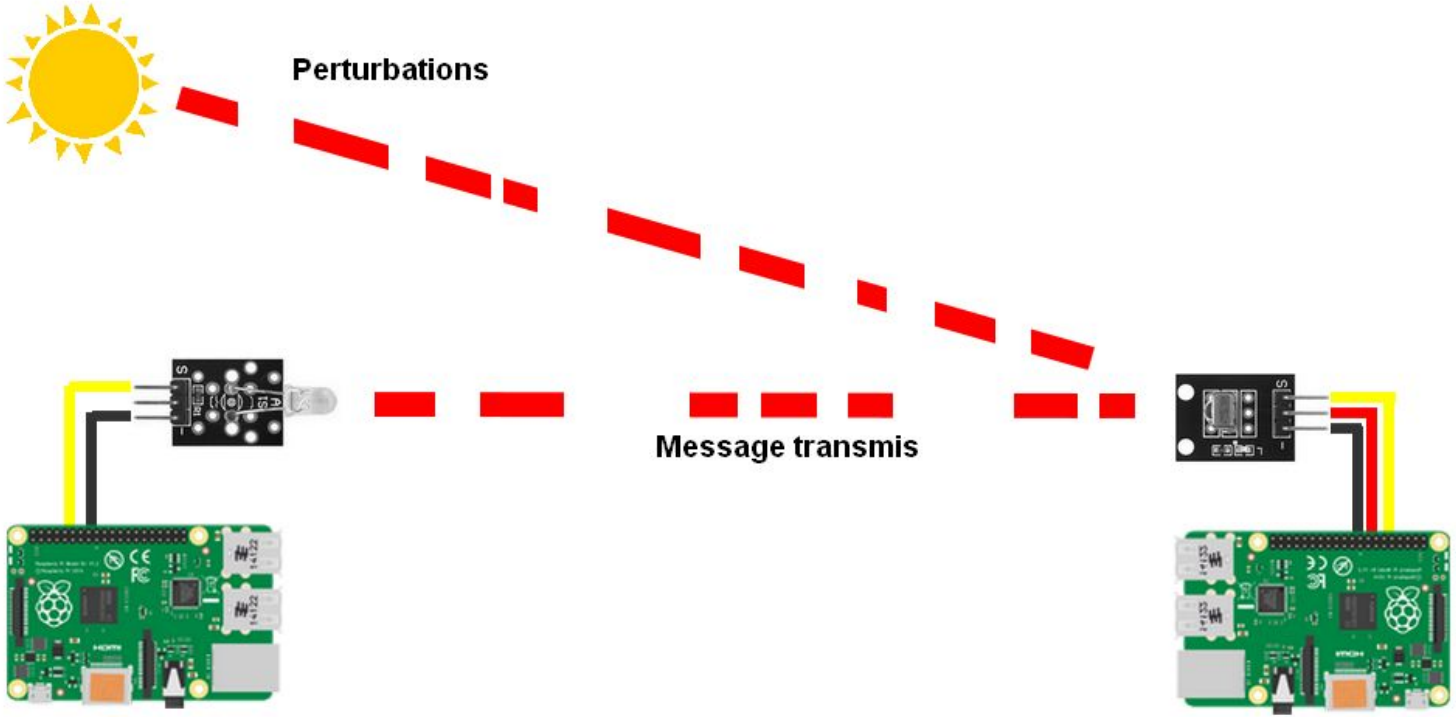
# Séance n°5 : Prog réseau

Alexis DELPLACE

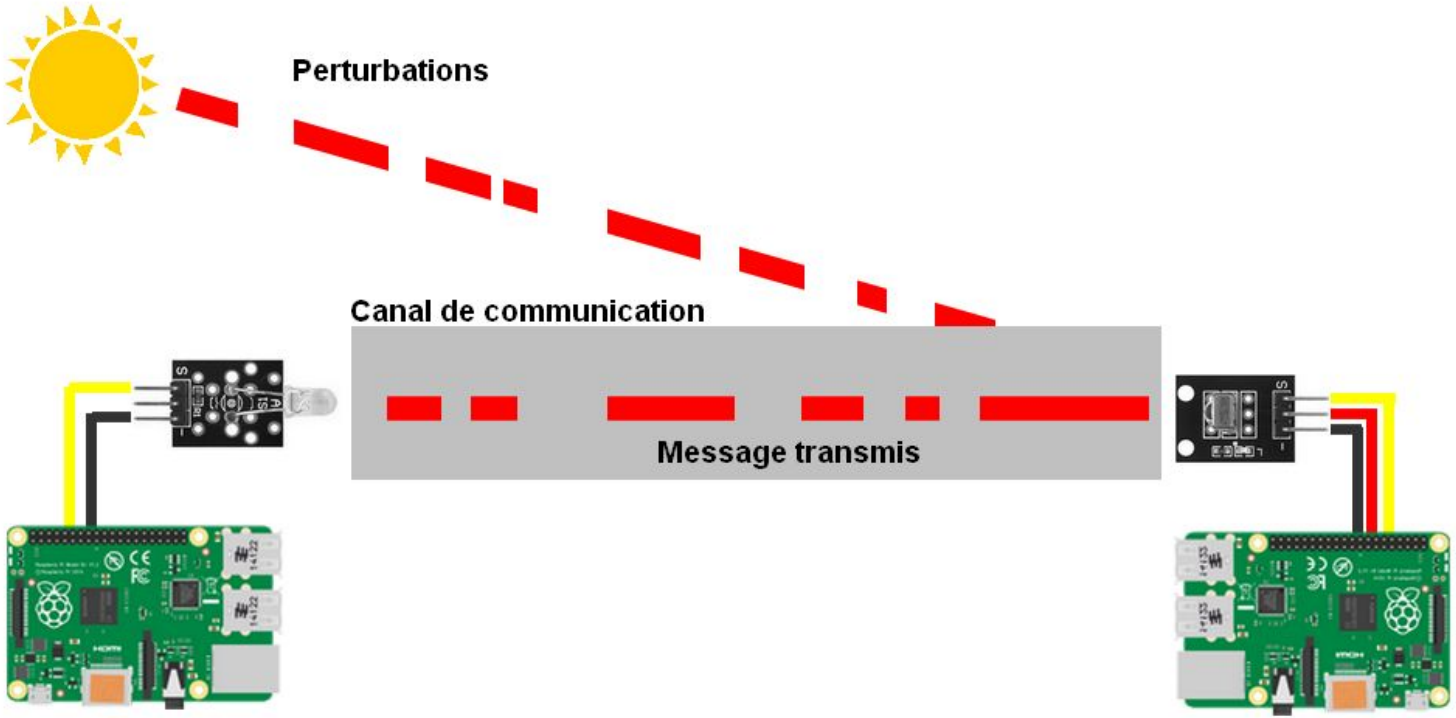
# Programme

- Communication infrarouge
- Module suiveur de ligne

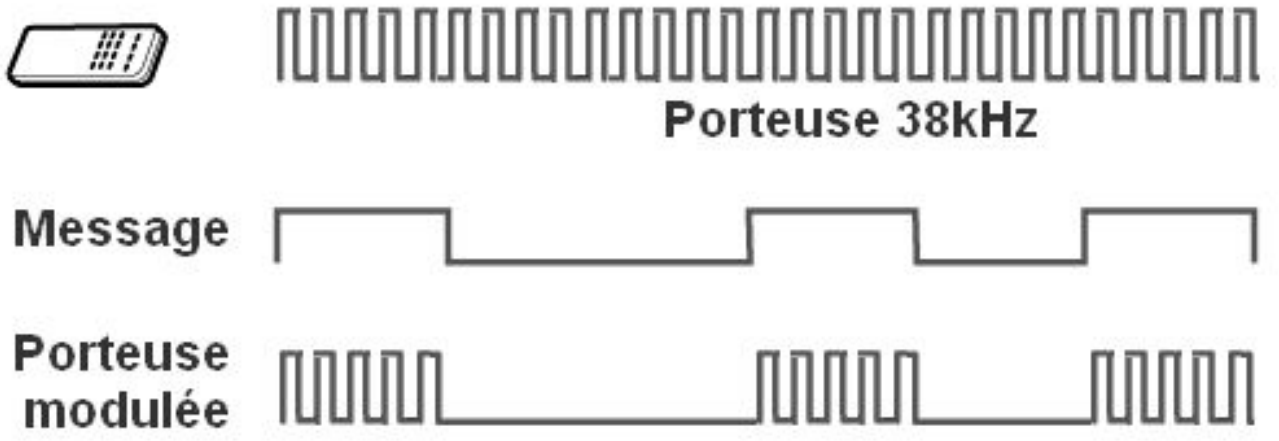
# Communication infrarouge



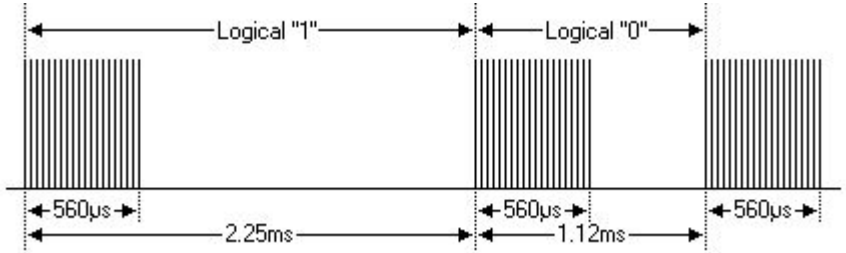
# Communication infrarouge



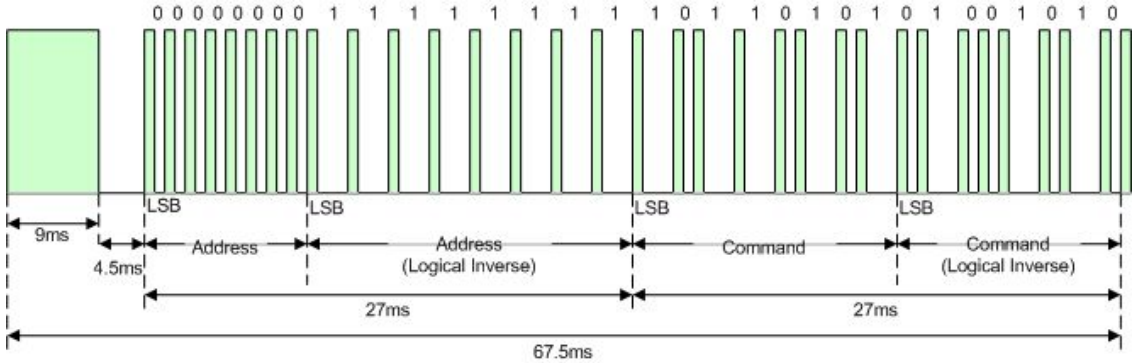
# La porteuse



# Le protocole NEC




- Pour envoyer un 1 logique, il faut “allumer” la LED infrarouge pendant 562.5µs puis l'éteindre pour une durée de 1687.5µs.
- Pour envoyer un 0 logique, il faut “allumer” la LED infrarouge pendant 562.5µs puis l'éteindre pour une durée de 562.5µs



# Implémentation (émission d'un tir)

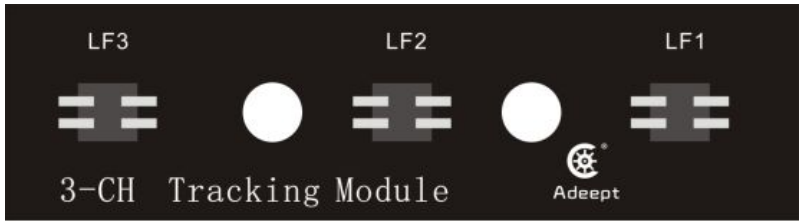
```
import InfraLib
InfraLib.IRBlast(tankID, "LASER")
