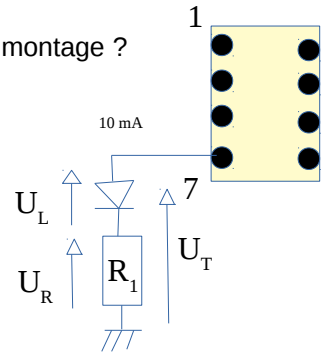
Communication entre systèmes

TD n°1

Liaison parallèle

1 Schéma de câblage

On souhaite piloter une LED sur la broche 7 du connecteur GPIO d'un Raspberry Pi.



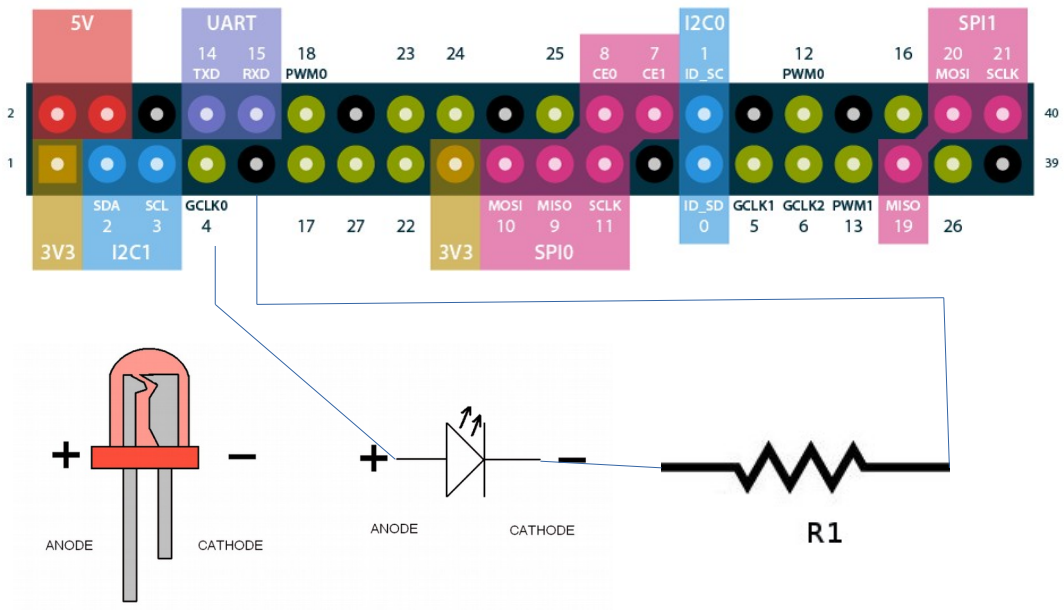
1. Indiquez pourquoi est-il nécessaire de mettre une résistance dans le montage ?
   Sinon la LED explose a cause de la surtension.

