

PR311 - Développement système

# **Transfert de fichier entre Raspberry Pi via laser**

Rapport de projet

Réalisé par Mathieu Petit, Hervé Périn, Romain Pétro

Sous la direction de Antoine Rollet et Mathieu Renard

*Année universitaire 2016 2017*

## Introduction

---

Le transfert de fichier est une opération très courante en informatique surtout quand on consulte Internet, on regarde ses méls, ou des vidéos par exemple. Derrière les quelques clics nécessaires pour l'utilisateur, un ensemble de mécanismes se met en place. Ces derniers apparaissent à plusieurs niveaux du modèle OSI, dont les couches les plus basses.

Le projet consistait à reproduire un transfert de fichier en développant en particulier la partie bas niveau, la couche physique dans le modèle OSI. Pour cela, nous avons utilisé deux ordinateurs Raspberry Pi 3 disposant pour l'un d'un émetteur laser et pour l'autre d'un récepteur de lumière. L'objectif était de partir de cela pour arriver à transférer un fichier.

Ce rapport présentera donc la mise en place du dispositif physique, la réalisation logicielle du transfert et enfin une analyse du travail réalisé.

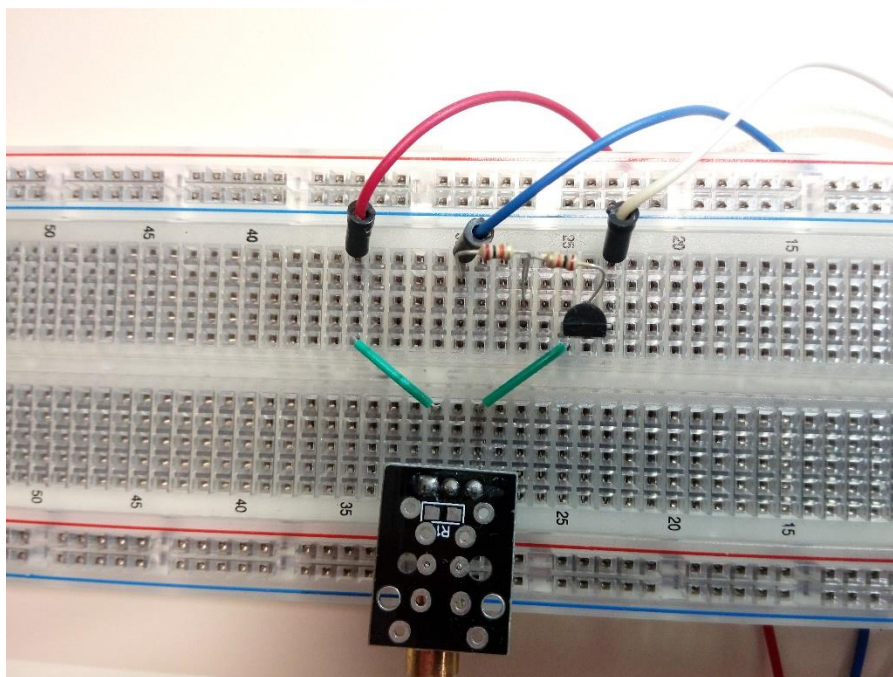
## Partie 1 – Mise en place du dispositif

La première partie du travail consistait à effectuer les branchements entre le laser, le récepteur et les Raspberry Pi.

### 1. Branchement du laser

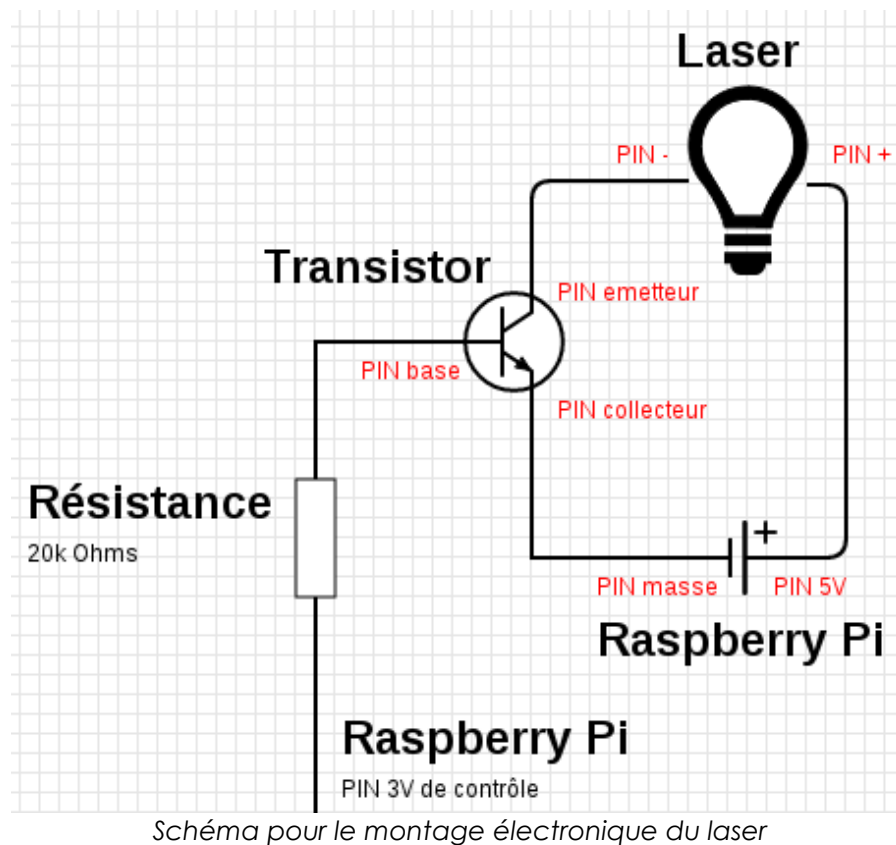
La Raspberry Pi est équipée d'une série de PIN qui permettent de brancher le laser à l'aide d'adaptateurs. Cependant, les PIN contrôlables ne fournissent que du 3V tandis que le laser a besoin de 5V pour fonctionner. Il existe un PIN pour le 5V mais il est directement branché sur l'alimentation de la Raspberry Pi et donc n'est pas contrôlable.

