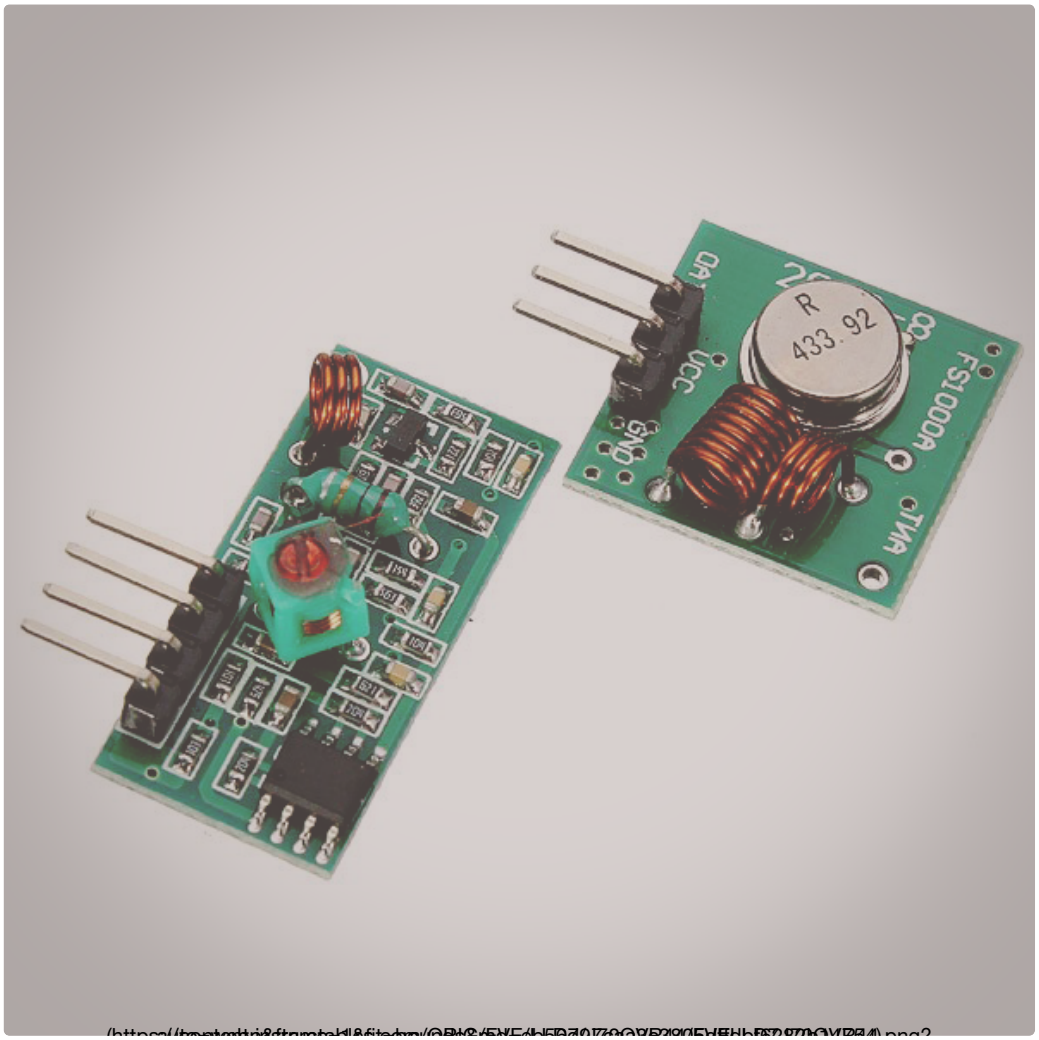
RF 433 MHZ (Raspberry Pi)

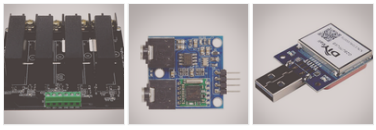
By piddlerintheroot (/member/piddlerintheroot/) in Circuits (/circuits/) > Raspberry Pi (/circuits/raspberry-pi/projects/) 51,142 34 2

Download Favorite



(/member/piddlerintheroot/)
By **piddlerintheroot (/member/piddlerintheroot/)**
Follow

