

# Flipper Zero : reproduire les codes d'une télécommande infrarouge

**La platine minidisc de Radio Kerne (Sony MDS-JE510) peut insérer des pauses à la lecture d'un MD (3 secondes de blanc), ces pauses peuvent permettre avec sox de faire une séparation des pistes, ce qui sera pratique car il reste une cinquantaine de minidisc à numériser.**

Caractéristiques de la platine : <https://www.minidisc.wiki/equipment/sony/deck/mds-je510>

Manuel en français : <https://www.sony.fr/electronics/support/audio-video-audio-components/mds-je510/manuals>

Page 22 du manuel : «La fonction d'espacement automatique sert à insérer un espace vierge de 3 secondes entre les plages pendant la copie d'un MD, ce qui vous permet d'utiliser la fonction AMS lors de la lecture. Appuyez de façon répétée sur A.SPACE jusqu'à ce que "A.SPACE" apparaisse dans l'afficheur.»

**Problème : pour activer cette fonction, il faut utiliser la télécommande dédiée RM-D7M qui n'existe plus...**

## Plusieurs solutions

1. utiliser une application android de télécommande universelle, mais il faut un téléphone qui fait de la communication infrarouge («\*IR blaster\*»), ce n'est pas si commun dans les smartphones
2. faire un montage avec arduino et une led IR et envoyer le bon code...
3. programmer une télécommande universelle avec les bons codes
4. utiliser le flipper zero qui est doté d'un diode infrarouge

Je laisse de côté d'autres possibilités plus exotiques, comme cette application dédiée au minidisc sur Palm OS :

<https://palmdb.net/app/md-titler> ou la possibilité d'utiliser la sortie jack audio pour brancher un petit émetteur IR depuis un tel android (palme d'or de la bidouille pour cette solution!) : <https://irplus-remote.github.io/#audio>

Pas de télécommande universelle sous la main, ni de téléphone android avec led IR. La méthode avec arduino est sûrement intéressante mais peut-être un peu longue à mettre en place. En revanche, on a un flipper zero qui dort sagement.

Liens utiles pour les méthodes abandonnées:

Liste de téléphones android avec IR : [https://en.wikipedia.org/wiki/Category:Mobile\\_phones\\_with\\_infrared\\_transmitter](https://en.wikipedia.org/wiki/Category:Mobile_phones_with_infrared_transmitter)

Application android IRplus (selon un message lu sur un forum, les codes de la télécommande RM-D7M sont disponibles avec cette application) : <https://irplus-remote.github.io/>

circuit arduino : <https://www.instructables.com/How-to-Emulate-a-TV-Remote-or-Else-With-Arduino-Ir/>

bibliothèque IR pour arduino : <https://github.com/Arduino-IRremote/Arduino-IRremote>

Codes IR pour commander la platine MDS-JE510 : [https://www.minidisc.org/je500\\_codes.html](https://www.minidisc.org/je500_codes.html)

## Méthode utilisée avec Flipper Zero

On accède aux fonctions infrarouge du flipper depuis le menu principal, il est fourni avec des codes de télécommande universels pour le contrôle d'appareils comme TV, lecteurs audio et climatiseurs! Par exemple, pour allumer une télévision, il enverra tous les codes d'allumage selon les différents protocoles, fabricants, etc.

- <https://docs.flipper.net/infrared>
- <https://github.com/jamisonderek/flipper-zero-tutorials/wiki/Infrared>

Pour commander notre appareil, il va falloir envoyer des codes adaptés à placer sur la carte microSD.

1. Démonter la carte microSD : <https://docs.flipper.net/basics/sd-card#g4-removing-the-sd-card>
2. Cloner ce dépôt de codes IR : <https://github.com/UberGuidoZ/Flipper-IRDB>
3. Copier le code IR adapté (Sony RM-D10E) dans le répertoire infrared du flipper
4. Éjecter la carte microSD et la replacer dans le Flipper (settings/storage/unmount SDcard)
5. Dans le menu du flipper, retourner dans infrared/saved remotes et choisir le fichier Sony RM-D10E
6. Tester, ça fonctionne, youpi

Ça fonctionne, la platine MD réagit mais il n'y a pas le code A.SPACE ... (sans surprise, ce n'est pas le même modèle de télécommande).

