

```

pi@raspberrypi:~ $ cd ~/iot/lesson10
pi@raspberrypi:~/iot/lesson10 $ cat hash_value.py
"""
https://docs.python.org/3/using/cmdline.html#envvar-PYTHONHASHSEED
If PYTHONHASHSEED is not set or set to random, a random value is used to seed the hashes of str and bytes objects.
If PYTHONHASHSEED is set to an integer value, it is used as a fixed seed for generating the hash() of the types covered by the hash randomization.
Its purpose is to allow repeatable hashing, such as for selftests for the interpreter itself, or to allow a cluster of python processes to share hash values.
The integer must be a decimal number in the range [0,4294967295]. Specifying the value 0 will disable hash randomization.

https://www.programiz.com/python-programming/methods/built-in/hash
hash(object) returns the hash value of the object (if it has one). Hash values are integers.
They are used to quickly compare dictionary keys during a dictionary lookup.
Numeric values that compare equal have the same hash value even if they are of different types, as is the case for 1 and 1.0.
For objects with custom __hash__() methods, note that hash() truncates the return value based on the bit width of the host machine.
"""

# hash for integer unchanged
print('The hash for 1 is:', hash(1))

# hash for decimal
print('The hash for 1.0 is:', hash(1.0))
print('The hash for 3.14 is:', hash(3.14))

# hash for string
print('The hash for Python is:', hash('Python'))

# hash for a tuple of vowels
vowels = ('a', 'e', 'i', 'o', 'u')
print('The hash for a tuple of vowels is:', hash(vowels))

# hash for a custom object
class Person:
    def __init__(self, age, name):
        self.age = age
        self.name = name
    def __eq__(self, other):
        return self.age == other.age and self.name == other.name
    def __hash__(self):
        return hash((self.age, self.name))
person = Person(23, 'Adam')
print('The hash for an object of person is:', hash(person))

```

```

pi@raspberrypi:~/iot/lesson10 $ python3 hash_value.py
The hash for 1 is: 1
The hash for 1.0 is: 1
The hash for 3.14 is: 1846836513
The hash for Python is: 1536444896
The hash for a tuple of vowels is: 417059579
The hash for an object of person is: 944420809
pi@raspberrypi:~/iot/lesson10 $ python3 hash_value.py
The hash for 1 is: 1
The hash for 1.0 is: 1
The hash for 3.14 is: 1846836513
The hash for Python is: -1383454192
The hash for a tuple of vowels is: -514686197
The hash for an object of person is: 1918813856

```

```

pi@raspberrypi:~/iot/lesson10 $ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import hashlib
>>> m = hashlib.sha256(b"hello, world")
>>> m.hexdigest()
'09ca7e4eaa6e8ae9c7d261167129184883644d07dfba7cbfbc4c8a2e08360d5b'
>>> m.digest_size
32
>>> m.block_size
64
>>> exit()

```

```

pi@raspberrypi:~/iot/lesson10 $ cat snakecoin.py
# Gerald Nash, "Let's build the tiniest blockchain in less than 50 lines of Python"
import hashlib as hasher
import datetime as date

# Define what a Snakecoin block is
class Block:
    def __init__(self, index, timestamp, data, previous_hash):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.hash = self.hash_block()

    def hash_block(self):
        sha = hasher.sha256()
        sha.update((str(self.index).encode() + str(self.timestamp).encode() + str(self.data).encode() + str(self.previous_hash).encode()).encode())
        return sha.hexdigest()

# Generate genesis block
def create_genesis_block():
    # Manually construct a block with
    # index zero and arbitrary previous hash
    return Block(0, date.datetime.now(), "Genesis Block", "0")

# Generate all later blocks in the blockchain
def next_block(last_block):
    this_index = last_block.index + 1
    this_timestamp = date.datetime.now()
    this_data = "Hey! I'm block " + str(this_index)
    this_hash = last_block.hash
    return Block(this_index, this_timestamp, this_data, this_hash)

# Create the blockchain and add the genesis block
blockchain = [create_genesis_block()]
previous_block = blockchain[0]

# How many blocks should we add to the chain
# after the genesis block
num_of_blocks_to_add = 20

# Add blocks to the chain
for i in range(0, num_of_blocks_to_add):
    block_to_add = next_block(previous_block)
    blockchain.append(block_to_add)
    previous_block = block_to_add
    # Tell everyone about it!
    print("Block #{} has been added to the blockchain!".format(block_to_add.index))
    print("Hash: {}".format(block_to_add.hash))