La solution retenue utilise un transistor. Ainsi, le circuit d'alimentation du laser peut être ouvert ou fermé via le contrôle d'un PIN. On obtient le montage de la photo suivante :



Montage électronique du laser

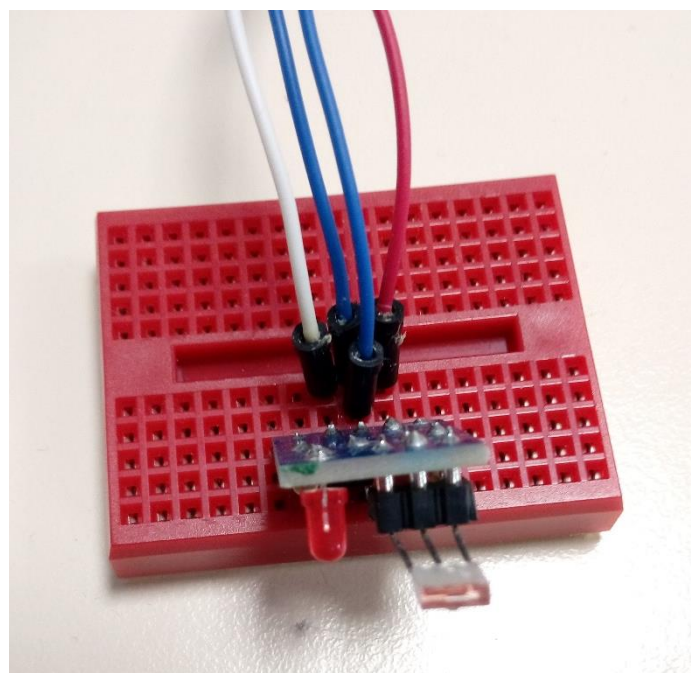
Sur ce circuit, la PIN 5V de la Raspberry Pi est branchée à l'entrée 5V du laser. La masse du laser est reliée à la patte « émetteur » du transistor. La masse de la Raspberry Pi est reliée à la patte « collecteur » du transistor. Enfin, un PIN de contrôle est relié à la patte « base » du transistor via deux résistances pour diminuer l'intensité envoyée au résistor. Le schéma du circuit correspondant est :



Avec ce montage, il est possible d'allumer ou d'éteindre le laser. Ce système ne permet pas de faire de la modulation sur la lumière.

## 2. Système récepteur

Le branchement du récepteur ne pose pas le problème de contrôle rencontré avec le laser. Le récepteur a besoin d'être alimenté d'une part et il envoie des informations sur une troisième patte que la Raspberry Pi peut lire via un PIN. La photo ci-après montre le montage effectué :



Montage électronique du récepteur

Le fil rouge apporte 5V au récepteur et le fil blanc le relie à la masse. Quant aux deux fils bleus, ils sont reliés à deux PIN contrôlable sur la Raspberry Pi. Deux fils sont nécessaires car il faut détecter deux fronts et que la configuration spécialise un PIN sur un front (un PIN pour détecter le front montant, d'éteint à allumé et un PIN pour détecter le front descendant, de allumé vers éteint). Cela donne schématiquement le circuit suivant :

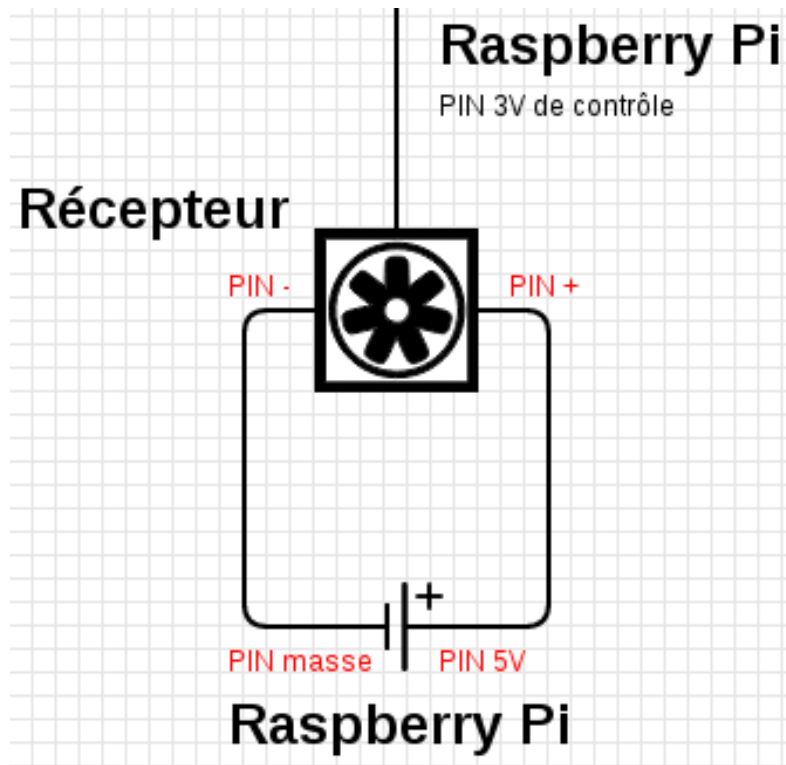


Schéma pour le montage électronique du récepteur

Du coup, le récepteur est connecté à sa Raspberry Pi.