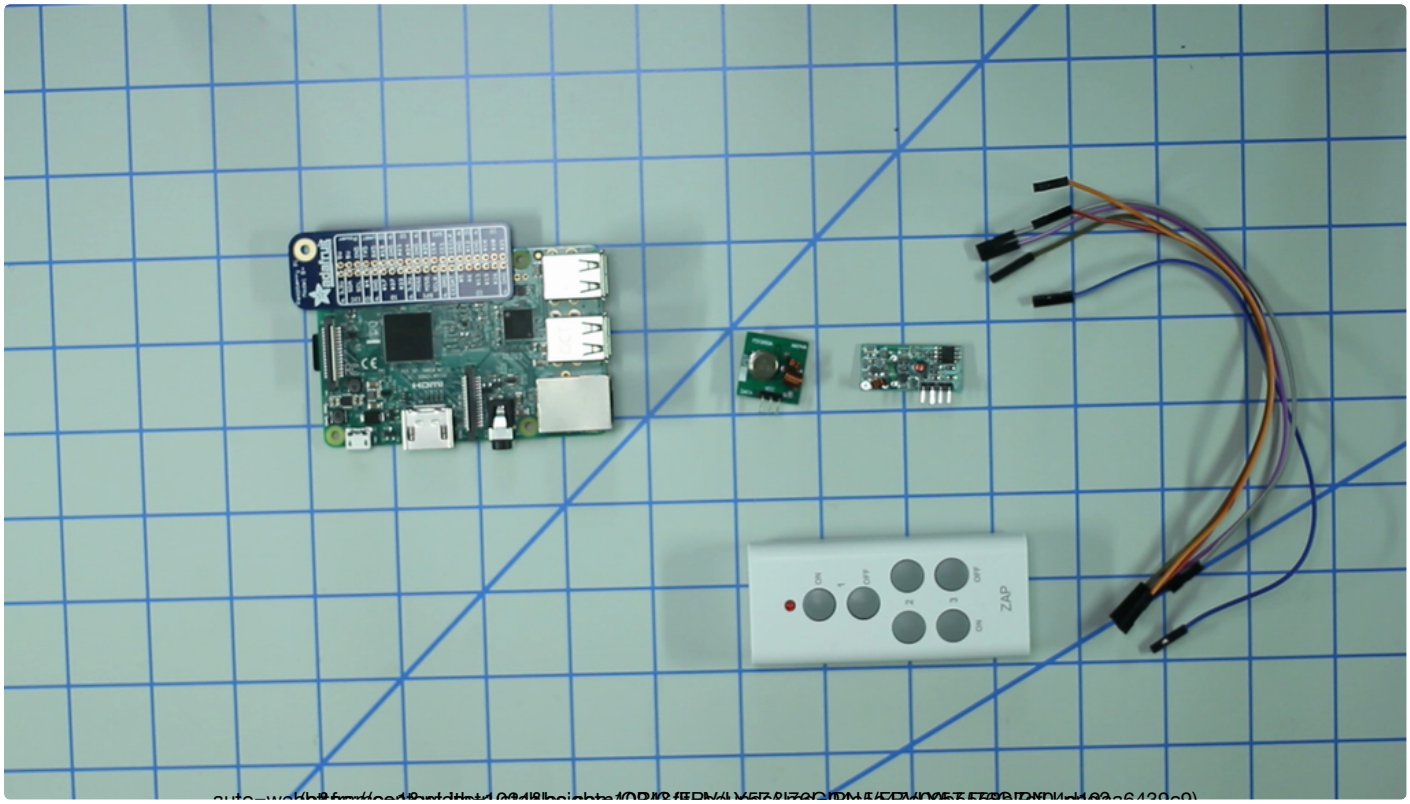
More by the author:



About: the Raspberry Pi is Awesome! More About piddlerintheroot » (/member/piddlerintheroot/)

Basic tutorial of how to setup a generic 433 MHZ transmitter/reciever with the Raspberry Pi.

Step 1: Parts



PARTS:

RPI 3 - https://amzn.to/2VA9pQY (https://amzn.to/2VA9pQY)

4 Amp Power Adapter - https://amzn.to/2CTptWu (https://amzn.to/2CTptWu)

16GB micro SD - https://amzn.to/2SFMwd3 (https://amzn.to/2SFMwd3)