```

```
pi@raspberrypi:~/iot/lesson10 $ python3 snakecoin.py
Block #1 has been added to the blockchain!
Hash: dc65d1a1b146ba3390eba0158ae6851704addb3dffe6079ba0584c411bcd97bfb

Block #2 has been added to the blockchain!
Hash: 48e4d0d2a3e43ba522f8974296b34a4ec791e9223bf4d756dd165368f0e5893

Block #3 has been added to the blockchain!
Hash: ff0c50463ddee586eada25452a1d208869ad746272b5d767d52c7deebae93b4

Block #4 has been added to the blockchain!
Hash: 794645ddbeba172e4399e5f49133b58ebca5ccbd47ff7a11092c5a1e840df79c

Block #5 has been added to the blockchain!
Hash: fb4ee17f7c23533c0fc80a7169b3f210629f0a9f24e5e8c899651bb8c45b3a9f

Block #6 has been added to the blockchain!
Hash: 8d1c3d846c75574d63e86ec55959097e48eae23c2e57029af96cf7701a9154

Block #7 has been added to the blockchain!
Hash: 54544ffed7172855df53cd133fecaebaf0e2740c921ae7bc65a282229e9ce487

Block #8 has been added to the blockchain!
Hash: 03babe4ad396c2b3f01dfc06221dfd2f660a8258554b8ed440e89a76411a884

Block #9 has been added to the blockchain!
Hash: d5cd44073e9c376b7c50fc6ead1e44a78821f80f6870426b26f077539134392

Block #10 has been added to the blockchain!
Hash: 74ced21e3627994dd2b3a6c021c0742aff67eaf73d95bf77d14001ddcf5391c

Block #11 has been added to the blockchain!
Hash: 82dea599dc1cab45059ec303a4f9c1784eed3ea0b6f79e16c2bfb01198766b65

Block #12 has been added to the blockchain!
Hash: 62ef352cfc64bb38a586d616ccc6a3ecd8101e85074d233296c038e530b706de

Block #13 has been added to the blockchain!
Hash: 75d6be07c0d7e907c5e63de6b9ef510a3aa63f33992baff940f010cd57c8e7f8

Block #14 has been added to the blockchain!
Hash: a8a7aefff97d51863e64b7b99870bb1bf6d110a7222f4d5a543dale8e2e6c6

Block #15 has been added to the blockchain!
Hash: 46a005603c53da605b685d47delacal6d6477a5b950cee82a2804d1d4881b2e2

Block #16 has been added to the blockchain!
Hash: 8bd2aa36ec5b87f71e8b17862c09a67801b425edd7fcbd91e8fb3bbc5a43ab6

Block #17 has been added to the blockchain!
Hash: c79330174fba0813bab3ca958a43d2020471dlaba7f1d05a9b60c0ca5a7e5

Block #18 has been added to the blockchain!
Hash: 231c461b2eafdc5a7e60ab33d6b0bf2820e3b802153224d4f92346cf07d3c8a6

Block #19 has been added to the blockchain!
Hash: 329eddf636cd729b35a5065f974b3b77338731fee9475f6e164d5c365218062d

Block #20 has been added to the blockchain!
Hash: 85b4a9174951782edd94dfd8b80c50e29537e4b0916b68a90a6eb82c0e2045d7

def transaction():
    # On each new POST request,
    # we extract the transaction data
    new_txion = request.get_json()
    # Then we add the transaction to our list
    this_nodes_transactions.append(new_txion)
    # Because the transaction was successfully
    # submitted, we log it to our console
    print("New transaction")
    print(("FROM: {}".format(new_txion['from']).encode('ascii','replace'))))
    print(("TO: {}".format(new_txion['to']).encode('ascii','replace'))))
    print(("AMOUNT: {}".format(new_txion['amount'])))
    # Then we let the client know it worked out
    return "Transaction submission successful\n"

@node.route('/blocks', methods=['GET'])
def get_blocks():
    chain_to_send = blockchain
    # Convert our blocks into dictionaries
    # so we can send them as json objects later
    for i in range(len(chain_to_send)):
        block = chain_to_send[i]
        block_index = str(block.index)
        block_timestamp = str(block.timestamp)
        block_data = str(block.data)
        block_hash = block.hash
        chain_to_send[i] = {
            "index": block.index,
            "timestamp": block.timestamp,
            "data": block.data,
            "hash": block.hash
        }
    chain_to_send = json.dumps(chain_to_send)
    return chain_to_send

def find_new_chains():
    # Get the blockchains of every
    # other node
    other_chains = []
    for node_url in peer_nodes:
        # Get their chains using a GET request
        block = requests.get(node_url + "/blocks").content
        # Convert the JSON object to a Python dictionary
        block = json.loads(block)
        # Add it to our list
        other_chains.append(block)
    return other_chains

def consensus():
    # Get the blocks from other nodes
    other_chains = find_new_chains()
    # If our chain isn't longest,
    # then we store the longest chain
    longest_chain = blockchain
    for chain in other_chains:
        if len(longest_chain) < len(chain):
            longest_chain = chain
    # If the longest chain isn't ours,
    # then we stop mining and set
    # our chain to the longest one
    blockchain = longest_chain

pi@raspberrypi:~/iot/lesson10 $ cat snakecoin-server-full-code.py
# Gerald Nash, "Let's Make the Tiniest Blockchain Bigger Part 2: With More Lines of Python"
# Referred to https://www.pythonanywhere.com/forums/topic/12382/ that fixed sha.update() TypeError: Unicode-objects must be encoded before hashing
# Running on http://127.0.0.1:5000/mine (Reload the page to mine and press CTRL+C to quit)
from flask import Flask
from flask import request
import json
import requests
import hashlib as hasher
import datetime as date
from flask import send_from_directory
import os
node = Flask(__name__)

# Define what a Snakecoin block is
class Block:
    def __init__(self, index, timestamp, data, previous_hash):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.hash = self.hash_block()

    def hash_block(self):
        sha = hasher.sha256()
        sha.update((str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash))
        sha.update((str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash)).encode("utf-8"))
        return sha.hexdigest()

# Generate genesis block
def create_genesis_block():
    # Manually construct a block with
    # index zero and arbitrary previous hash
    return Block(0, date.datetime.now(), {
        "proof-of-work": 9,
        "transactions": None
    }, "0")

# A completely random address of the owner of this node
miner_address = "q3nf394hjg-random-miner-address-34nf314nflkn3oi"
# This node's blockchain copy
blockchain = []
blockchain.append(create_genesis_block())
# Store the transactions that
# this node has in a list
this_nodes_transactions = []
# Store the url data of every
# other node in the network
peer_nodes = []
# so that we can communicate
# with them
peer_nodes = []
# A variable to deciding if we're mining or not
mining = True

@node.route("/")
def hello():
    return "SnakeCoin Server"

@node.route('/favicon.ico')
def favicon():
    return send_from_directory(os.path.join(node.root_path, 'static'),
        'favicon.ico',
        mimetype='image/vnd.microsoft.icon')

@node.route('/txion', methods=['POST'])

    def proof_of_work(last_proof):
        # Create a variable that we will use to find
        # our next proof of work
        incrementor = last_proof + 1
        # Keep incrementing the incrementor until
        # it's equal to a number divisible by 9
        # and the proof of work of the previous
        # block in the chain
        while not (incrementor % 9 == 0 and incrementor % last_proof == 0):
            incrementor += 1
        # Once that number is found,
        # we can return it as a proof
        # of our work
        return incrementor

    @node.route('/mine', methods = ['GET'])
    def mine():
        # Get the last proof of work
        last_block = blockchain[len(blockchain) - 1]
        last_proof = last_block.data['proof-of-work']
        # Find the proof of work for
        # the current block being mined
        # Note: The program will hang here until a new
        # proof of work is found
        proof = proof_of_work(last_proof)
        # Once we find a valid proof of work,
        # we know we can mine a block so
        # we reward the miner by adding a transaction
        this_nodes_transactions.append(
            { "from": "network", "to": miner_address, "amount": 1 }
        )
        # Now we can gather the data needed
        # to create the new block
        new_block_data = {
            "proof-of-work": proof,
            "transactions": list(this_nodes_transactions)
        }
        new_block_index = last_block.index + 1
        new_block_timestamp = this.timestamp = date.datetime.now()
        last_block_hash = last_block.hash
        # Empty transaction list
        this_nodes_transactions[:] = []
        # Now create the
        # new block!
        mined_block = Block(
            new_block_index,
            new_block_timestamp,
            new_block_data,
            last_block_hash
        )
        blockchain.append(mined_block)
        # Let the client know we mined a block
        return json.dumps({
            "index": new_block_index,
            "timestamp": str(new_block_timestamp),
            "data": new_block_data,
            "hash": last_block_hash
        }) + "\n"

node.run()
```

