Activité :
Décoder les trames infrarouges des télécommandes

L'objectif de cette activité est d'acquérir et de décoder des **trames de données** envoyées par une **télécommande IR** et de réutiliser cette télécommande pour commander des Leds.

Après une période de commande par ultrasons, les télécommandes à infrarouges se sont imposées comme un quasi standard depuis 1975 en raison de leur faible coût (pour le fabricant), de leur portée importante et de leur sécurité de transmission.

Objectif de l'activité

L'objectif de cette activité est d'acquérir et décoder des **trames de données** envoyées par une **télécommande IR** selon le protocole NEC et d'obtenir sont décodage en Hexadécimale dans un terminal série Arduino.

Quelques bases de la transmission infrarouges

Les télécommandes IR sont partout, facilement récupérables sur de nombreux objets du quotidien et à défaut, leur coût d'acquisition est dérisoire.

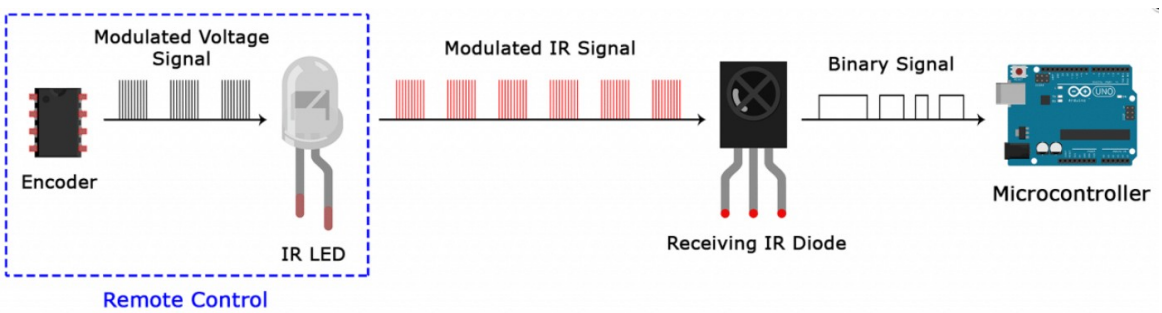
Pour transmettre des informations à un équipement (un téléviseur par exemple), il est nécessaire que la TV et la télécommande utilise le même **protocole**. Cela signifie que n'importe télécommande ne peut pas commander n'importe quel équipement !

Par contre, il est possible de détourner l'utilisation d'une télécommande en la réutilisant pour commander tout autre équipement comme des leds, des moteurs, etc.

Il existe de nombreux protocoles de transmissions (**NEC**, Sony SIRC, Philips RC5, Philips RC6, données brutes...) et plusieurs manières de coder les informations. Heureusement une librairie pour Arduino, « **Irremote.h »,** gère tout ça pour nous facilement.

➢ Principe de fonctionnement

Une diode émet des pulses en infra-rouge pour transmettre un signal de télécommande codé, vers à un récepteur infra-rouge placé dans un appareil que l'on commande à distance. Dans notre cas, on récupérera le code dans une carte Arduino.



La portée est de plusieurs mètres, en ligne droite sans obstacles, le signal est invisible à l'œil humain.

Les signaux IR sont modulés à 38 kHz pour éviter les interférences avec la lumière ambiante. Le récepteur 38 kHz gère cette modulation directement.

Matériels nécessaires

- Une carte Arduino
- un oscilloscope (pour plus de confort utiliser les modules Picoscope)
- Le logiciel Picoscope
- un récepteur infrarouge VS1838B
- un breadboard (plaque de prototypage rapide).
- Une télécommande utilisant le protocole NEC.
- Une résistance de 22 kΩ.

Identification du protocole utilisé par la télécommande

- **Réaliser** le cablage conformément à la figure suivante :