## Partie 2 – Développement du programme pour le transfert

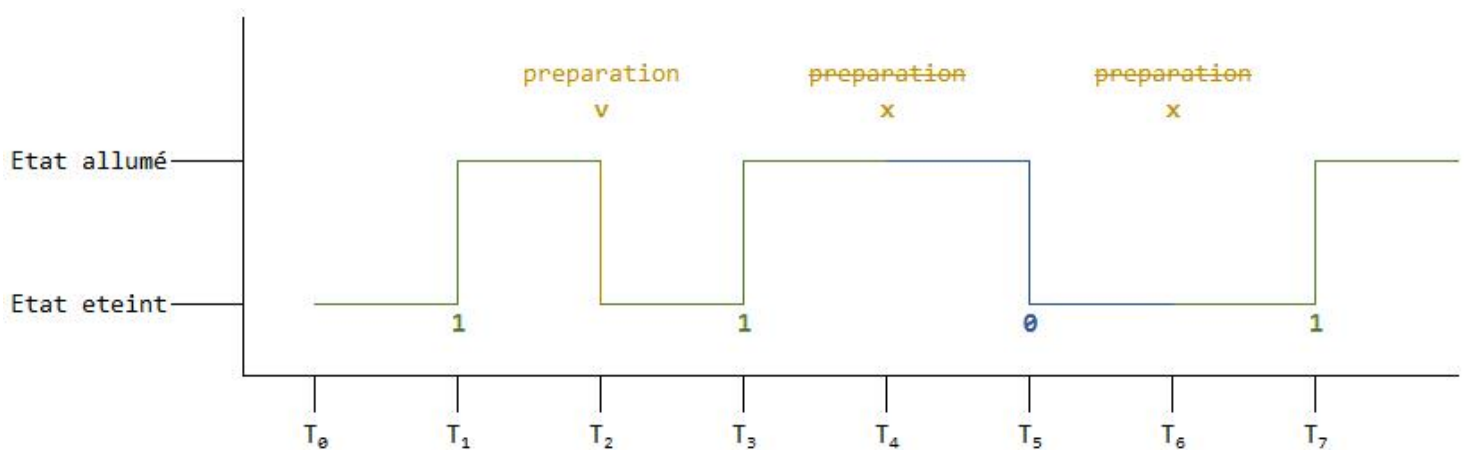
Maintenant que le laser et le récepteur sont branchés à leur Raspberry Pi, il faut développer le transfert de fichier. Cela conduit à un programme qui contrôle le laser et envoie un fichier et un programme contrôlant le récepteur pour recevoir un fichier.

### 1. Principe à la base du transfert : le codage Manchester

Pour transmettre des bits, il faut que d'une part le laser, l'émetteur, change d'état et que d'autre part, le récepteur détecte ce changement d'état. Le montage électrique dont nous disposons ne permet de jouer qu'avec deux états : laser allumé et laser éteint. Dans ce cas, le fonctionnement proposé par le codage Manchester est le plus évident à utiliser.

Ce dernier se base sur la détection par le récepteur d'un changement d'état de l'émetteur. Dans notre système, le changement d'état, c'est le passage du laser éteint au laser allumé et inversement. Cela représente donc deux fronts : le front montant d'éteint vers allumé et le front descendant, d'allumé vers éteint. Dans notre application, le front montant modélise le bit 1 tandis que le front descendant modélise le bit 0.

La transmission d'un bit, autrement dit d'un front, se passe en deux temps. Le temps de préparation pendant lequel, le laser va être positionné dans l'état initial du front à transmettre et le temps de transfert où le laser passera dans l'état final du front à transmettre. Pendant le temps de préparation, le laser peut ne pas changer d'état. Le schéma suivant résume le concept :



Codage Manchester appliqué à la séquence de bits 1101

Considérons la transmission du deuxième bit. Cela commence au temps T<sub>2</sub> où le laser va s'éteindre. Puis, à T<sub>3</sub>, il va s'allumer. Sur le schéma, le récepteur s'intéresse qu'aux tics d'horloge, soit les temps impairs (les temps pairs correspondent au temps de demi tic d'horloge) : c'est le front reçu à cet instant qui l'intéresse. La transmission des troisième et quatrième bits ne nécessite pas d'action particulière pendant le temps de préparation car le laser est déjà dans le bon état.

## 2. Première version du protocole d'échange de fichier

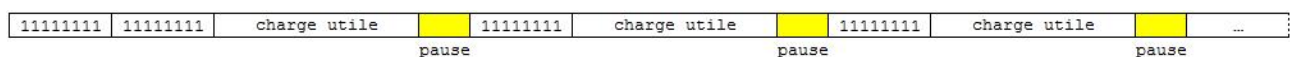
En se basant sur le codage de Manchester, la première étape du travail consiste à fixer un protocole pour communiquer entre l'émetteur et le récepteur. Ce protocole va permettre par exemple de synchroniser les deux entités pour que le récepteur comprenne ce qu'envoie l'émetteur. Notons que dans le système mis en place, il n'y a pas de possibilité pour le récepteur de communiquer avec l'émetteur. Autrement dit, ce dernier ne sera jamais informé de ce que le récepteur a reçu.