Terminal 1

```
pi@raspberrypi:~/iot/lesson10 $ python3 snakecoin-server-full-code.py
python3 snakecoin-server-full-code.py * Serving Flask app "snakecoin-server-full-code" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

New transaction
FROM: b'akjflw'
TO: b'fjladkj'
AMOUNT: 3

127.0.0.1 - - [13/May/2022 11:46:56] "POST /txion HTTP/1.1" 200 -
127.0.0.1 - - [13/May/2022 11:47:01] "GET /mine HTTP/1.1" 200 -
```

Terminal 2

```
pi@raspberrypi:~ $ curl "localhost:5000/txion" \
> -H "Content-Type: application/json" \
> -d '{"from": "akjflw", "to": "fjladkj", "amount": 3}'
Transaction submission successful
pi@raspberrypi:~ $ curl localhost:5000/mine
{"index": 1, "timestamp": "2022-05-13 11:47:01.629860", "data": {"proof-of-work": 18, "transactions": [{"from": "akjflw", "to": "fjladkj", "amount": 3}, {"from": "network", "to": "q3nf394hjh-random-miner-address-34nf3i4nflkn3oi", "amount": 1}]}, "hash": "b3404dbdad200b2005b22e73f1e3665d7d1db3cd383277019187a0d17594d292"}
```

Terminal 1

```
pi@raspberrypi:~/iot/lesson10 $ git clone https://github.com/satwikkansal/python_blockchain_app.git
Cloning into 'python_blockchain_app'...
remote: Enumerating objects: 146, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 146 (delta 39), reused 36 (delta 36), pack-reused 100
Receiving objects: 100% (146/146), 222.48 KiB | 6.74 MiB/s, done.
Resolving deltas: 100% (71/71), done.
```

```
pi@raspberrypi:~/iot/lesson10 $ cd python_blockchain_app/
pi@raspberrypi:~/iot/lesson10/python_blockchain_app $ nano node_server.py
GNU nano 3.2 node_server.py

from hashlib import sha256
import json
import time

from flask import Flask, request
import requests

class Block:
    def __init__(self, index, transactions, timestamp, previous_hash, nonce=0):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = nonce

    def compute_hash(self):
        """
        A function that return the hash of the block contents.
        """
        block_string = json.dumps(self.__dict__, sort_keys=True)
        return sha256(block_string.encode()).hexdigest()

class Blockchain:
    # difficulty of our PoW algorithm
    difficulty = 2

    def __init__(self):
        self.unconfirmed_transactions = []
        self.chain = []

    def create_genesis_block(self):
        """
        A function to generate genesis block and appends it to
        the chain. The block has index 0, previous_hash as 0, and
        a valid hash.
        """
        genesis_block = Block(0, [], 0, "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        return self.chain[-1]
```

```
pi@raspberrypi:~/iot/lesson10/python_blockchain_app $ python3 node_server.py
pi@raspberrypi:~/iot/lesson10/python_blockchain_app $
```