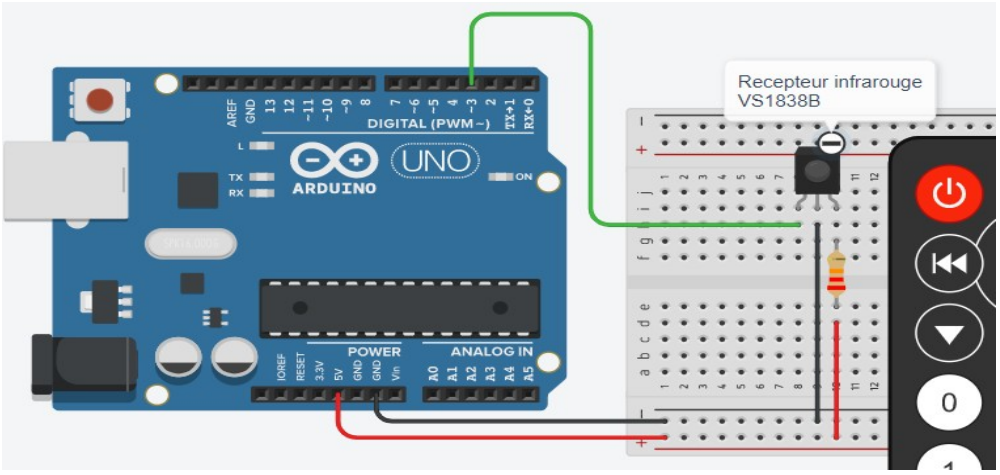


Figure 1

En cliquant sur l'image vous accéderez à la simulation TinkerCad (qui ne fonctionnement pas d'ailleurs) où vous y trouverez le code à téléverser ou directement en téléchargeant le dossier « detectionProtocole ».

- **Téléverser** le programme dans la carte Arduino, **lancer** le terminal serie et **appuyer** sur n'importe quelle touche de la télécommande la dirigeant vers le récepteur infrarouge.
- **Identifier** le protocole utilisé.

Extraction des codes hexadécimaux d'une télécommandes

A partir du câblage réaliser figure 1 :

- **Télécharger** le programme decoderMessagesRecus.ino
- **Téléverser** le programme dans la carte Arduino, **lancer** le terminal série et **appuyer** sur n'importe quelle touche de la télécommande la dirigeant vers le récepteur infrarouge.
- **Relever** le code hexadécimal reçu par la carte Arduino pour l'ensemble des touches de la télécommande (présenter sous forme de tableau).

Analyse d'une trame

- **Prendre** connaissance de l'annexe relatif au protocole NEC (il est indispensable de bien le comprendre pour réaliser la suite).

Pour rappel, le code hexadécimal interprété par la carte Arduino pour la touche 1 est FD08F7.

- **Lancer** le logiciel Picospoe et ouvrir le fichier « touche1TelecommandeGriseMarcheModeMute.psdata »
- **Zoomer** correctement sur la trame contenant le message (exclure le « Repeat Code », répétition NEC) et **imprimer** la trame.

Après le « Leading Pulse » et le « Space :
- **Identifier** les 0 logiques et les 1 logiques
- **Réaliser** des paquets de 8 bits.
- **Identifier** les octets d'adresse et de commande.
- **Convertir** les octets d'adresse et de commande en hexadécimal et les **comparer** avec l'interprétation Arduino
- **Indiquer** si la carte Arduino nous donne la combinaison des valeurs hexadécimales dans le bon ordre ? **Justifier**

Capture et analyse de trame

- **Proposer** un montage à base d'oscilloscope pour capture la trame en sortie du récepteur.
- **Procéder** à la capture de la trame et à son analyse avec comparaison du code Arduino interprété pour la touche de votre choix de la télécommande.

Commande d'un led à partir de la télécommande

- **Proposer** un schéma de câblage et un code permettant de piloter un led à partir de votre télécommande.
- **Valider** votre code par l'expérimentation.

Pour aller plus loin

- Protocoles de télécommandes Infrarouge.
- Codes hexadecimaux de nombreuses télécommandes
- tiptopboards

ANNEXE PROTOCOLE NEC

Le **protocole NEC** a été développé par l'entreprise japonaise NEC (aujourd'hui Renesas) et a été adopté par les principaux fabricants japonais d'électronique grand public
Dans le protocole NEC, chaque fabricant se voit attribuer un code unique qui est contenu dans la commande transmise, ce qui évite la possibilité de faux déclenchements par d'autres combinés distants.

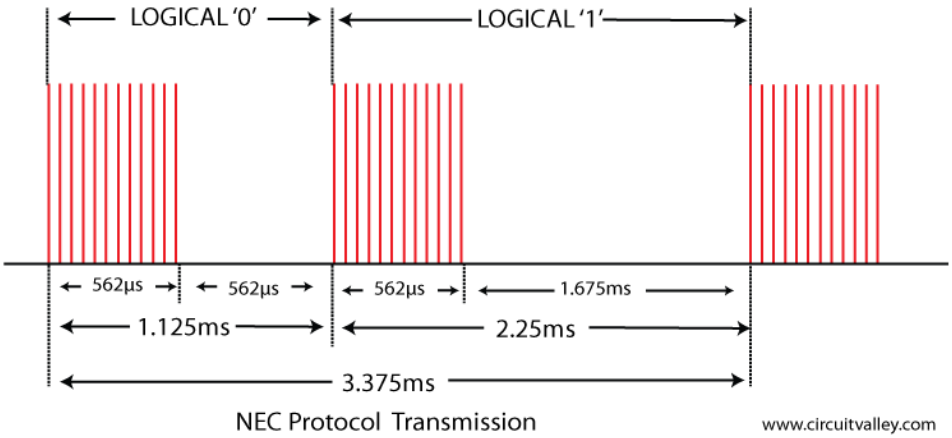
➢ Caractéristiques

- **Adresse** de 8 bits (mode étendu disponible, doublant la taille de l'adresse)
- **Commande** de 8 bits
- L'adresse et la commande sont transmises deux fois pour plus de fiabilité.
- Fréquence porteuse de 38 kHz.
- Temps de bit de 1,125 ms ou 2,25 ms.