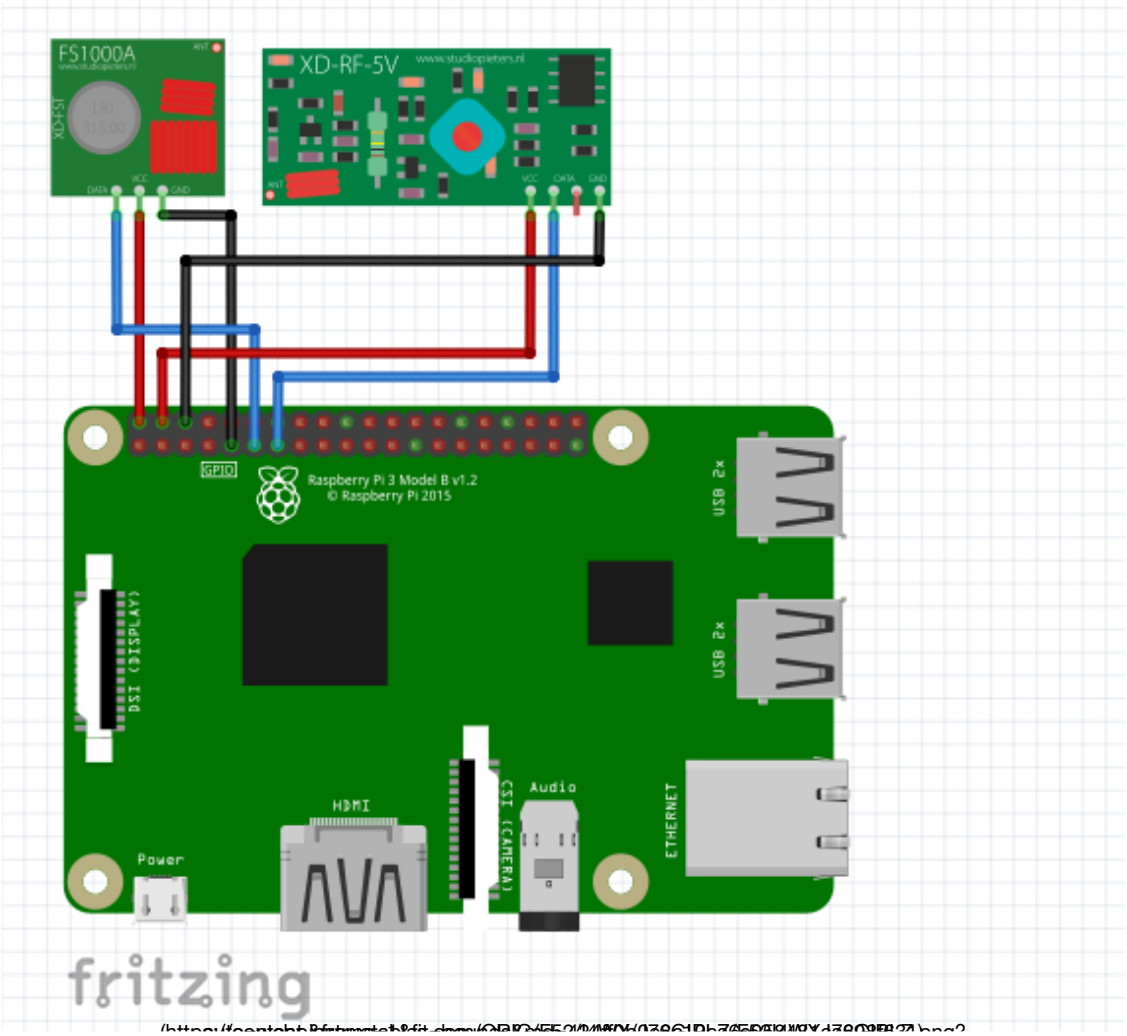
(https://amzn.to/2SFMwd3)120 pcs jumper cable: https://ebay.to/2VAb9cY (https://ebay.to/2VAb9cY)

433 MHZ TX/RX kit: https://amzn.to/2M9saGC (https://amzn.to/2M9saGC)

RF Outlet Set: https://amzn.to/2M91DJu (https://amzn.to/2M91DJu)

(https://goo.gl/PL4ryu)

Step 2: Setup



rpi-rf: https://pypi.python.org/pypi/rpi-rf (https://pypi.python.org/pypi/rpi-rf)

SSH into Raspberry Pi

1. "sudo apt-get install python3-pip"

2. "sudo pip3 install rpi-rf"

Step 3: Code

recieve.py

```python
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)

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 code via a 433/315MHz GPI
parser.add_argument('-g', dest='gpio', type=int, default=27,
                    help="GPIO pin (Default: 27)")
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
        logging.info(str(rfdevice.rx_code) +
                     " [pulselength " + str(rfdevice.rx_pulselength) +
                     ", protocol " + str(rfdevice.rx_proto) + "]")
    time.sleep(0.01)
rfdevice.cleanup()
```

*Note use python3

1. Run recieve.py and note code, pulselength, protocol

2. Run send.py with code, pulselength, and protocol arguments

recieve.py (https://content.instructables.com/ORIG/F3N/KBEX/J76GIPWM.py)

send.py (https://content.instructables.com/ORIG/F27/XCD5/J76GIPWN.py)

Add Tip Ask Question Comment Download

Step 4: Additional Info