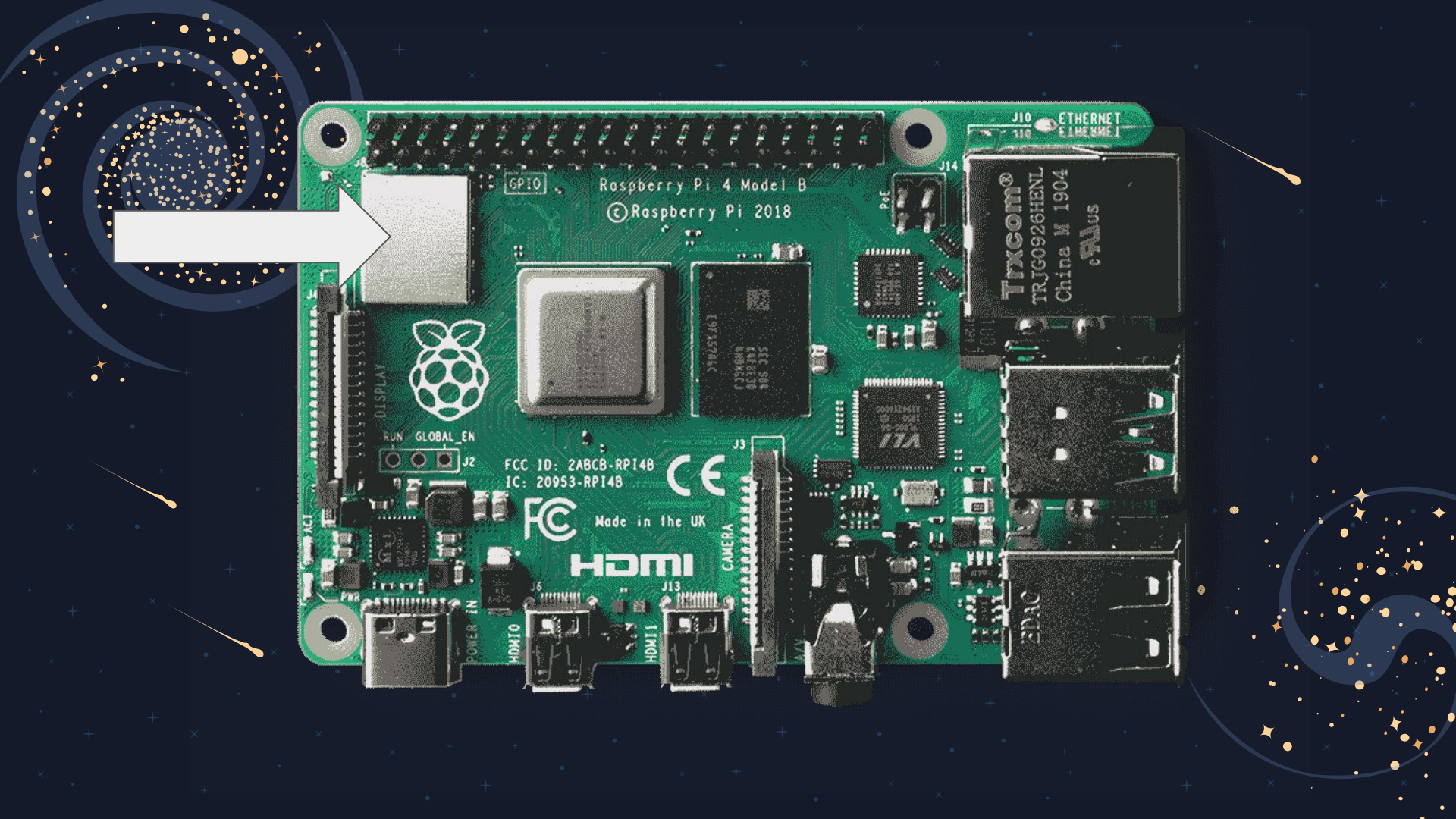
Communication via Bluetooth

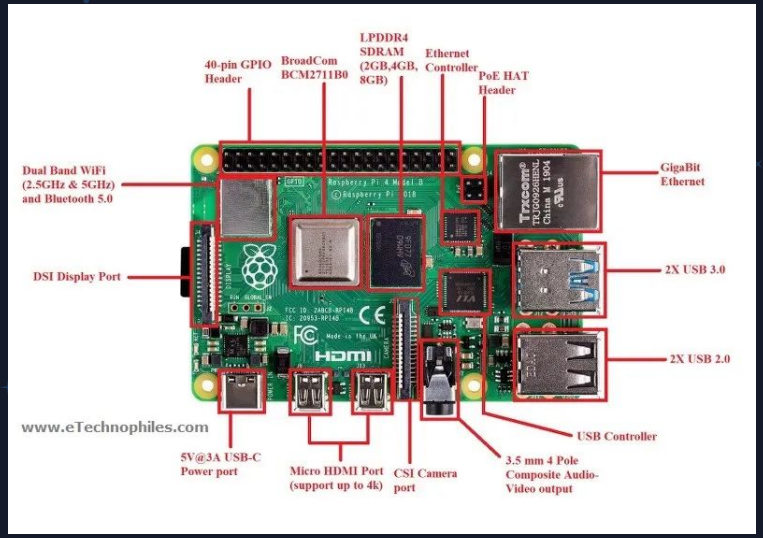
Ranvitha, Stephen, Jessie

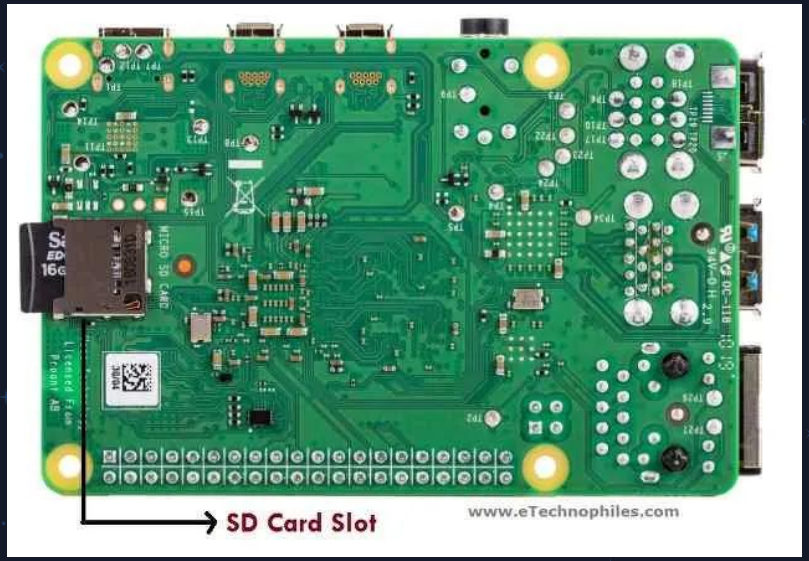
Mechanical Interface

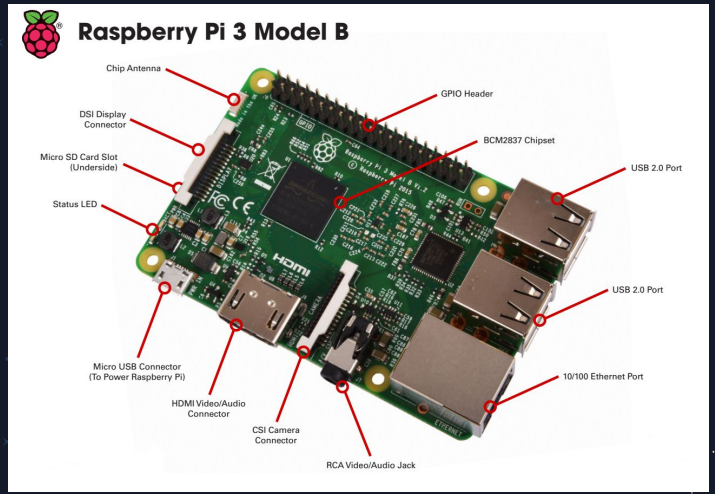
Built-in bluetooth:
Pi 4: HCI version 5.0
Pi 3: HCI version 4.1
Frequency 2.4GHz

The bluetooth antenna could be found at the top left hand corner of the board, next to the 40 GPIO pins.


GPIO
Raspberry Pi 4 Model B
©Raspberry Pi 2018
J10
ETHERNET
J14
POE
Trxcom®
TRJG0926HENL
China M 1904
DISPLAY
RUN
GLOBAL_EN
J2
J3
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
Made in the UK
HDMI
CAMERA
J6
J13
POWER IN
HDMI0
HDMI1
PWR
ACT
EDAC







Raspberry Pi4 and Raspberry Pi3 Electrical Interfaces

Critical Functions

1. **Connect to Bluetooth between 2 Pi's**

https://bluedot.readthedocs.io/en/latest/pairpipi.html

https://pybluez.readthedocs.io/en/latest/index.html

- Installing bluedot library:
    - → sudo pip3 install bluedot
    - → sudo pip3 install bluedot --update
- Installing pybluez library:
    - → pip install pybluez
- In Command Line in the 1st Pi:
    - → bluetoothctl
    - → at the [bluetooth]# prompt, input:
        - → power on
        - → discoverable on
        - → pairable on
        - → agent on
        - → default-agent

- Repeat steps with the 2nd Pi, but input "scan on" in addition at the [bluetooth]# prompt
- Check the 2nd Pi for a message that the 1st Pi has been found:

    [NEW] Device 12:23:34:45:56:67 devicename
- Type "pair" with MAC address of the 1st Pi:

    pair 12:23:34:45:56:67
- Type "connect" with MAC address of the 1st Pi
- Confirm authentication on both Pi's
- On both Pi's: confirm the code → type "quit" and press Enter to get back to cmd.