➢ Modulation

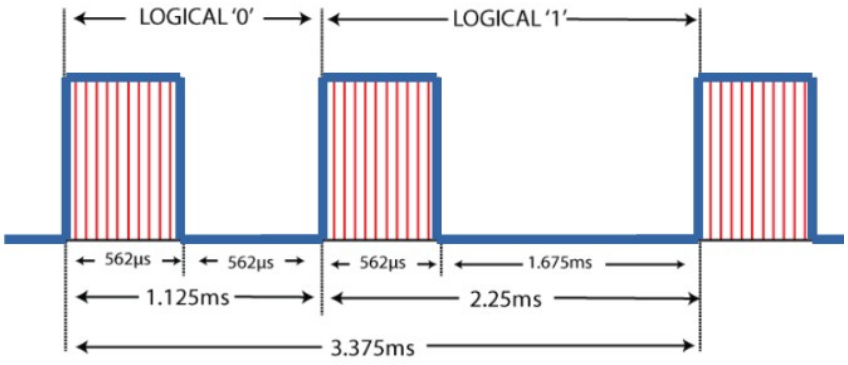
Le protocole NEC utilise le **codage par distance d'impulsions** (*Pulse Distance Encoding*).
Chaque impulsion est une salve de porteuse de 38 kHz d'une longueur de 560 µs (soit environ 21 cycles).

- Un « 1 » logique prend 2,25 ms pour être transmis,
- Un « 0 » logique n'en représente que la moitié, soit 1,125 ms.



➢ Démodulation

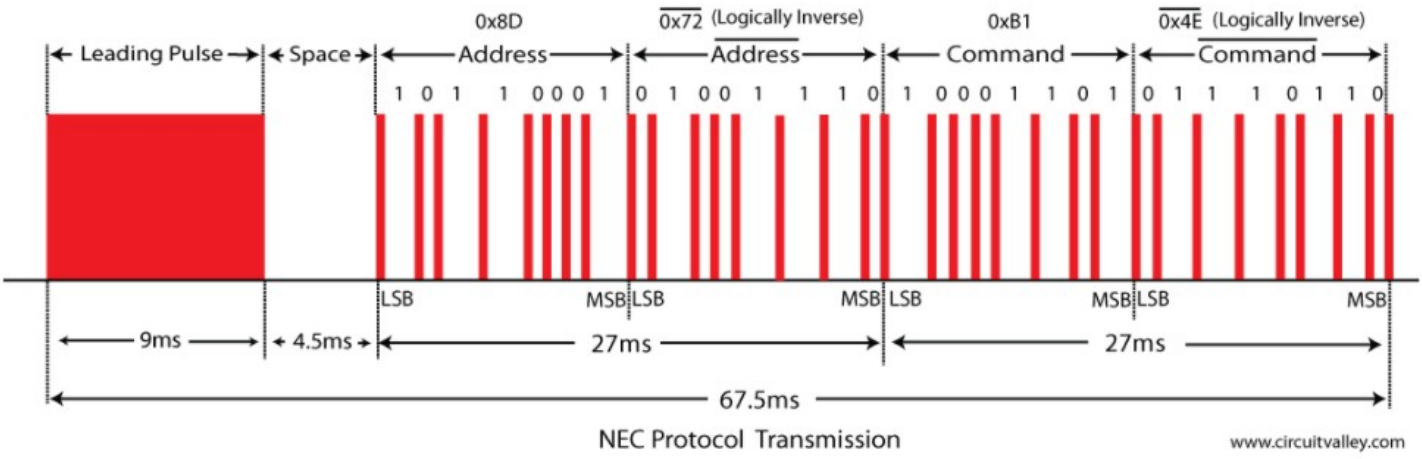
Le récepteur infrarouge va capter ce signal modulé à 38kHz et va le démoduler de la manière suivante (traits épais) convertissant les salves de modulation en signaux hauts de largeur $562\,\mu s$



Très important : Prendre soin de noter la manière dont le **0 logique** et le **1 logique** sont codés.