Terminal 2

```
pi@raspberrypi:~ $ vncserver
VNC(R) Server 6.7.2 (r42622) ARMv6 (May 13 2020 19:34:20)
Copyright (C) 2002-2020 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6/foss.html
OS: Raspbian GNU/Linux 10, Linux 5.10.17, armv7l

VNC(R) Server 6.7.2 (r42622) ARMv6 (May 13 2020 19:44:08)
Copyright (C) 2002-2020 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6/foss.html
OS: Raspbian GNU/Linux 10, Linux 5.10.17, armv7l
<11> 2022-05-13T15:57:20.230Z raspberrypi Xvnc[5913]: VendorConfig: Error in Certificate "CN=GlobalSign,O=Glo
balSign,OU=GlobalSign Root CA - R2": X.509 Error: Certificate expired
Generating private key... done
On some distributions (in particular Red Hat), you may get a better experience
by running vncserver-virtual in conjunction with the system Xorg server, rather
than the old version built-in to Xvnc. More desktop environments and
applications will likely be compatible. For more information on this alternative
implementation, please see: https://www.realvnc.com/doclink/kb-546

Running applications in /etc/vnc/xstartup

VNC Server catchphrase: "Sting monkey Jackson. Chamber Harlem east."
signature: f9-a2-9e-90-43-9d-f6-24

Log file is /home/pi/.vnc/raspberrypi:1.log
New desktop is raspberrypi:1 (192.168.1.230:1)

pi@raspberrypi:~/iot/lesson10/python_blockchain_app $ python3 run_app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 120-405-419
```

```

pi@raspberrypi:~/iot/lesson10/python_blockchain_app $ sudo pip3 install pyota[ccurl]
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pyota[ccurl]
  Downloading https://files.pythonhosted.org/packages/f7/3e/b3c56dc39579d6f8d258e1cef35ceb76cd5463a99032e31ef63703996596/PyOTA-2.1.0-py2.py3-none-any.whl (113kB)
    100% |#####| 122kB 3.0MB/s
Collecting phx-filters; python_version >= "3.5" (from pyota[ccurl])
  Downloading https://files.pythonhosted.org/packages/bc/f5/25ca588777ddebdfld6adf98781d6353ae6910116e6753533da1739f725f/phx_filters-2.0.2-py3-none-any.whl
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from pyota[ccurl]) (1.12.0)
Collecting pysha3 (from pyota[ccurl])
  Downloading https://www.piwheels.org/simple/pysha3/pysha3-1.0.2-cp37-cp37m-linux_armv7l.whl (64kB)
    100% |#####| 71kB 385kB/s
Requirement already satisfied: requests[security]>=2.4.1 in /usr/lib/python3/dist-packages (from pyota[ccurl]) (2.21.0)
Collecting pyota-ccurl; extra == "ccurl" (from pyota[ccurl])
  Downloading https://www.piwheels.org/simple/pyota-ccurl/PyOTA-CCurl-1.0.9-cp37-cp37m-linux_armv7l.whl
Collecting regex>=2018.8.17 (from phx-filters; python_version >= "3.5"->pyota[ccurl])
  Downloading https://www.piwheels.org/simple/regex/regex-2022.4.24-cp37-cp37m-linux_armv7l.whl (329kB)
    100% |#####| 337kB 1.3MB/s
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages (from phx-filters; python_version >= "3.5"->pyota[ccurl]) (2.8.2)
Collecting phx-class-registry (from phx-filters; python_version >= "3.5"->pyota[ccurl])
  Downloading https://files.pythonhosted.org/packages/32/70/666e3ccf2e8b49c3cabd807570b3e410d9fbda9dcae15421f656a770d377/phx_class_registry-3.0.5-py3-none-any.whl
Requirement already satisfied: pytz in /usr/lib/python3/dist-packages (from phx-filters; python_version >= "3.5"->pyota[ccurl]) (2019.1)
Requirement already satisfied: cryptography>=1.3.4 in /usr/lib/python3/dist-packages (from requests[security]>=2.4.1->pyota[ccurl]) (2.6.1)
Requirement already satisfied: idna>=2.0.0 in /usr/lib/python3/dist-packages (from requests[security]>=2.4.1->pyota[ccurl]) (2.6)
