

# Télécommande InfraRouge

Codage RC-5

- <http://en.wikipedia.org/wiki/RC-5>
- <http://www.positron-libre.com/electronique/protocole/code-rc5/code-rc5.php>
- <http://www.oumnad.123.fr/Telecommande-IR/RC5.htm>
- <http://scan78.free.fr/Elektor/Elektor%202001%20FR/f013024.PDF>
- [http://www.pcbheaven.com/userpages/The\\_Philips\\_RC5\\_Protocol/](http://www.pcbheaven.com/userpages/The_Philips_RC5_Protocol/)
- [http://www.sonelec-musique.com/electronique\\_realisations\\_alim\\_led.html](http://www.sonelec-musique.com/electronique_realisations_alim_led.html)

## Histoire

C'est en 1956 qu'apparaît la première télécommande à ultrason pour une télévision de Zenith Electronics.

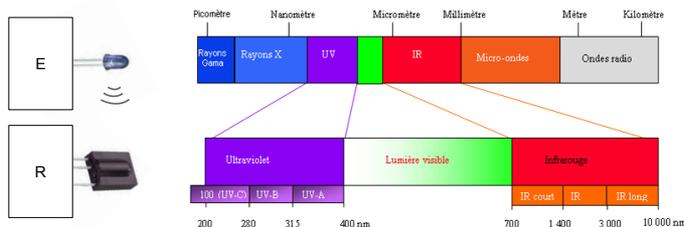
La télécommande à infrarouge apparaît dans les années 1980

Philips développe le protocole RC-5 à la fin des années 1980 pour la commande des appareils électroniques grand public avec le souhait d'une interopérabilité entre les télécommandes et les appareils.



## Principe

Une diode infrarouge produit une lumière de longueur d'onde de 700 à 950nm.  
Un récepteur capte la lumière.



L'information est codée dans une séquence d'allumage et d'extinction (ON-OFF) de la diode infrarouge.

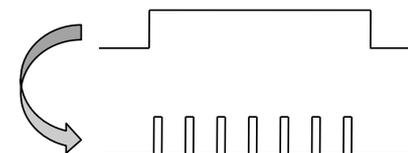
## Modulation de l'infrarouge

La chaleur produit aussi un rayonnement infrarouge, il faut :

⇒ que le récepteur ne voit que le rayonnement de la diode,

⇒ "allumer la diode" signifie effectuer une séquence de ON-OFF à une fréquence de l'ordre de 38kHz, on parle de modulation

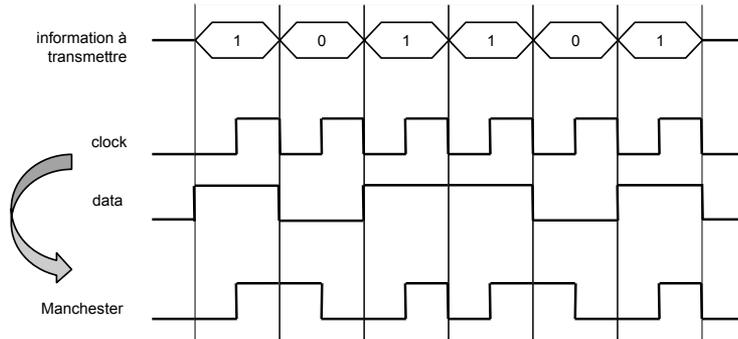
Le récepteur est muni d'un filtre qui ne voit que l'IR modulé à 38kHz.



La fréquence de modulation varie selon les protocoles pour RC-5 c'est 36kHz

## Codage Manchester

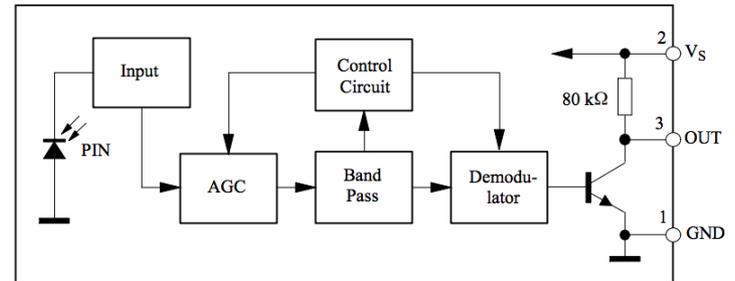
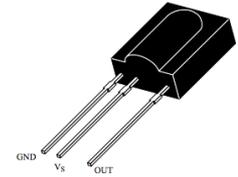
Le signal émit doit contenir la donnée ET l'horloge,  
 ⇒ c'est-à-dire le moyen de distinguer les début et fin de bit



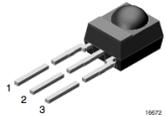
## Block Diagram du récepteur TSOP1736

TSOP1736

- La porteuse à 36kHz est filtrée
- La sortie du signal est inversé
- vs entre 4.5V et 5.5V