Le système laser émet des trames, qui correspondent à un découpage des données à transmettre. Ces dernières sont composées comme ci-après :

- Seize fois le bit 1 pour indiquer au récepteur le début d'une nouvelle trame.
- X octets correspondants à la charge utile de la trame en question. Ce chiffre est un paramètre de l'émetteur. Rappelons qu'un octet, c'est huit bits.
- **Un temps de pause. Le récepteur, ne recevant plus rien, se préparera à recevoir huit fois le bit, indiquant une réémission de la charge utile.**
- **Huit fois le chiffre 1 pour commencer la réémission de la charge utile.**
- **X octets correspondants à la charge utile de la trame en question. Ce chiffre est un paramètre de l'émetteur. Ce sont les mêmes qu'au point numéro deux.**
- Les trois points précédents, en gras, sont répétés autant de fois que l'indique la configuration du laser concernant la réémission de la charge utile de la trame.

Concernant le dernier point, pour pallier au problème de l'absence de message entre le récepteur et le laser qui auraient permis d'indiquer une erreur de transmission, un nombre de réémission de la charge utile de chaque trame est configuré dans les programmes.

Le schéma correspondant à ces trames est le suivant :



*Trame du premier protocole*

Les trames sont issues du découpage du fichier à transmettre par le laser : tous les x octets, le fichier est découpé et chaque coupe est la charge utile d'une trame. Le laser émet dans l'ordre les trames. Il n'y a pas de mécanisme pour ordonner les trames. La dernière trame fait au maximum la taille des autres mais peut être plus petite.

Du côté du récepteur, ce protocole impose de savoir à l'avance le nombre de répétitions de la charge utile et sa taille. De plus, une trame peut être

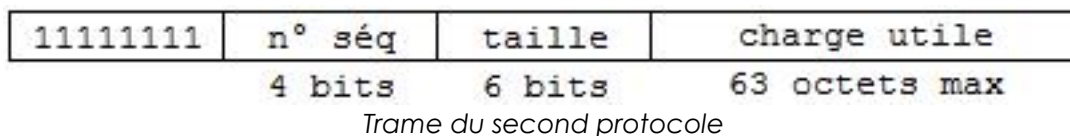
intégralement perdue si le décodage des huit premiers bits rate entraînant la perte de la charge utile et de ses répétitions.

### 3. Deuxième version du protocole d'échange de fichier

Dans la deuxième version du protocole, le découpage du message en trame existe toujours et le concept de réémission de la charge utile d'une trame plusieurs fois aussi. A cela s'ajoute plusieurs avantages :

- Dans un premier temps, le récepteur n'a plus besoin d'être configuré. C'est-à-dire qu'il n'y a pas besoin de lui indiquer par avance le nombre de répétitions de la charge utile ou la taille de cette dernière. Ces deux informations sont soit dans la trame, soit détectables automatiquement en fonction d'un ensemble de trames.
- Ensuite, la trame gagne en simplicité : elle ne contient plus que cinq champs et non un nombre variable dépendant du nombre de répétitions, comme dans la version précédente du protocole.
- Troisièmement, le nouveau protocole permet, quand on perd une trame et sa charge utile, de ne pas perdre les répétitions. En effet, chaque répétition est dans une trame séparée donc perdre une trame ne veut pas dire perdre les répétitions donc la charge utile peut être quand même obtenue.

Ainsi, le schéma suivant représente la trame du nouveau protocole :



La composition de cette trame se définit comme suit :

- Huit fois le chiffre 1 pour indiquer au récepteur la début d'une trame.
- Un numéro de séquence qui distingue les trames de réémission des trames contenant une nouvelle charge utile. Si deux trames qui se suivent ont le même numéro de séquence, cela veut dire que la deuxième répète la charge utile de la première. Si les numéros sont différents, c'est que la deuxième contient une charge utile nouvelle. Les numéros de séquence sont compris entre 0 et 15.
- Six bits permettant de coder la taille de la charge utile. Cette taille varie entre 0 et 63 octets.
- La charge utile pouvant faire au maximum soixante-trois octets.

Dans ce protocole, un temps de pause sépare réellement deux trames contrairement à l'ancien protocole où une pause apparaissait entre deux réémissions pourtant dans une même trame.



Enfin, la répétition de la charge utile se fait par l'envoi de la même trame plusieurs fois. Le récepteur détecte qu'il a reçu des trames avec le même numéro de séquence ensemble : c'est la même charge utile. Le traitement de ces trames identiques est expliqué dans la gestion des erreurs, ci-après.

Notons que le numéro de séquence des trames boucle comme dans le protocole IP. C'est l'horodatage qui permet de réorganiser tous les numéros de séquence.

#### 4. Mécanismes supplémentaires

Les protocoles clarifient l'échange des données au niveau de la transmission mais pas dans l'univers que sont les systèmes qui les exécutent. Il faut donc rajouter quelques modules pour finir l'intégration.

##### a. Synchronisation entre le laser et le récepteur

Dans les deux protocoles, la synchronisation entre le laser et le récepteur se fait de la même manière.

Avant de commencer à émettre la première trame, le laser envoie huit fois un front montant (un bit 1). Le récepteur les détecte et stocke pour chacun l'heure de réception. Puis il calcule la médiane ce qui lui permet d'obtenir la fréquence. Ensuite, à la réception de chaque bit, le temps au bout duquel le prochain bit sera reçu est calculé en fonction de la fréquence découverte initialement.

Notons qu'après l'envoi des huit premiers bits de synchronisation, un temps de pause est marqué avant d'émettre la première trame du message.

##### b. Gestion des erreurs

A chaque réception d'un bit, le récepteur retient l'heure d'arrivée. Cette dernière est comparée à l'heure théorique d'arrivée issue d'un calcul en fonction de l'heure réelle d'arrivée du dernier bit et de la fréquence de l'émetteur calculée au moment de la synchronisation.