Requirement already satisfied: pyOpenSSL>=0.14 in /usr/lib/python3/dist-packages (from requests[security]>=2.4.1->pyota[ccurl]) (19.0.0)
Installing collected packages: regex, phx-class-registry, phx-filters, pysha3, pyota-ccurl, pyota
Successfully installed phx-class-registry-3.0.5 phx-filters-2.0.2 pyota-2.1.0 pyota-ccurl-1.0.9 pysha3-1.0.2 regex-2022.4.24
pi@raspberrypi:~/iot/lesson10/python_blockchain_app $ cd ..
pi@raspberrypi:~/iot/lesson10 $ cat iri_node_info.py
from iota import Iota

# Create a new instance of the IOTA API object
# Specify which node to connect to
api = Iota(adapter = 'https://nodes.devnet.iota.org:443')

# Call the 'get_node_info()' method for information about the node and the Tangle
response = api.get_node_info()

print(response)
pi@raspberrypi:~/iot/lesson10 $ python3 iri_node_info.py
{'latestMilestone': TransactionHash(b'HRFBCEPR9FIDQFSGSHJJDZCJHWEPMFGDTNNHQMHPFHOLFPVBK9DSBEHXDTLESOIUBWHARFUCHXXJWYD999'), 'latestSolidSubtangleMilestone': TransactionH
ash(b'HRFBCEPR9FIDQFSGSHJJDZCJHWEPMFGDTNNHQMHPFHOLFPVBK9DSBEHXDTLESOIUBWHARFUCHXXJWYD999'), 'appName': 'HORNET', 'appVersion': '0.5.6', 'coordinatorAddress': 'GYISMBVRKS
CEXXTUPBWTIHRZC2IKIRPYAHAYKMNTPZSCSDNADDWAEUNHKUERZCTVAYJCNFGTNUH9OGTW', 'duration': 0, 'features': ['RemotePOW', 'WereAddressesSpentFrom'], 'isHealthy': True, 'isSync
ed': True, 'lastSnapshotMilestoneIndex': 3610585, 'latestMilestoneIndex': 3610679, 'latestSolidSubtangleMilestoneIndex': 3610679, 'milestoneStartIndex': 3209121, 'ne
ighbors': 3, 'time': 1652457929000, 'tips': 8, 'transactionsToRequest': 0}

pi@raspberrypi:~/iot/lesson10 $ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
--2022-05-13 12:06:18-- http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
Resolving www.airspayce.com (www.airspayce.com)... 192.185.48.187
Connecting to www.airspayce.com (www.airspayce.com)|192.185.48.187|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 265906 (260K) [application/x-gzip]
Saving to: 'bcm2835-1.60.tar.gz'

bcm2835-1.60.tar.gz      100%[=====>] 259.67K  713KB/s   in 0.4s

2022-05-13 12:06:19 (713 KB/s) - 'bcm2835-1.60.tar.gz' saved [265906/265906]

pi@raspberrypi:~/iot/lesson10 $ tar zxvf bcm2835-1.60.tar.gz
bcm2835-1.60/
bcm2835-1.60/configure.ac
bcm2835-1.60/COPYING
bcm2835-1.60/examples/
bcm2835-1.60/examples/spi/
bcm2835-1.60/examples/spi/spi.c
bcm2835-1.60/examples/input/
bcm2835-1.60/examples/input/input.c
bcm2835-1.60/examples/spiram/
bcm2835-1.60/examples/spiram/spiram.h
bcm2835-1.60/examples/spiram/spiram_test.c
bcm2835-1.60/examples/spiram/spiram.c
bcm2835-1.60/examples/event/
bcm2835-1.60/examples/event/event.c
bcm2835-1.60/examples/pwm/
bcm2835-1.60/examples/pwm/pwm.c
bcm2835-1.60/examples/gpio/
bcm2835-1.60/examples/gpio/gpio.c
bcm2835-1.60/examples/blink/
bcm2835-1.60/examples/blink/blink.c
bcm2835-1.60/examples/spin/
bcm2835-1.60/examples/spin/spin.c
bcm2835-1.60/examples/i2c/
bcm2835-1.60/examples/i2c/i2c.c
bcm2835-1.60/config.guess
bcm2835-1.60/NEWS
bcm2835-1.60/depcomp
bcm2835-1.60/configure
bcm2835-1.60/aclocal.m4
bcm2835-1.60/compile
bcm2835-1.60/INSTALL
bcm2835-1.60/Makefile.in
bcm2835-1.60/missing
bcm2835-1.60/config.sub
bcm2835-1.60/Makefile.am
bcm2835-1.60/ChangeLog
bcm2835-1.60/README
bcm2835-1.60/install-sh
bcm2835-1.60/config.h.in
bcm2835-1.60/src/
bcm2835-1.60/src/test.c
bcm2835-1.60/src/bcm2835.h
bcm2835-1.60/src/bcm2835.c
bcm2835-1.60/src/Makefile.in
bcm2835-1.60/src/Makefile.am
bcm2835-1.60/ltmain.sh
bcm2835-1.60/AUTHORS
bcm2835-1.60/doc/
bcm2835-1.60/doc/Doxyfile.in
bcm2835-1.60/doc/Makefile.in
bcm2835-1.60/doc/Makefile.am
bcm2835-1.60/test-driver

