There are several ways to connect multiple Arduinos to a Raspberry Pi to read data from them. Here are two common methods:

1. **Using the Serial Interface:**

One way to connect multiple Arduinos to a Raspberry Pi is to use the serial interface. You can connect each Arduino to the Raspberry Pi using a USB-to-serial converter, such as the FTDI or PL2303 module. You will need to install the required driver for each module.

***

To install the drivers for FTDI or PL2303 modules, follow these steps:

1. Plug in the module to a USB port on your computer.
2. Check the Device Manager to see if the driver has been installed automatically. Open Device Manager and look under the "Ports (COM & LPT)" section. If you see the name of your module (e.g. "USB Serial Port (COM3)"), then the driver has been installed automatically.
3. If the driver is not installed automatically, you will need to download and install the driver manually. You can download the driver from the manufacturer's website or from a third-party website. Be sure to download the driver that is compatible with your operating system.
4. After downloading the driver, run the installer and follow the on-screen instructions to install the driver.
5. Once the driver is installed, check the Device Manager again to ensure that the module is listed under "Ports (COM & LPT)".
6. You can now use the module to communicate with your microcontroller using serial communication. In the Arduino IDE, select the appropriate serial port from the Tools > Port menu to upload code to your microcontroller or read data from it.

***

Next, you will need to configure the serial communication between the Arduino and Raspberry Pi. You can use the Serial library in the Arduino IDE to send data to the Raspberry Pi over serial communication.

***

To configure serial communication between an Arduino and a Raspberry Pi, you will need to specify the serial communication parameters on both the Arduino and Raspberry Pi sides. Here are the basic steps:

1. On the Arduino side, open the Arduino IDE and select the appropriate board and serial port from the Tools menu.

2. Use the Serial.begin() function in your Arduino code to set the baud rate of the serial communication. For example, to set the baud rate to 9600, use the following line of code:
   Serial.begin(9600);

3. In your Arduino code, use the Serial.print() or Serial.write() functions to send data over the serial port. For example, to send the value of a variable named "data" over the serial port, use the following line of code:
   Serial.print(data);

4. On the Raspberry Pi side, open a terminal window and enter the following command to install the PySerial library:
   sudo apt-get install python3-serial

5. In your Python script on the Raspberry Pi, import the serial module and create a new Serial object to communicate with the Arduino. For example, to create a new Serial object with a baud rate of 9600 and a timeout of 1 second, use the following code:

```
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
```

(Note that you will need to replace "/dev/ttyUSB0" with the appropriate serial port name for your Arduino.)

6. In your Python script, use the Serial.read() or Serial.readline() functions to read data from the serial port. For example, to read a line of data from the Arduino and store it in a variable named "data", use the following code:

```
data = ser.readline().decode().strip()
```

The decode() function is used to convert the received bytes into a string, and the strip() function removes any newline or carriage return characters that may be included.

***

On the Raspberry Pi side, you can use a Python script to read the serial data from each Arduino. You can use the PySerial library to communicate with the Arduinos. Here is an example code snippet:

```
import serial
# open the serial port for each Arduino
ser1 = serial.Serial('/dev/ttyUSB0', 9600)
ser2 = serial.Serial('/dev/ttyUSB1', 9600)
while True:
    # read the serial data from each Arduino
    data1 = ser1.readline().decode().strip()
    data2 = ser2.readline().decode().strip()
    # do something with the data from each Arduino
    print(f"Arduino 1: {data1}")
    print(f"Arduino 2: {data2}")
```

2. **Using the I2C Interface:**

Another way to connect multiple Arduinos to a Raspberry Pi is to use the I2C interface. This requires connecting each Arduino to the I2C bus on the Raspberry Pi using the SDA and SCL pins.

Rig the receptors such that there is one master, multiple slave devices so that you can read from all connected devices.

You will need to configure each Arduino to act as an I2C slave device, and then use the Wire library in the Arduino IDE to communicate with the Raspberry Pi over the I2C bus.

On the Raspberry Pi side, you can use a Python script to read the data from each Arduino. You can use the smbus library to communicate with the Arduinos. Here is an example code snippet:

```
import smbus

# initialize the I2C bus
bus = smbus.SMBus(1)

# the I2C addresses of the Arduinos
addr1 = 0x04
addr2 = 0x05

while True:
    # read the data from each Arduino
    data1 = bus.read_byte(addr1)
    data2 = bus.read_byte(addr2)
```

```
    # do something with the data from each Arduino
    print(f"Arduino 1: {data1}")
    print(f"Arduino 2: {data2}")
```

Note that the specific implementation will depend on your specific use case, such as the type of data being transferred and the frequency of communication required.