Le résultat de la comparaison permet d'indiquer au récepteur s'il y a eu préemption dans l'ensemble du système de transfert. Deux cas sont en effet possibles : soit le microprocesseur de la Raspberry Pi du laser a préempté le programme d'émission, soit celui de la Raspberry Pi de l'émetteur a préempté le programme de réception. Dans les deux cas, le résultat est le même : l'écart de temps est trop grand par rapport à la tolérance.

Si ce problème est détecté par l'émetteur, la trame en cours de réception sera jetée sans autre traitements. Il n'est pas possible de savoir où se situe, dans la trame, le problème. Et il n'y a pas de mécanisme pour réparer l'erreur. Par contre, grâce à la pluri émission des trames, les données arriveront quand même au récepteur.

Ainsi, quand une trame est reçue correctement, le récepteur ignore toutes celles qui participent à la réémission de la charge utile de la trame correctement reçue.

Notons un autre mécanisme de sécurité. Comme le programme transfère des fichiers textes ASCII, chaque octet de la charge utile commence par le bit 0 donc le récepteur jette la trame entière si l'un des octets de la charge utile commence par le bit 1. Pour transférer n'importe quel fichier, les programmes traitant les fichiers sous leur forme binaire, il suffit de désactiver cette sécurité.

#### c. Gestion des temps d'attente

Entre le mot de synchronisation initial et les trames ainsi qu'entre chaque trame (cas du deuxième protocole), il y a un temps de non émission de la part du système laser.

Lors de nos premiers tests, nous utilisons une attente passive, c'est-à-dire que le programme rendait la main au processeur et demandait à être réveillé au bout du temps prévu.

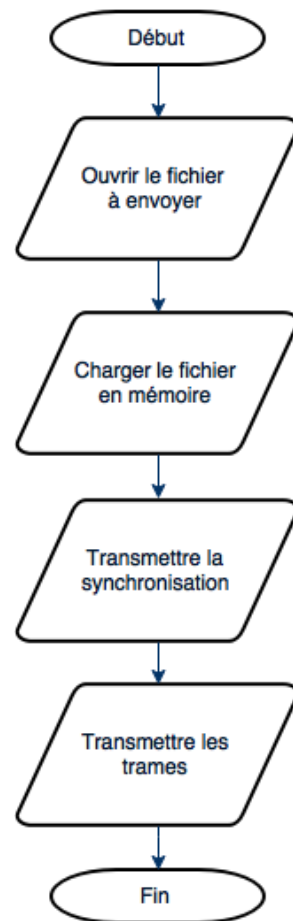
Avec une utilisation du laser à une fréquence très haute (très peu de temps entre chaque tic de l'horloge), cela bloquait l'horloge. En effet, quand le programme demandait 200 microsecondes d'attente passive au processeur, ce dernier ne le réveillait qu'après 320 microsecondes donc aucune donnée n'était recevable.

La version actuelle fait donc de l'attente active pendant les temps de pause mis en place par le protocole.

#### d. Traitement, émission et réception

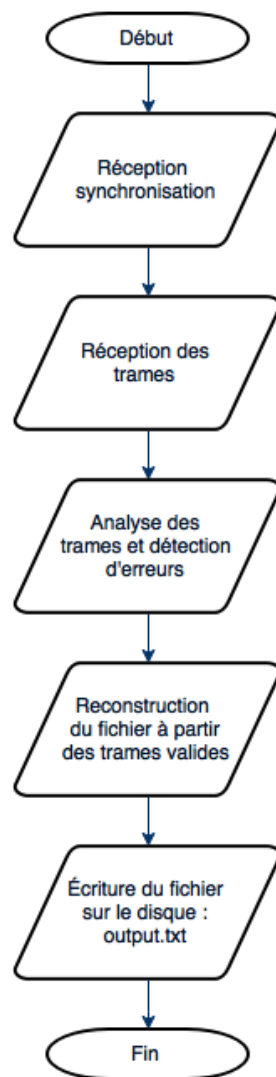
Le programme pour le laser comme celui pour le récepteur fonctionnent en séparant la partie traitement des données de la partie émission. Cela limite l'impact de la préemption par le microprocesseur.

Pour le laser, le programme commence par récupérer tous les bits à transmettre en lisant le fichier en entrée puis il les envoie en faisant une simple lecture des données préparées. Le schéma suivant résume le processus :



*Fonctionnement macro du processus implémenté dans le laser*

Le récepteur enregistre les données reçues en les stockant de manière simple pour limiter le temps de travail puis, après la fin de la transmission, il les traite pour reformer le fichier reçu. Autrement écrit :



Fonctionnement macro du processus implémenté dans le récepteur

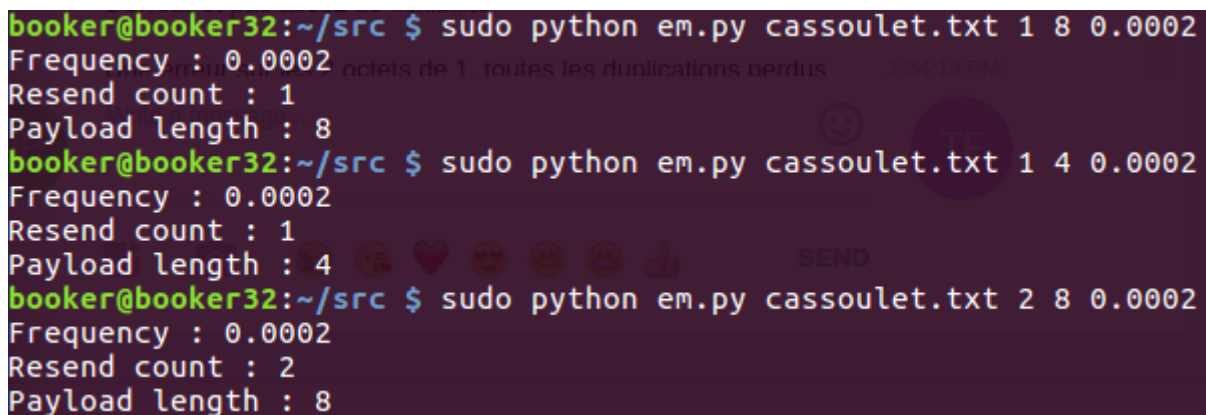
## Partie 3 – Analyse et bilan

---

Maintenant que le laser et le récepteur sont branchés à leur Raspberry Pi, il faut développer le transfert de fichier. Cela conduit à un programme contrôlant le laser et envoyant un fichier et un programme contrôlant le récepteur pour recevoir un fichier.