```



```

pi@raspberrypi:~/iot/lesson10 $ cd bcm2835-1.60
pi@raspberrypi:~/iot/lesson10/bcm2835-1.60 $ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for clock_gettime in -lrt... yes
checking for doxygen... no
configure: WARNING: Doxygen not found - continuing without Doxygen support
checking for ranlib... ranlib
checking for gcc... (cached) gcc
checking whether we are using the GNU C compiler... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for gcc option to accept ISO C89... (cached) none needed
checking whether gcc understands -c and -o together... (cached) yes
checking dependency style of gcc... (cached) gcc3
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating doc/Makefile
config.status: creating config.h
config.status: executing depfiles commands
pi@raspberrypi:~/iot/lesson10/bcm2835-1.60 $ make
make all-recursive
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60'
Making all in src
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
gcc -DHAVE_CONFIG_H -I. -I.. -g -O2 -MT bcm2835.o -MD -MP -MF .deps/bcm2835.Tpo -c -o bcm2835.o bcm2835.c
mv -f .deps/bcm2835.Tpo .deps/bcm2835.Po
rm -f libbcm2835.a
ar cru libbcm2835.a bcm2835.o
ar: `u' modifier ignored since `D' is the default (see `U')
ranlib libbcm2835.a
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
Making all in doc
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60'
pi@raspberrypi:~/iot/lesson10/bcm2835-1.60 $ sudo make check
Making check in src
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make test
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
gcc -DHAVE_CONFIG_H -I. -I.. -g -O2 -MT test.o -MD -MP -MF .deps/test.Tpo -c -o test.o test.c
mv -f .deps/test.Tpo .deps/test.Po
gcc -g -O2 -o test test.o ./libbcm2835.a -lrt
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make check-TESTS
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make[3]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
PASS: test

=====
Testsuite summary for bcm2835 1.60
=====
# TOTAL: 1
# PASS: 1
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
make[3]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
Making check in doc
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[1]: Nothing to be done for 'check'.
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60'
pi@raspberrypi:~/iot/lesson10/bcm2835-1.60 $ sudo make install
Making install in src
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
/usr/bin/mkdir -p '/usr/local/lib'
/usr/bin/install -c -m 644 libbcm2835.a '/usr/local/lib'
(cd '/usr/local/lib' && ranlib libbcm2835.a)
/usr/bin/mkdir -p '/usr/local/include'
/usr/bin/install -c -m 644 bcm2835.h '/usr/local/include'
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/src'
Making install in doc
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60/doc'
make[1]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[2]: Entering directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60'
make[1]: Leaving directory '/home/pi/iot/lesson10/bcm2835-1.60'
pi@raspberrypi:~/iot/lesson10/bcm2835-1.60 $ cd
pi@raspberrypi:~ $

