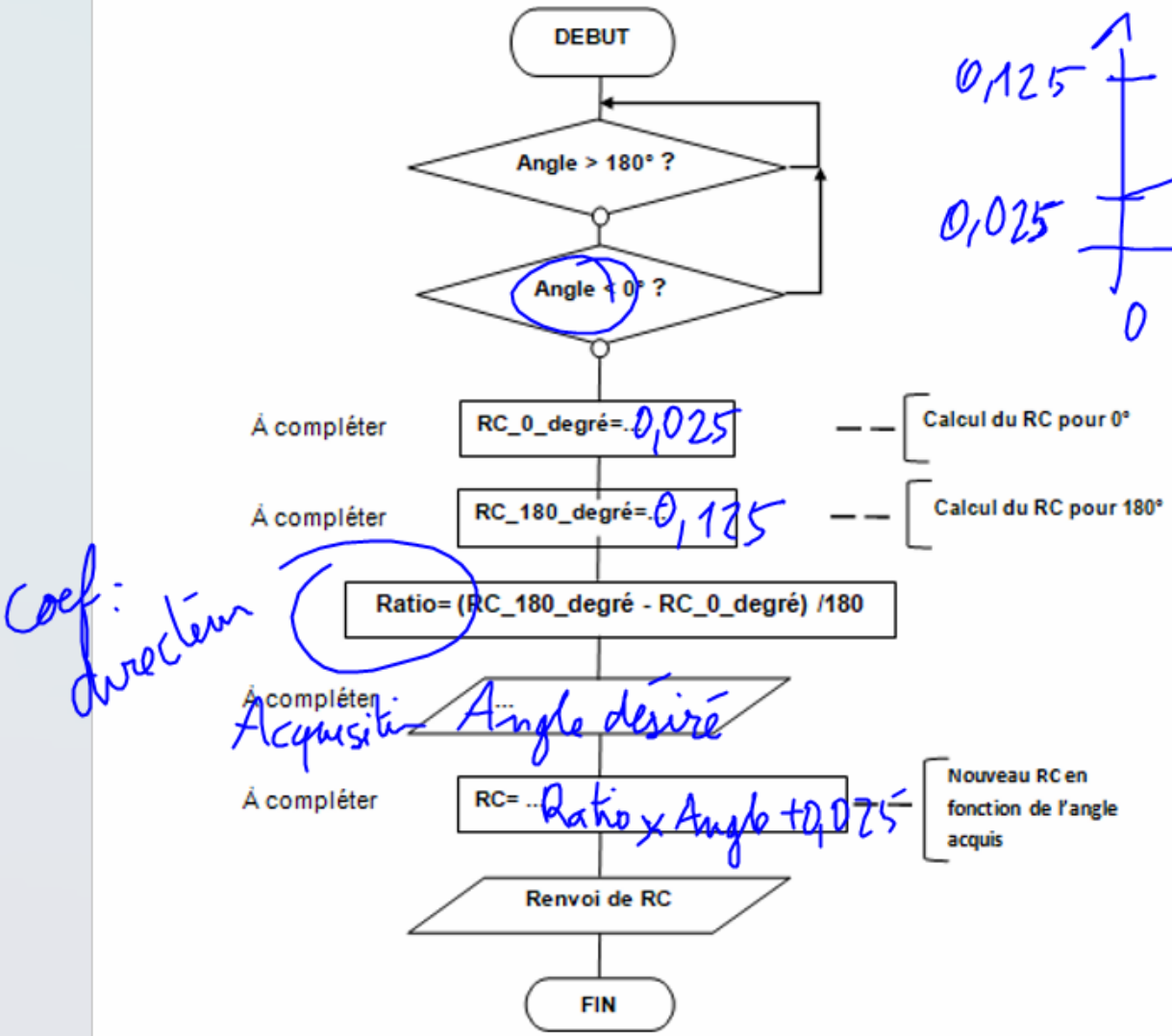
$T_{imp\ 100°} = 1{,}61\ ms$

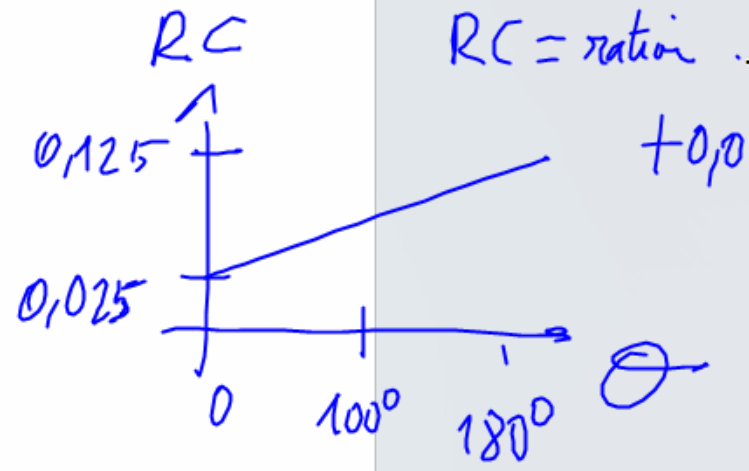
2,5 → 180

x → 100

x = 1,39 ms

Question 1.18





coef directeur

$$RC = \frac{Timp}{20}$$

```python
"""sous programme: Commande du servomoteur d'orientation de la tablette"""

import RPi.GPIO as GPIO                          #importation de la librairie
import time                                      #importation de la librairie time


angle=90                                         #affectation de l'angle milieu de positionnement
                                                 #de la tablette (90°)

def rapport_cyclique_en_pourcentage(angle) :     #fonction qui permet de calculer le rapport cyclique
                                                 #en % en fonction de l'angle
    if angle > 180 or angle < 0 :
        return False

    RC_0_degre=2.5                               #rapport cyclique: impulsion pour 0° / temps
                                                 #pour un cycle= 0.5/20= 0.025=2.5%
    RC_180_degre = 12.5                          #rapport cyclique: impulsion pour 180° / temps
                                                 #pour un cycle= 2.5/20= 0.125=12.5%
    ratio = (RC_180_degre - RC_0_degre)/180      #Calcul du ratio pour l'angle en pourcentage

    RC_en_pourcentage = angle * ratio

    return RC_0_degre + RC_en_pourcentage        #la fonction renvoie le nouveau rapport cyclique en %

GPIO.setmode(GPIO.BOARD)                         #utilisation du mode de numérotation de la carte
GPIO.setwarnings(False)                          #désactivé les avertissements

#utilisation de la broche 12 pour commander en PWM le servomoteur
pwm_gpio = 12                                    #numéro du port de commande du servomoteur
frequence = 50                                   #fréquence 50Hz
GPIO.setup(pwm_gpio, GPIO.OUT)                   #affectation de la broche 12 en sortie
pwm = GPIO.PWM(pwm_gpio, frequence)              #création d'une instance pwm

#utilisation des broches 15 et 16 pour récupérer l'appui sur les boutons de l'application
GPIO.setup(15, GPIO.IN)                          #broche 15 affecter en entrée
```

```
pwm_gpio = 12                                   #numéro du port de commande du servomoteur