Mais on a une bonne base pour ajouter des codes, puisque la communication se fait bien

Le fichier de configuration, pour Flipper, de la télécommande RM-D10E ressemble à ça :

[https://github.com/Lucaslhm/Flipper-IRDB/blob/main/MiniDisc/Sony\\_RM-D10E.ir](https://github.com/Lucaslhm/Flipper-IRDB/blob/main/MiniDisc/Sony_RM-D10E.ir)

Et les codes pour commander la platine MDS-JE510 : [https://www.minidisc.org/je500\\_codes.html](https://www.minidisc.org/je500_codes.html)

Si on prend l'exemple du code pour la fonction «repeat» Dans Flipper :

```
name: Repeat
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 26 00 00 00
```

Dans le fichier je500\_codes :

```
#07A6 - Repeat
...
#07B3 - A.Space
```

Dans ce fichier l'adresse correspondant à la commande 0 est #0780, c'est probablement l'adresse de départ, en hexadécimal A6-80 = 26 qui correspond à la commande du fichier flipper. Donc on peut supposer que le code pour A.SPACE sera (en hex) B3 - 80 = 33, j'essaie donc un bloc supplémentaire dans un nouveau fichier .ir

```
name: A.Space
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 33 00 00 00
```

Ça fonctionne bien, on peut envoyer la commande A.SPACE à la platine !

Ce serait possible d'aller plus loin pour intégrer les codes IR complets *mais flemme*

Le fichier modifié :

### **Sony\_RM-D7M.ir (cliquer pour afficher le code)**

```
Filetype: IR signals file
Version: 1
#
# Used with Sony MDS-JE480 MiniDisc Deck
#
name: POWER
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 15 00 00 00
#
name: Eject
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 16 00 00 00
#
name: Play
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 2A 00 00 00
#
name: Pause
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 29 00 00 00
#
name: Prev
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 20 00 00 00
#
name: Next
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 21 00 00 00
#
```

```
name: Rew
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 2B 00 00 00
#
name: FF
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 2C 00 00 00
#
name: Record
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 2D 00 00 00
#
name: LevelUp
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 10 00 00 00
#
name: LevelDn
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 11 00 00 00
#
name: Stop
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 28 00 00 00
#
name: 1
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 00 00 00 00
#
name: 2
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 01 00 00 00
#
name: 3
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 02 00 00 00
#
name: 4
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 03 00 00 00
#
name: 5
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 04 00 00 00
#
name: 6
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 05 00 00 00
#
name: 7
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 06 00 00 00
#
name: 8
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 07 00 00 00
#
name: 9
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 08 00 00 00
#
name: 10_0
type: parsed
protocol: SIRC
address: 0F 00 00 00
```

```
command: 09 00 00 00
#
name: Gt10
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 0A 00 00 00
#
name: TRec
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 30 00 00 00
#
name: Clear
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 0F 00 00 00
#
name: Input
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 66 00 00 00
#
name: Rec_Mode
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 60 00 00 00
#
name: Play_Mode
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 17 00 00 00
#
name: Display
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 18 00 00 00
#
name: Scroll
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 19 00 00 00
#
name: Fader
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 69 00 00 00
#
name: Menu_No
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 6A 00 00 00
#
name: Yes
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 6B 00 00 00
#
name: NameEdit_Sel
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 61 00 00 00
#
name: CDSync_Stop
type: parsed
protocol: SIRC
address: 11 00 00 00
command: 38 00 00 00
#
name: CDSync_Start
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 2A 00 00 00
#
name: CDSync_Stby
type: parsed
protocol: SIRC
address: 11 00 00 00
command: 32 00 00 00
#
name: Grp_OnOff
type: parsed
```

```
protocol: SIRC
address: 0F 00 00 00
command: 73 00 00 00
#
name: Grp_Skip
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 72 00 00 00
#
name: MusicSync
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 3F 00 00 00
#
name: Repeat
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 26 00 00 00
#
name: A_B
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 27 00 00 00
#
name: A.Space
type: parsed
protocol: SIRC
address: 0F 00 00 00
command: 33 00 00 00
```

Article extrait de : <http://www.lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse :

[http://www.lesporteslogiques.net/wiki/ressource/electronique/flipper\\_zero/reproduire\\_telecommande\\_infrarouge?rev=1751296938](http://www.lesporteslogiques.net/wiki/ressource/electronique/flipper_zero/reproduire_telecommande_infrarouge?rev=1751296938)

Article mis à jour: **2025/06/30 17:22**