```

```

pi@raspberrypi:~/dht11-raspi3 $ npm install
npm WARN npm npm does not support Node.js v10.24.0
npm WARN npm You should probably upgrade to a newer version of node as we
npm WARN npm can't make any promises that npm will work with this version.
npm WARN npm Supported releases of Node.js are the latest release of 4, 6, 7, 8, 9.
npm WARN npm You can find the latest version at https://nodejs.org/
npm WARN notice [SECURITY] lodash has the following vulnerabilities: 1 critical, 3 high, 2 moderate. Go here for more details: https://github.com/advisories?query=lodash
h - Run 'npm i npm@latest -g' to upgrade your npm version, and then 'npm audit' to get more info.

> node-dht-sensor@0.0.34 preinstall /home/pi/dht11-raspi3/node_modules/node-dht-sensor
> ./check-lib.sh

Library bcm2835 found.

> node-dht-sensor@0.0.34 install /home/pi/dht11-raspi3/node_modules/node-dht-sensor
> node-gyp configure

> node-dht-sensor@0.0.34 postinstall /home/pi/dht11-raspi3/node_modules/node-dht-sensor
> node-gyp build

make: Entering directory '/home/pi/dht11-raspi3/node_modules/node-dht-sensor/build'
  CX(target) Release/obj.target/node_dht_sensor/node-dht-sensor.o
In file included from ../node-dht-sensor.cpp:2:
../nan/nan.h: In function 'v8::Local<v8::Value> Nan::MakeCallback(v8::Local<v8::Object>, v8::Local<v8::Function>, int, v8::Local<v8::Value>*)':
../nan/nan.h:840:60: warning: 'v8::Local<v8::Value> node::MakeCallback(v8::Isolate*, v8::Local<v8::Object>, v8::Local<v8::Function>, int, v8::Local<v8::Value>*)' is deprecated: Use MakeCallback(..., async_context) [-Wdeprecated-declarations]
   v8::Isolate::GetCurrent(), target, func, argc, argv);
   ^
In file included from ../node-dht-sensor.cpp:1:
/usr/include/nodejs/src/node.h:177:50: note: declared here
   NODE_EXTERN v8::Local<v8::Value> MakeCallback(
   ^~~~~~
/usr/include/nodejs/src/node.h:91:42: note: in definition of macro 'NODE_DEPRECATED'
   __attribute__((deprecated(message))) declarator
   ^~~~~~
In file included from ../node-dht-sensor.cpp:2:
../nan/nan.h:840:60: warning: 'v8::Local<v8::Value> node::MakeCallback(v8::Isolate*, v8::Local<v8::Object>, v8::Local<v8::Function>, int, v8::Local<v8::Value>*)' is deprecated: Use MakeCallback(..., async_context) [-Wdeprecated-declarations]
   v8::Isolate::GetCurrent(), target, func, argc, argv);
   ^
In file included from ../node-dht-sensor.cpp:1:
/usr/include/nodejs/src/node.h:177:50: note: declared here
   NODE_EXTERN v8::Local<v8::Value> MakeCallback(
   ^~~~~~
/usr/include/nodejs/src/node.h:91:42: note: in definition of macro 'NODE_DEPRECATED'
   __attribute__((deprecated(message))) declarator
   ^~~~~~
In file included from ../node-dht-sensor.cpp:2:
../nan/nan.h: In function 'v8::Local<v8::Value> Nan::MakeCallback(v8::Local<v8::Object>, v8::Local<v8::String>, int, v8::Local<v8::Value>*)':
../nan/nan.h:855:62: warning: 'v8::Local<v8::Value> node::MakeCallback(v8::Isolate*, v8::Local<v8::Object>, v8::Local<v8::String>, int, v8::Local<v8::Value>*)' is deprecated: Use MakeCallback(..., async_context) [-Wdeprecated-declarations]
   v8::Isolate::GetCurrent(), target, symbol, argc, argv);
   ^
In file included from ../node-dht-sensor.cpp:1:
/usr/include/nodejs/src/node.h:170:50: note: declared here
   NODE_EXTERN v8::Local<v8::Value> MakeCallback(
   ^~~~~~
/usr/include/nodejs/src/node.h:91:42: note: in definition of macro 'NODE_DEPRECATED'
   __attribute__((deprecated(message))) declarator
   ^~~~~~