```

```
def IRBlast(tankID, projectile_type, verbose=False):
        if projectile_type == "LASER":
                projectile_id = 0xF1
        else:
                if verbose:
                        print("unknown projectile type")
                return False

        msg = (str(bin(projectile_id))[2:] + str(bin(tankID))[2:])
        if verbose:
                print("send :", msg)
        IR(23, "NEC", dict()).send_code(encodeMsg(msg)+"0")
        # Last bit not receive so we add an artificial one for the actual last bit to be received
        return True
```

# Implémentation (réception d'un tir)

```
import InfraLib

# IR Receiver
IR_RECEIVER = 15
GPIO.setup(IR_RECEIVER, GPIO.IN)
GPIO.add_event_detect(IR_RECEIVER, GPIO.FALLING, callback=lambda x: InfraLib.getSignal(IR_RECEIVER, client), bouncetime=100)
```

# Module suiveur de ligne




```
pi@raspberrypi:~ $ sudo python3 adeept_trackingmodule/trackingmodule.py
LF3: 0   LF2: 0   LF1: 0

LF3: 0   LF2: 0   LF1: 0

LF3: 0   LF2: 0   LF1: 0

LF3: 0   LF2: 0   LF1: 0

LF3: 0   LF2: 0   LF1: 0

LF3: 0   LF2: 0   LF1: 0
```

# Implémentation

```
# Tracking Module
LINE_PIN_MIDDLE = 36
GPIO.setup(LINE_PIN_MIDDLE,GPIO.IN)
GPIO.add_event_detect(LINE_PIN_MIDDLE, GPIO.BOTH, callback=enterFlagArea, bouncetime=100)


def enterFlagArea(channel):
    """
    Send flag zone related data to the server
    """
    if GPIO.input(LINE_PIN_MIDDLE) == GPIO.LOW:
        client.publish("tanks/"+hex(tankID)+"/flag","ENTER_FLAG_AREA")
    else:
        client.publish("tanks/"+hex(tankID)+"/flag", "EXIT_FLAG_AREA")
```