2. Complétez le schéma de montage.

3. Calculez la résistance R1

   $R = U_R / I$ $\qquad$ $U_R = U_T - U_L$

   $R = 1,5 / 0,01$

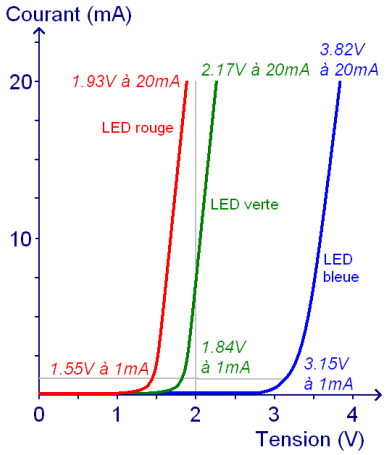


Caractéristiques des LED

LED rouge, orange, jaune : 1.8V à 2V

LED verte standard : 1.8V à 2.2V

Schéma équivalent



2 Écriture du programme avec WiringPi

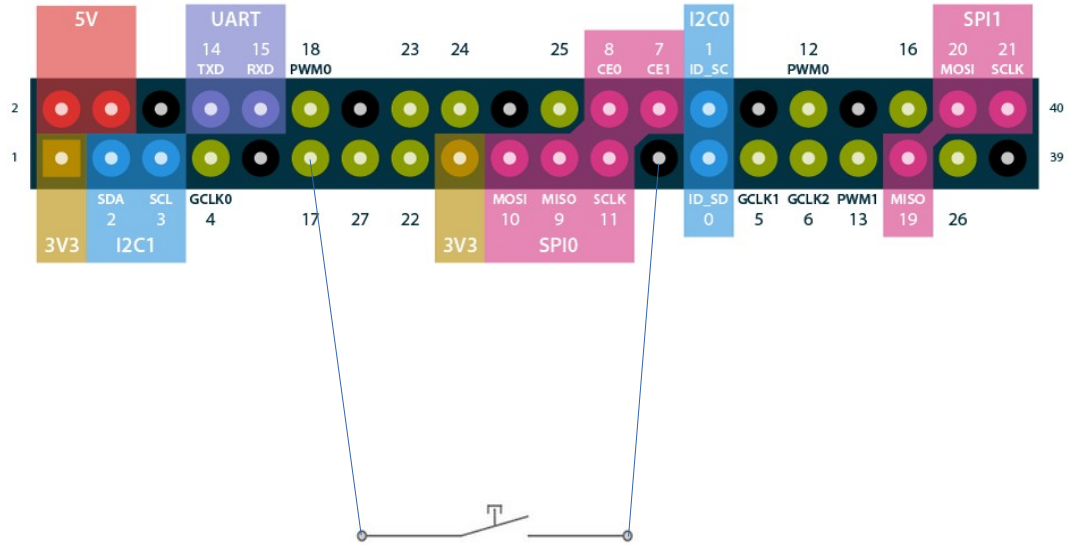
On souhaite faire clignoter la LED avec une fréquence de 2 Hz, réalisez le programme avec la librairie WiringPi.

```
#include <wiringPi.h>
#include <unistd.h>

#define LED 7

int main(void)
{       winingPiSetup() ;
        PinMode(LED, OUTPUT) ;
                do {
                digitalWhite (LED, HIGHT) ;
                usleep (250000) ;
                digitalWhite (LED, LOW) ;
                usleep (250000) ;
                } white (1) ;
}
```

3 Prise en compte d'un bouton poussoir

On souhaite maintenant utiliser un bouton poussoir sur la broche 11 du connecteur GPIO du Raspberry PI

1. Proposez un schéma de câblage (rappel une entrée en l'air est considérée au niveau haut en TTL). Indiquez si vous utiliser une résistance de rappel interne au Raspberry PI



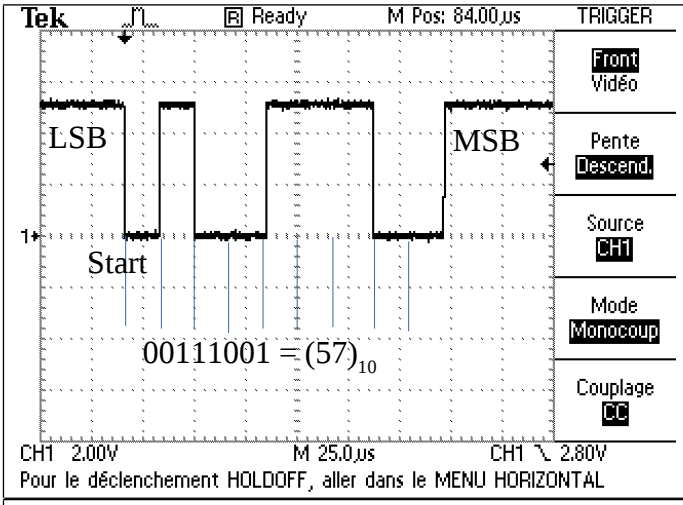
2. Proposez un programme qui arrête le clignotement de la LED lorsque l'on appuie sur le bouton poussoir (en rouge)
3.

```
#include <wiringPi.h>
#include <unistd.h>

#define LED 7
#define POUSSOIR 0

int main(void)
{
	int bouton;
	wiringPiSetup( );
	pinMode(LED,OUTPUT);
	pinMode(POUSSOIR,INPUT);
	pullUpDnControl(POUSSOIR,PUD_UP);
	do{
		digitalWrite(LED, HIGH);
		usleep(250000);
		digitalWrite(LED,LOW);
		usleep(250000);
		bouton = digitalRead(POUSSOIR);
	}while(bouton !=LOW);
}
```