### 1. Essai du système

Pour tester le système à sa fréquence la plus élevée, 200 microsecondes, nous avons fait trois transferts. Le premier consistait à envoyer des trames avec une charge utile de 8 octets sans réémission, le deuxième, 4 octets sans réémission et le dernier, 8 octets avec une réémission. L'émetteur laser affiche l'image suivante :



```
booker@booker32:~/src $ sudo python em.py cassoulet.txt 1 8 0.0002
Frequency : 0.0002 octets de 1 : toutes les duplications perdus 3:24:18 PM
Resend count : 1
Payload length : 8
booker@booker32:~/src $ sudo python em.py cassoulet.txt 1 4 0.0002
Frequency : 0.0002
Resend count : 1
Payload length : 4
booker@booker32:~/src $ sudo python em.py cassoulet.txt 2 8 0.0002
Frequency : 0.0002
Resend count : 2
Payload length : 8
```

*Affichage fourni par l'émetteur lors de l'envoi des trois fichiers*

Pour détailler la première ligne de commande, on demande d'effectuer le transfert du fichier cassoulet.txt par le script python em.py en envoyant une fois chaque morceau de 8 octets et en utilisant une fréquence de 200 microsecondes. Le tout est exécuté en tant qu'utilisateur privilégié pour pouvoir contrôler les PIN de la Raspberry Pi sur lesquelles le laser est branché.

Du côté du récepteur, l'affichage graphique donne l'illustration suivante :

```
booker@booker31:~/src $ sudo python rcpt.py && cat output.txt
Frequency : 0.00022292137146 s
Laser speed : 2932.31202263 bps
Download speed : 2291.60679763 bps
????????rin, romain petro et mathieu petit sont a l'origine de ce magnifique pro????
????ransfert des donnees via laser. Cette nouvelle technolog????????volutionner le m
onde de ?????????atique et particulierement nos amis de france telecom qui vont enfin
pouvoir fournir un ?????????a leurs clients de plus de 1Gb/s ... Nous embrassons a c
ette occasion l'inventeur du cassoulet.
booker@booker31:~/src $ sudo python rcpt.py && cat output.txt
Frequency : 0.000220060348511 s
Laser speed : 2694.47463269 bps
Download speed : 1725.56518951 bps
herve perin, romain petro et mathieu petit sont a l'origine de ce magnifique projet
de transfert des donnees via laser. Cette nouvelle technologie va revolutionner le m
onde de l'informatique et particulierement nos amis de france telecom qui vo????nfin
pouvoir fournir un serv????a leurs clients de plus de 1Gb/s ... Nous embrassons a c
ette occasio????inventeur du cassoulet.
booker@booker31:~/src $ sudo python rcpt.py && cat output.txt
Frequency : 0.000221014022827 s
Laser speed : 3175.8723562 bps
Download speed : 1242.26580591 bps
herve perin, romain petro et mathieu petit sont a l'origine de ce magnifique projet
de transfert des donnees via laser. Cette nouvelle technologie va revolutionner le m
onde de l'informatique et particulierement nos amis de france telecom qui vont enfin
pouvoir fournir un service a leurs clients de plus de 1Gb/s ... Nous embrassons a c
ette occasion l'inventeur du cassoulet.
```

*Affichage fournit par le récepteur lors de la réception des trois fichiers*

On peut observer que :

- la fréquence calculée est plus grande que celle demandée en émission de près de 22 microsecondes. Ceci est dû à la latence lors de l'exécution du système d'exploitation, du laser et du récepteur.
- le débit utile dans le premier cas représente 78,15% du débit total, 64,04% dans le second cas et 39,13% dans le troisième cas. Ce débit dépend, en effet, de la configuration demandée à l'émission. Le meilleur débit de notre système serait de 96,55% du débit total (63 octets de charge utile) dans la configuration actuelle des trames et du temps d'attente entre chaque (5 millisecondes).
- au niveau des erreurs, il manque 5 trames soit 40 octets pour le premier test, 3 trames soit 12 octets pour le deuxième test et aucune perte pendant le troisième test. La réémission fait chuter le débit mais supprime énormément d'erreurs dues à la préemption par le microprocesseur.