➢ Protocole

L'image ci-dessous montre un train d'impulsions typique du protocole NEC :



Cette trame est constituée ainsi :

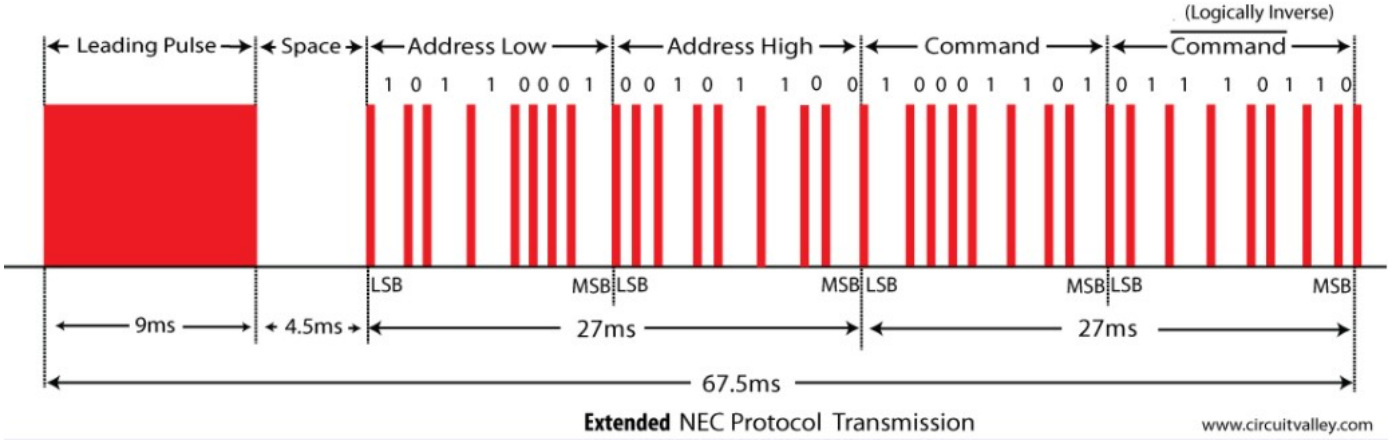
- Un **code de démarrage**, composé d'une impulsion de 9 ms suivie d'une pause de 4,5 ms ;
- Les octets d'adresse et de commande dont les bits sont codés par distance d'impulsion

Les octets sont transmis en mode *LSB First* : **bit de poids faible en premier**.

Chaque octet (l'adresse et la commande) est transmis deux fois (on parle de **redondance**) : une fois « normalement », et une fois avec tous les bits inversés (cette technique permet d'augmenter la fiabilité de la transmission) ;

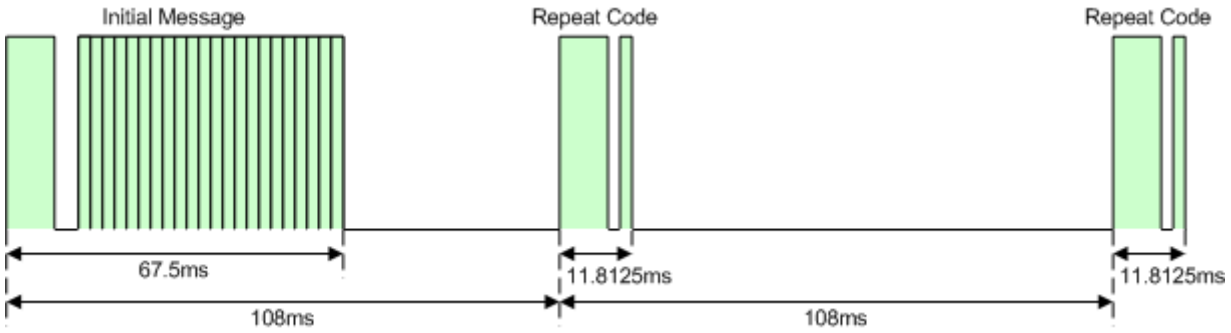
➢ Protocole NEC étendu

Le protocole NEC est si largement utilisé que toutes les adresses possibles ont rapidement été épuisées. En sacrifiant la redondance des adresses, la plage d'adresses a été étendue de 256 valeurs possibles (8 bits) à environ 65 000 valeurs différentes (16 bits).



➢ Répétition NEC

Une commande n'est transmise qu'une seule fois, même si la touche de la télécommande reste enfoncée.
Toutes les 110 ms, un code de répétition est transmis tant que la touche reste enfoncée. Ce code de répétition est simplement une impulsion de 9 ms suivie d'un espace de 2,25 ms et d'une nouvelle impulsion de 560 µs.



➢ Réception des trames

Attention quand à l'interprétation des trames. En raison de la technologie utilisée, les signaux hauts et bas sont inversés à la réception par rapport à l'émission.