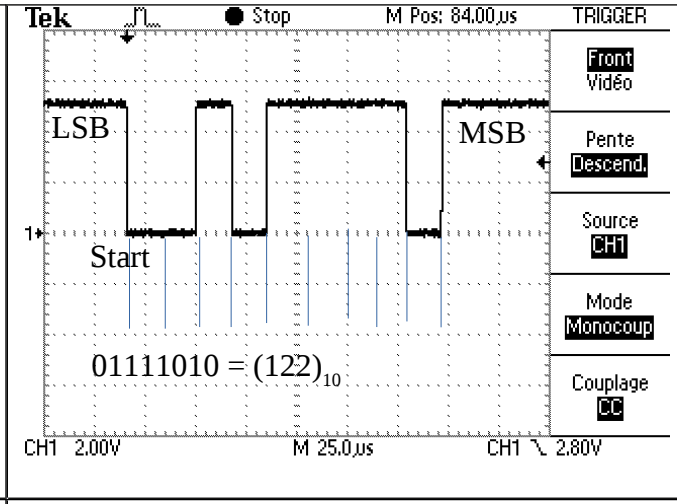
Liaison série

4 Décodage de la trame de la liaison série asynchrone

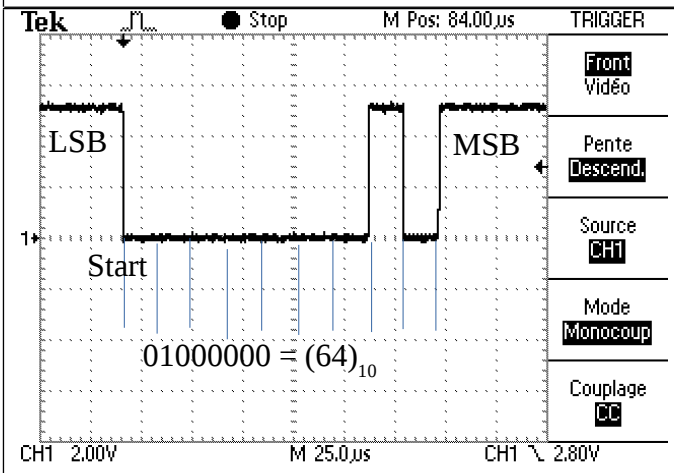
Vitesse de transmission : Nombre de bits de la trame : Parité :



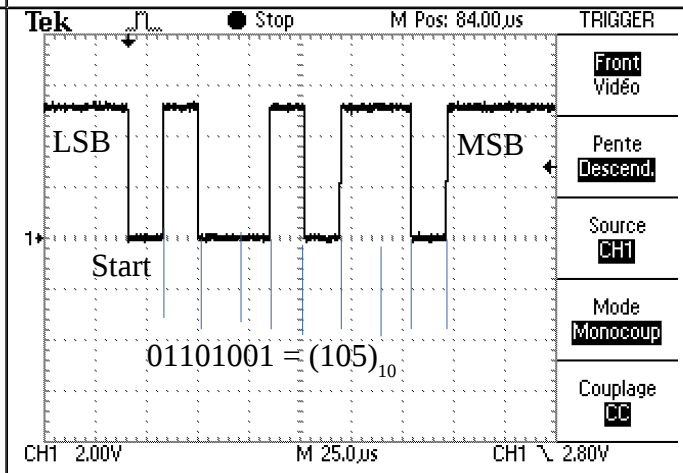
Code ASCII : 9



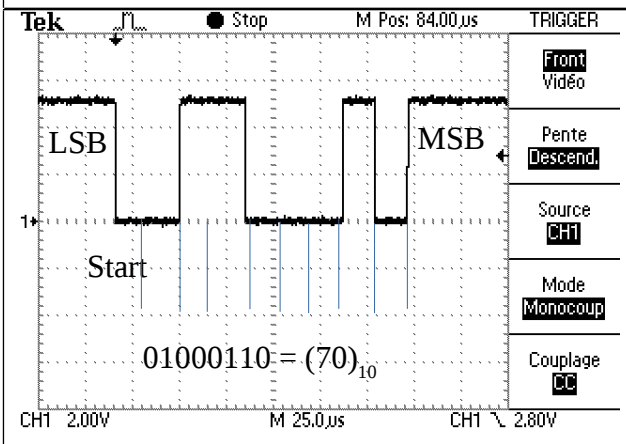
Code ASCII : z



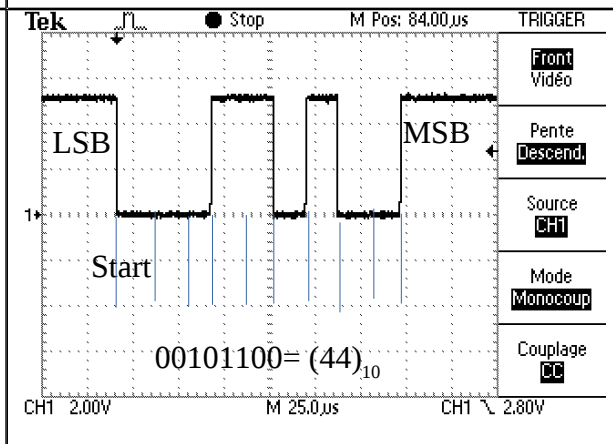
Code ASCII : @



Code ASCII : i



Code ASCII : F



Code ASCII : ,

5 Transmission de données GPS

Le standard NMEA-0183 (pour National Marine & Electronics Association) définit le protocole de transmission de données entre des équipements électroniques destinés à la réception d'informations de positionnement et un microsystème.