Notons que le fichier servant au test des transferts fait 377 octets et contient le texte :

```
herve perin, romain petro et mathieu petit sont a l'origine de ce
magnifique projet de transfert des donnees via laser. Cette nouvelle
technologie va revolutionner le monde de l'informatique et
particulierement nos amis de france telecom qui vont enfin pouvoir
fournir un service a leurs clients de plus de 1Gb/s ... Nous
embrassons a cette occasion l'inventeur du cassoulet.
```

Le tableau suivant présente les résultats d'un ensemble de tests sur le même fichier que précédemment :

Fréquence	Temps de pause	Débit utile	Débit total	Charge utile	Nombre d'émission	Taux de réussite des transferts
0,0002 s	0,005 s	2,292 kbps	2,932 kbps	8 B (64 b)	1	0,00 %
0,0002 s	0,005 s	1,726 kbps	2,694 kbps	4 B (32 b)	1	0,00 %
0,0002 s	0,005 s	1,242 kbps	3,176 kbps	8 B (64 b)	2	20,00 %
0,0002 s	0,0006 s	3,295 kbps	3,407 kbps	63 B (504 b)	1	0,00 %
0,002 s	0,006 s	458 bps	475 bps	63 B (504 b)	1	99,99 %
0,0002 s	0,001 s	1,467 kbps	3,750 kbps	8 B (64 b)	2	20,00 %
0,0002 s	0,006 s	1,310 kbps	3,953 kbps	8 B (64 b)	3	80,00 %
0,0002 s	0,0006 s	787 bps	4,000 kbps	8 B (64 b)	4	99,99 %

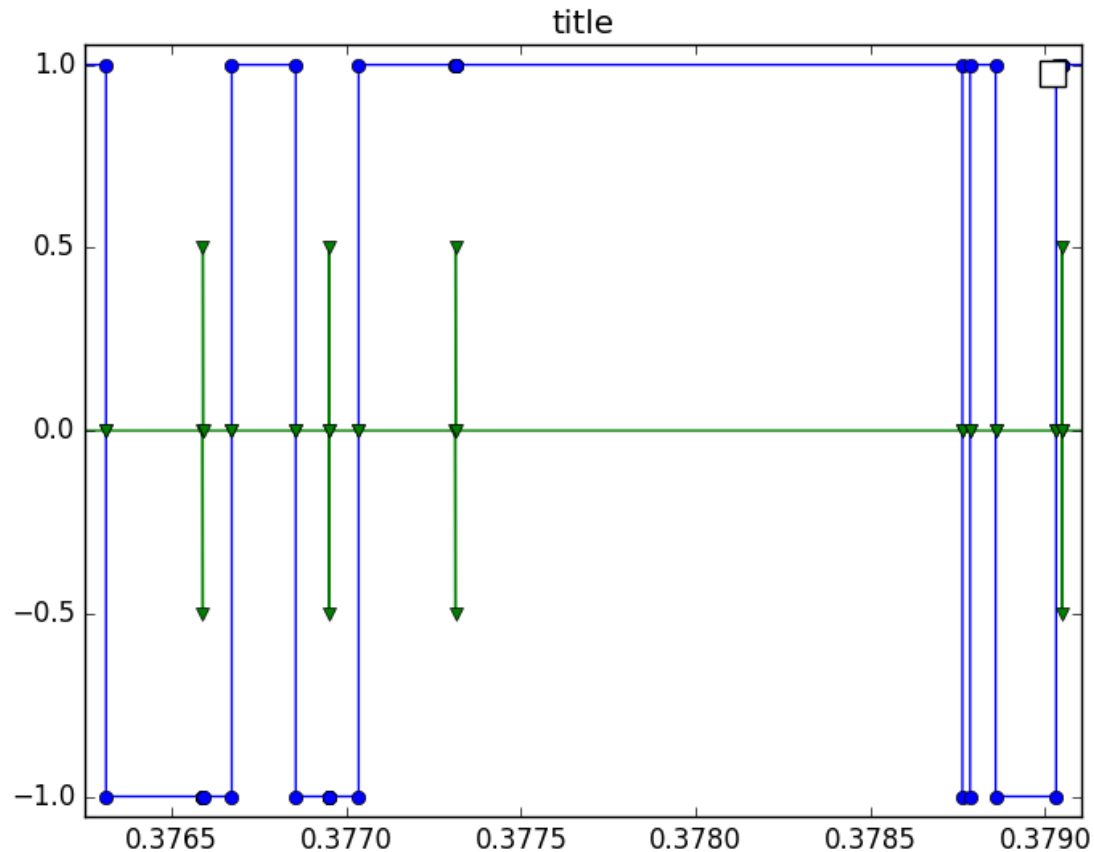
Tableau récapitulatif des tests significatifs effectués sur le système

Une petite fréquence du laser garantit un transfert sans erreurs malgré l'absence de répétition de la charge utile car cela absorbe la préemption du microprocesseur. A grande fréquence, il faut une petite charge utile répétée plusieurs fois pour obtenir un transfert sans erreurs. Le débit est limité soit par la faible fréquence soit par la faible charge utile. Le meilleur débit utile que peut proposer notre système est de presque 3,3 kbps mais donne un fichier totalement erroné à l'arrivée.

## 2. Modélisation du problème de préemption par le microprocesseur

Contrairement à une carte réseau qui gère la réception des messages indépendamment du microprocesseur et qui les lui transmet quand il les demande, notre programme est entièrement dépendant du système, de sa mémoire et du microprocesseur. La grande conséquence réside dans le non contrôle des périodes d'émission ou de réception.

La différence se fait ressentir dans la gestion de la réception : pour comprendre le message, il faut que les deux entités soient synchronisées. Or quand le microprocesseur met l'une des deux en pause, cela casse la synchronisation. Le graphique suivant met en avant ce problème :