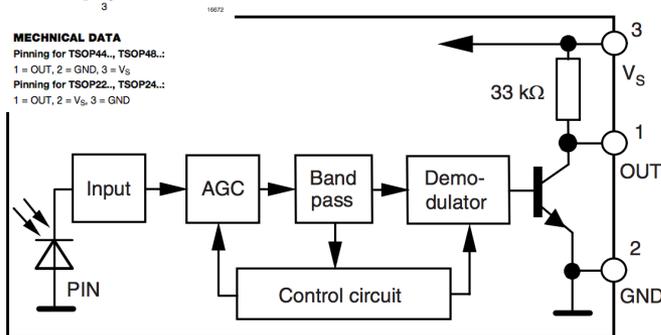
## Block Diagram du récepteur TSOP4838



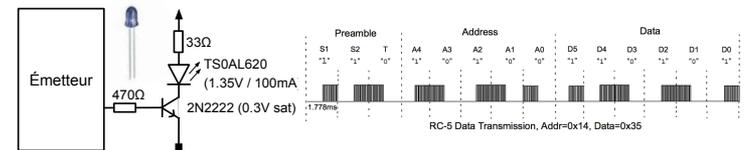
TSOP4838

- Comportement identique au TSOP1736
- VS de 2.5V à 5.5V

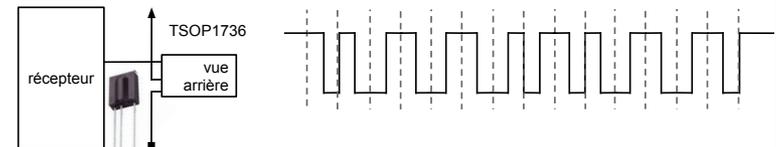
**MECHANICAL DATA**  
 Pinning for TSOP44., TSOP48.:  
 1 = OUT, 2 = GND, 3 =  $V_S$   
 Pinning for TSOP22., TSOP24.:  
 1 = OUT, 2 =  $V_S$ , 3 = GND



## Emetteur - Récepteur RC-5



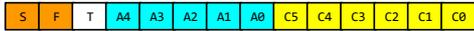
Le Pic reçoit le signal inversé



# Codage RC-5

Une trame RC-5 est une suite de 14 bits

- 2 bits de départ Start(1)+Field(1|0)
- 1 bit de basculement
- 5 bits d'adressage
- 6 bits de commandes



Les 2 bits de départ servent à calibrer le gain du circuit intégré de réception.

Le bit de basculement indique une nouvelle transmission de données.

⇒ Sa valeur change à chaque nouvel appui d'une touche pour distinguer une nouvelle pression d'une pression continue sur la même touche.

Les 5 bits suivants déterminent l'adresse du dispositif commandé.

⇒  $2^5 = 32$  groupes d'adressage.

La commande destinée à l'appareil est codée dans les 6 derniers bits.

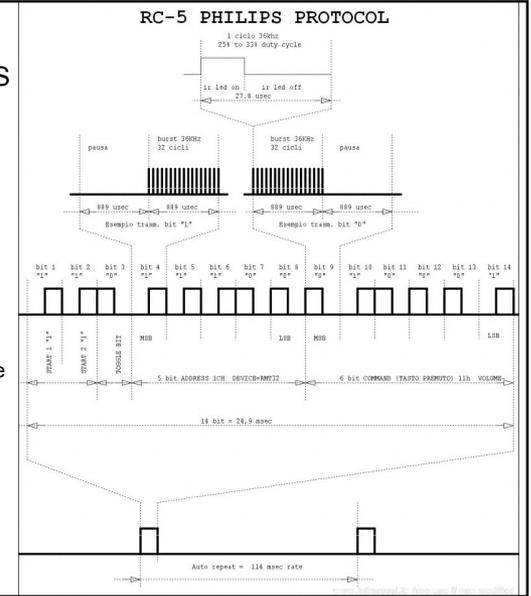
⇒  $2^6 = 64$  commandes.

⇒ Version étendue le bit F (Field) code le bit n°6 de la commande  $2^{1+6} = 128$

Le document PHILIPS officiel de décembre 1992

qui résume :

- la modulation de l'IR
- le codage Manchester
- la trame RC-5
- la répétition automatique



# Codes RC-5

Les codes sont standards afin de permettre l'interopérabilité.

Ce n'est toutefois pas toujours respecté.

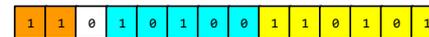
RC-5 Address	Device
\$00 - 0	TV1
\$01 - 1	TV2
\$02 - 2	Teletext
\$03 - 3	Video
\$04 - 4	LV1
\$05 - 5	VCR1
\$06 - 6	VCR2
\$07 - 7	Experimental
\$08 - 8	Sat1
\$09 - 9	Camera
\$0A - 10	Sat2
\$0B - 11	Standby
\$0C - 12	CDV
\$0D - 13	Camcorder
\$0E - 14	
\$0F - 15	
\$10 - 16	Pre-amp
\$11 - 17	Tuner
\$12 - 18	Recorder1
\$13 - 19	Pre-amp
\$14 - 20	CD Player
\$15 - 21	Phono
\$16 - 22	SatA
\$17 - 23	Recorder2
\$18 - 24	
\$19 - 25	
\$1A - 26	CDR
\$1B - 27	
\$1C - 28	
\$1D - 29	Lighting
\$1E - 30	Lighting
\$1F - 31	Phone