frequence = 50                                  #fréquence 50Hz
GPIO.setup(pwm_gpio, GPIO.OUT)                  #affectation de la broche 12 en sortie
pwm = GPIO.PWM(pwm_gpio, frequence)             #création d'une instance pwm

#utilisation des broches 15 et 16 pour récupérer l'appui sur les boutons de l'application
GPIO.setup(15, GPIO.IN)                         #broche 15 affecter en entrée
GPIO.setup(16, GPIO.IN)                         #broche 16 affecter en entrée

pwm.start(rapport_cyclique_en_pourcentage(angle))   #positionnement de la tablette à 90°
while True:
    monter_tablette = GPIO.input(15)        # renommage de l'entrée 15:ici recevra l'information d'augmenter l'angle
    descendre_tablette = GPIO.input(16)     # renommage de l'entrée 16:ici recevra l'information de dimuninuer l'angle
    if (monter_tablette==True):
        pwm.ChangeDutyCycle(rapport_cyclique_en_pourcentage(angle))     #pwm.ChangeDutyCycle permet de changer le rapport
                                                                        #du cycle qui est le résulat de la fonction
                                                                        #def rapport_cyclique_en_pourcentage(angle):
                                                                        #RQ: Le temps d'execution de la commande pour 1° est
                                                                        #de 20 ms (temps de la période) donc 3.6s pour 180°.
        angle= angle + 1                                                #incrémentation de l'angle
        time.sleep (0.02)                                               #temps de pause en seconde par degré: 0.020s
                                                                        #(donc 3.6s de pause pour 180°).
```

Pwm BP (

elif (..descendre_tablette == True) #cas où l'on souhaite diminuer l'angle de la tablette

... pwm.Change - - - - - - - - -

... angle = angle - 1

... time.sleep (0,02)

```
    else:
        pwm.ChangeDutyCycle(rapport_cyclique_en_pourcentage(angle))     #garde le rapport cyclique pour la position actuelle
                                                                        #de la tablette
        time.sleep (0.02)
```