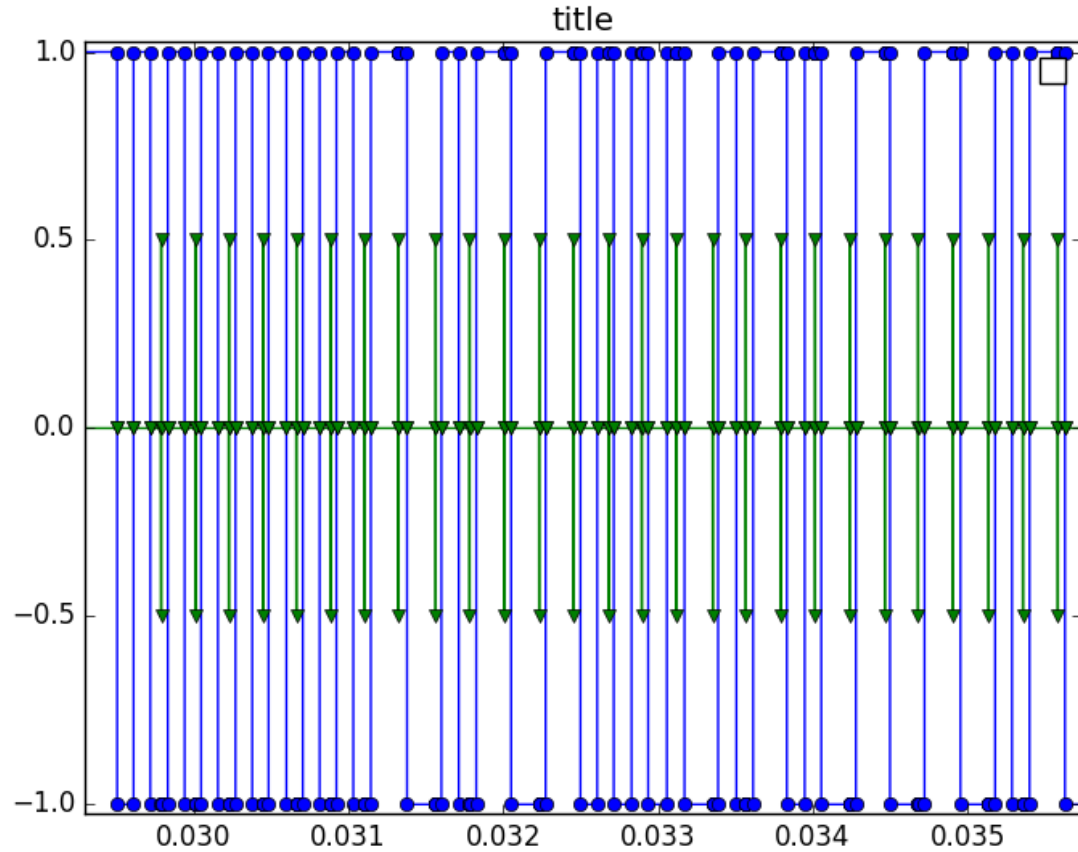
*Morceau de trame reçue lors d'une préemption par le microprocesseur*

Après être synchronisé, le récepteur sait à quelle heure il doit recevoir chaque bit. Pour prendre en compte la latence des différents systèmes concernés par la transmission, le récepteur recommence l'écoute au bout de 75% du temps d'attente théorique entre deux réceptions, moment symbolisé par les traits verts. Si on considère les réceptions après 0.3765 secondes, on observe que le récepteur reçoit un bit 1 (front montant) puis le laser s'éteint avant de retransmettre un bit 1. Et là, le récepteur ne reçoit plus rien pendant 1,4 millisecondes. Il y a donc eu un phénomène de préemption : le programme n'a pas reçu le bit au bon moment au regard de l'horodatage.

Deux conséquences apparaissent. Le récepteur ne sait pas si c'est lui ou le laser qui a été préempté donc on ne peut pas imaginer un mécanisme où quand c'est le laser, on peut reprendre la réception. De plus, le traitement de l'erreur ne pouvant être corrigé, toute la trame reçue est jetée. Ce qui, dans le cas du premier protocole, implique la perte intégrale de la charge utile.

Le graphique suivant montre comment se passe une réception normale :





*Trame reçue sans préemption par le microprocesseur*

On retrouve en vert le moment où le récepteur se remet à écouter, un peu avant le temps théorique d'arrivée d'un bit et en bleu le temps réel. Le début du graphique montre l'arrivée des 8 bits indiquant le début d'une nouvelle trame. La réception se fait régulièrement au fil du temps et les espaces blancs mettent en avant l'absence de repositionnement du laser dans le bon état (la mention « ~~recalibrage~~ » précédemment rencontrée).

## Conclusion

---

Mettre en place un transfert de fichier nécessite un travail en plusieurs étapes. Tout d'abord, les branchements entre le laser ou le récepteur et leur Raspberry Pi qui vont aussi conditionner les méthodes physiques utilisées pour la transmission. Dans notre cas, la modulation n'était pas envisageable. Ensuite, il faut mettre en place une technique de transfert des bits, le codage Manchester, et un algorithme pour mettre d'accord émetteur et récepteur sur la méthode de transfert des données. Finalement, il faut mettre à l'épreuve le système pour mesurer sa fiabilité et ses caractéristiques.

Dans notre cas, la vitesse de la lumière ne fait pas tout. Cela se voit par nos débits plus faibles que les vieilles technologies pour accéder à Internet par exemple. Mais c'est dû à deux limites : l'exécution des programmes liés au bon vouloir des microprocesseurs des deux Raspberry Pi d'une part et à la configuration unidirectionnelle de notre système nécessitant une réémission de la même charge utile plusieurs fois pour espérer qu'elle soit reçue sans erreurs. Internet, d'ailleurs, ne permet pas de trouver de débit de référence pour ce type de système.

Un autre problème de taille apparaît à l'usage aussi. Il faut, avec le laser, viser le récepteur. C'est assez facile à quelques centimètres de distance entre les deux éléments mais à partir de quelques mètres, c'est beaucoup plus dur. De plus, avec l'éloignement, la lumière ambiante peut polluer le faisceau laser. Ces motifs font peut-être partie de ceux retenus pour qu'aujourd'hui, ce système ne soit pas le plus répandu quand il s'agit de connecter des équipements entre eux.