RC-5 Command	TV Command	VCR Command
\$00 - 0	0	0
\$01 - 1	1	1
\$02 - 2	2	2
\$03 - 3	3	3
\$04 - 4	4	4
\$05 - 5	5	5
\$06 - 6	6	6
\$07 - 7	7	7
\$08 - 8	8	8
\$09 - 9	9	9
\$0A - 10	-/-	-/-
\$0B - 11	Standby	Standby
\$0C - 12	Mute	
\$10 - 16	Volume +	
\$11 - 17	Volume -	
\$12 - 18	Brightness +	
\$13 - 19	Brightness -	
\$20 - 32	Program +	Program +
\$21 - 33	Program -	Program -
\$32 - 50		Fast Rewind
\$34 - 52		Fast Forward
\$35 - 53		Play
\$36 - 54		Stop
\$37 - 55		Recording

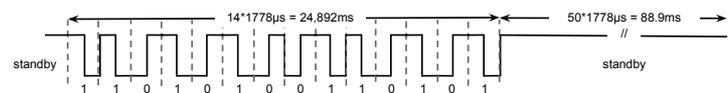
# Exemple

Emission

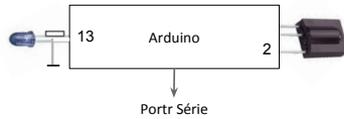
adresse = 0x35, code = 0x34



Réception



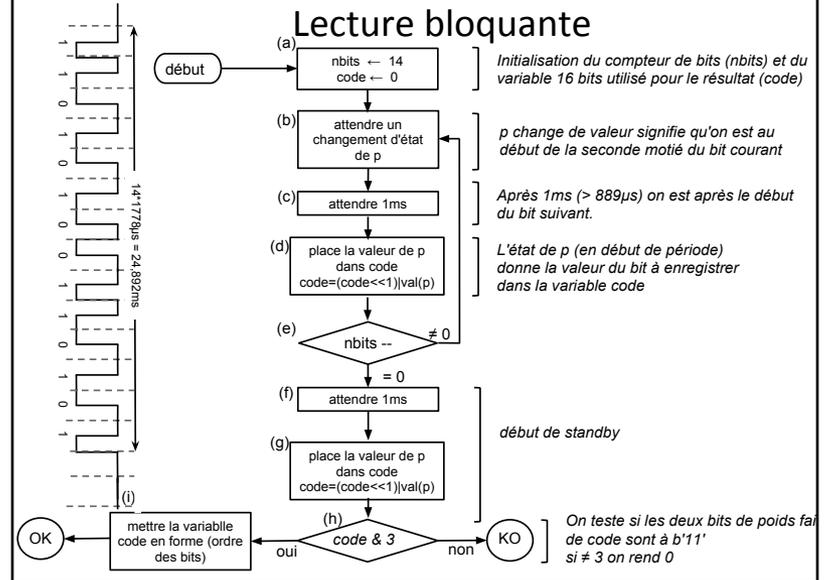
## Lecture bloquante



```
int readRC5(byte p)
```

```
MSB : 1 1 T A4 A3 A2 A1 A0
LSB : 1 F C5 C4 C3 C2 C1 C0
```

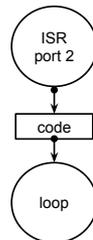
## Lecture bloquante



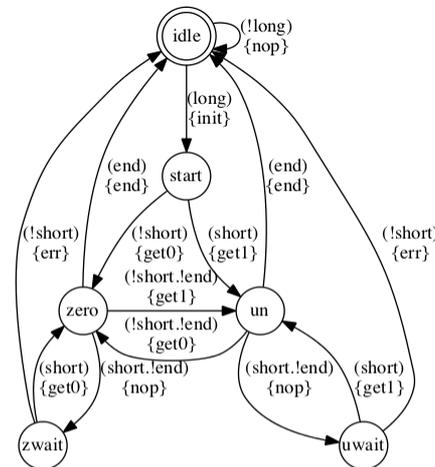
## Lecture non bloquante

Le but est de pas bloquer l'exécution de loop pendant la lecture du capteur

- Le capteur IR est connecté sur une entrée d'interruption
- L'ISR associée va être déclenchée à chaque changement d'état
- et décoder la trame pour l'écrire dans **code**



## Automate de lecture



INPUTS  
 long : ((time-begin) > 10ms)  
 short : ((time-begin) < 1ms)  
 end : (nbit==14)

ACTIONS  
 init : {begin=time; nbit=0; val=0;}  
 get0 : {begin=time; nbit++; val = val<<1;}  
 get1 : {begin=time; nbit++; val = val<<1;}  
 nop : {begin=time;}  
 end : {res=val; flagres=1;}