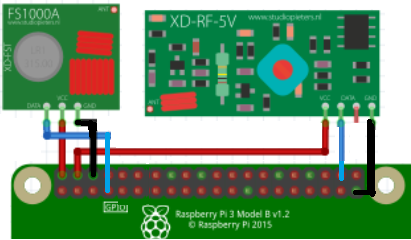
<https://www.instructables.com/RF-433-MHZ-Raspberry-Pi/>

### **[RF 433 MHZ (Raspberry Pi) : 4 Steps - Instructables](https://www.instructables.com/RF-433-MHZ-Raspberry-Pi/)**

Transmitter Receiver





**apt-get install python3-pip**

**sudo pip3 install rp-rf**

python3 send.py -p 174 -t 1 5330691

python3 send.py 12345

send.py

import argparse

import logging

from rpi\_rf import RFDevice

# Para utilizar logging.info

logging.basicConfig(level=logging.INFO, datefmt='%Y-%m-%d %H:%M:%S',

                    format='%(asctime)-15s - [%(levelname)s] %(module)s: %(message)s',)

parser = argparse.ArgumentParser(description='Sends a decimal 433/315MHz GPIO device')

parser.add\_argument('code', metavar='CODE', type=**int**,help="Decimal code to send")

parser.add\_argument('-g', dest='gpio', type=**int**, **default**=04, help="GPIO 04)")

parser.add\_argument('-p', dest='pulselength', type=**int**, **default**=None,

                    help="Pulselength (Default: 350)")

parser.add\_argument('-t', dest='protocol', type=**int**, **default**=None, help="Protocol (Def: 1)")

args = parser.parse\_args()

rfdevice = RFDevice(args.gpio)

rfdevice.enable\_tx()

**if** args.protocol:

    protocol = args.protocol

**else**:

    protocol = "default"

**if** args.pulselength:

    pulselength = args.pulselength

**else**:

    pulselength = "default"

logging.info(str(args.code) +

             " [protocol: " + str(protocol) +

             ", pulselength: " + str(pulselength) + "]")

rfdevice.tx\_code(args.code, args.protocol, args.pulselength)

rfdevice.cleanup()

En otra ventana

python3 receive.py

receive.py

#!/usr/bin/env python3

import argparse

import signal

import sys

import time

import logging

from rpi\_rf import RFDevice

rfdevice = None

# pylint: disable=unused-argument

def exithandler(signal, frame):

rfdevice.cleanup()

sys.exit(0)

# Para utilizar logging.info

logging.basicConfig(level=logging.INFO, datefmt='%Y-%m-%d %H:%M:%S',

                    format='%(asctime)-15s - [%(levelname)s] %(module)s: %(message)s',)

parser = argparse.ArgumentParser(description='Receives a decimal 433/315MHz')

parser.add\_argument('-g', dest='gpio', type=int, default=20, help="GPIO 20")

args = parser.parse\_args()

signal.signal(signal.SIGINT, exithandler)

rfdevice = RFDevice(args.gpio)

rfdevice.enable\_rx()

timestamp = None

logging.info("Listening for codes on GPIO " + str(args.gpio))

while True:

if rfdevice.rx\_code\_timestamp != timestamp:

timestamp = rfdevice.rx\_code\_timestamp

# escribe en pantalla el codigo, el pulselength, el protocol

logging.info(str(rfdevice.rx\_code) +

" [pulselength " + str(rfdevice.rx\_pulselength) +

", protocol " + str(rfdevice.rx\_proto) + "]")

time.sleep(0.01)

rfdevice.cleanup()

CON LOGIC LEVEL CONVERTER