Toutes les données sont transmises sous la forme de caractères ASCII imprimables, ainsi que les caractères [**CR**] Retour Chariot et [**LF**] Retour à la ligne, à la vitesse de transmission de 4800 bauds. Les caractères sont codés sur 8 bits avec 1 bit de parité (**parité paire**) 1 bit de **start** et 1 bit de **stop**,

Chaque trame commence par le caractère **$**

Suivi par un groupe de 2 lettres pour l'identifiant du récepteur. (Non limitatif):

GP pour Global Positioning System.

LC Loran-C receiver.

OM Omega Navigation receiver.

II Integrated Instrumentation (eg. AutoHelm Seatalk system).

Puis un groupe de 3 lettres pour l'identifiant de la trame.

GGA : pour GPS Fix et Date.

GLL : pour Positionnement Géographique Longitude - Latitude.

GSA : pour DOP et satellites actifs.

GSV : pour Satellites visibles.

VTG : pour Direction (cap) et vitesse de déplacement (en noeuds et km/h).

RMC: pour données minimales exploitables spécifiques.

Suivent ensuite un certain nombre de **champs** séparés par une "**virgule**". Le rôle de la virgule est d'être le séparateur de champs, qui permet la dé-concaténation des données dans le programme de traitement des données, calculateur, navigateur.

Et enfin un champ optionnel dit **checksum** précédé du signe ***** , qui représente le **OU exclusif** de tous les caractères compris entre **$** et ***** (sauf les bornes $ et *), certaines trames exigent le checksum.

Suit la fermeture de la séquence avec un **[CR][LF]**.

Un total de 82 caractères maximum pour une trame.

Soit la trame : **$GPGLL,4916.45,N,12311.12,W,225444,A**

4916.46,N *= Latitude 49 deg. 16.45 min. Nord.*

12311.12,W *= Longitude 123 deg. 11.12 min. West (ouest)*

225444 *= Acquisition du Fix à 22:54:44 UTC*

A *= Données valides*

Pas de checksum

*Non représentés **CR** et **LF***

1. Calculez le bit de parité pour les premiers caractères **$GPGLL** de la trame.

$ → $(36)_{10}$ →0x24 → $(00100100)_2$ → 0 (quand 1 paire bit 0 de parité, sinon 1)

G→0x47 → $(01000111)_2$ → 0

P→0x50 → $(01010000)_2$ → 0

L→0x4C → $(01001100)_2$ → 1

2. Quelle est la durée maximale de la transmission d'une trame.

(82*11)/4800=0,18789 s

3. Calculez la durée de transmission de cette trame en particulier.

(38*11)/4800=0,087 s

Soit la trame de type GSA suivante : **$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39**

A= *Sélection Automatique 2D ou 3D du FIX (M=Manuel)*

3 = *Fix 3D*

04,05*... = PRNs (N° d'Id) des satellites utilisés pour le FIX (maximum 12 satellites)*

2.5 = *PDOP (dilution de précision)*

1.3 = *Dilution de précision horizontale(HDOP)*

2.1 = *Dilution de précision verticale (VDOP)*

****39*** = *Checksum*

Non représentés ***CR*** *et* ***LF***

4. Indiquez le rôle du **checksum**

verifier si la trame est la bonne (verif de la somme)

5. vérifiez le calcul du checksum, vous pouvez utiliser le tableur de libre office pour effectuer ce calcul.

GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39

47 50 47 53 41 2C 41 2C 33 2C 30 34 2C 30 35 2C 2C 30 39 2C 31 32 2C 2C 2C 32 34 2C 2C 2C 2C 2C 32 2E 35 2C 31 2E 33 2C 32 2E 31

0100 0111	**0101 0000**	**0100 0111**	**0100 0011**	**0100 0001**	**0010 1100**
0100 0001	**0010 1100**	**0011 0011**	**0010 1100**	**0011 0000**	**0011 0100**
0010 1100	**0011 0000**	**0011 0101**	**0010 1100**	**0010 1100**	**0011 0000**
0011 1001	**0010 1100**	**0011 0001**	**0011 0010**	**0010 1100**	**0010 1100**
0010 1100	**0011 0010**	**0011 0100**	**0010 1100**	**0010 1100**	**0010 1100**
0010 1100	**0010 1100**	**0011 0010**	**0010 1110**	**0011 0101**	**0010 1100**
0011 0001	**0010 1110**	**0011 0011**	**0010 1100**	**0011 0010**	**0010 1110**
0011 0001					