To take a look at the list of commands after entering "bluetoothctl": type "bluetoothctl help"

2. Measuring the RSSI value (Received Signal Strength Indicator, in dB) (manually through cmd):

- Once both of the Pi's are connected:
    → Exit "[bluetooth]#" by typing "quit" or "exit"
    → Type "hcitool rssi <MAC address of the other Pi>"

3. Measuring the transmit power (in dB mW) (manually through cmd):

- Once both of the Pi's are connected:
    → Exit "[bluetooth]#" by typing "quit" or "exit"
    → Type "hcitool tpl <MAC address of the other Pi>"

*To see a list of command of *hcitool*, type "hcitool" into cmd*

To see address of the current pi, type "hcitool dev" into cmd

Changing the name of your device

In cmd, type “sudo hciconfig hci0 name <Device’s new name>”
Then, type “hciconfig -a” to check out the name changes.

```
#automatic connection and disconnection
'''
reference: people.csail.mit.edu/albert/bluez-intro/c212.html
reference: pybluez.readthedocs.io/en/lastest/index/html
reference: bluedot.readthedocs.io/en/latest/inedex/html
description: these functions allow you to set your device discoverable, scanning for other devices,
automatically connect and disconnect with a device with a specified MAC address.
'''

import bluetooth
import socket
import subprocess


def __init__(device, other_device):
    assert other_device is not socket.gethostbyname()
    device.other_device = other_device
    device.other_address = None
    device.send_sock = None

def scan_for_devices(): #scan for bluetooth devices
    client_address = None
    client_name = None
    bluetooth_devices = bluetooth.discover_devices()
    print("Searching for bluetooth devices...")

    for bdaddr in bluetooth_devices:
        if client_name == bluetooth.lookup_name( bdaddr ):
            client_address = bdaddr
            break

    if client_address is not None:
        print ("found" + client_name + "with MAC address" + client_address)
    else:
        print ("no device found")

def connect_as_host(device, port): #connect the device as a host
    subprocess.call(['sudo', 'hciconfig', 'hci0', 'piscan'])
    print (device + "is now discoverable")

    device.sever_sock_receive=bluetooth.BluetoothSocket( bluetooth.FRCOMM )
    device.sever_sock_receive.bind(("", port))
    device.sever_sock_receive.listen(port)

    device.client_sock_receive, device.other_address = device.server_sock.accept()
    print("Connected to" + device.other_address)
```

Stretch Goal 1

```
def connect_as_client(device, port): #connect the device as a client
    bluetooth_devices = bluetooth.discover_devices()
    print("Searching for blueetooth devices...")

    for address in bluetooth_devices:
        print(bluetooth.lookup_name(address) + "with address" + address)
        if device.other_device == bluetooth.lookup_name(address):
            device.other_address = address
            break

    if device.other_address is None:
        print("No device found with name" + device.other_device)
        return False
    else:
        print("Found" + device.other_device + "with address" + device.other_address)

    device.send_sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
    device.send_sock.connect((device.other_address, port))
    return True

def host_close(device):
    assert device.client_sock_receive is not None and device.server_sock_receive is not None
    device.client_sock_receive.close()
    device.server_sock_receive.close()

def client_close(device):
    assert device.send_sock is not None
    device.send_sock.close()

```

Stretch Goal 2 and 3

```python
# https://bluedot.readthedocs.io/en/latest/pairpipi.html - shows how to pair two raspberry pis (taken 2 minutes max to do)
#sudo pip3 install bluedot - run on command line

#for raspberry pi that will be you will input the message/data
from bluedot.btcomm import BluetoothServer
from time import sleep
from signal import pause

message = input("What encryted message do you want to send?\n")
data =  ''
i = len(message)  -  1
while  i >=  0 :
    data = data + message[i]
    i  =  i  =  1

def data_received(data):
    print("recv - {}".format(data))
    server.send(data)

def client_connected():
    print("client connected")

def client_disconnected():
    print("client disconnected")

print("init")
server = BluetoothServer(
    data_received,
    auto_start = False,
    when_client_connects = client_connected,
    when_client_disconnects = client_disconnected)

print("starting")
server.start()
print(server.server_address)
print("waiting for connection")

try:
    pause()
except KeyboardInterrupt as e:
    print("cancelled by user")
finally:
    print("stopping")
    server.stop()
print("stopped")



#for raspberrypi that will receive message
from bluedot.btcomm import BluetoothClient
from datetime import datetime
from time import sleep
from signal import pause

def data_received(data):
    print("recv - {}".format(data))

print("Connecting")
c = BluetoothClient("pi4", data_received)

print("Sending")
try:
    while True:
        c.send("hi {} \n".format(str(datetime.now())))
        sleep(1)
finally:
    c.disconnect()
```