```

```
pi@raspberrypi:~/dht11-raspi3 $ nano mam_publish.js
```

```
GNU nano 3.2
```

```
mam_publish.js
```



Author: Robert Lie (mobilefish.com)

The mam_publish.js file publishes random generated numbers on the tangle using MAM.

This file will work on a computer or Raspberry Pi.

The published data can be viewed using the mam_receive.js file or

<https://www.mobilefish.com/services/cryptocurrency/mam.html> (Select option: Data receiver)

Usage:

- 1) You can change the default settings: MODE, SIDEKEY, SECURITYLEVEL or TIMEINTERVAL
If you do, make the same changes in mam_receive.js file.
- 2) Start the app: node mam_publish.js

More information:

https://www.mobilefish.com/developer/iota/iota_quickguide_raspi_mam.html

*/

```

const Mam = require('./lib/mam.client.js');
const IOTA = require('iota.lib.js');
const moment = require('moment');
const iota = new IOTA({ provider: 'https://nodes.testnet.iota.org:443' });

const MODE = 'restricted'; // public, private or restricted
const SIDEKEY = 'mysecret'; // Enter only ASCII characters. Used only in restricted mode
const SECURITYLEVEL = 3; // 1, 2 or 3
const TIMEINTERVAL = 30; // seconds

// Initialise MAM State
let mamState = Mam.init(iota, undefined, SECURITYLEVEL);

// Set channel mode
if (MODE == 'restricted') {
  const key = iota.utils.toTrytes(SIDEKEY);
  mamState = Mam.changeMode(mamState, MODE, key);
} else {
  mamState = Mam.changeMode(mamState, MODE);
}

```

```
pi@raspberrypi:~/dht11-raspi3 $ nano mam_receive.js
```

```
GNU nano 3.2
```

```
mam_receive.js
```

```
/*
Author: Robert Lie (mobilefish.com)

The mam_receive.js file extracts stored data from the tangle using MAM.
This extracted data will be displayed on the screen.
This file will work on a computer or Raspberry Pi.
Instead of this file you can also use another tool to display the data:
https://www.mobilefish.com/services/cryptocurrency/mam.html (Select option: Data receiver)
```

Usage:

- 1) You can change the default settings: MODE or SIDEKEY
If you do, make the same changes in mam_publish.js and mam_sensor.js files.
- 2) Start the app: node mam_receive.js <root>

More information:

https://www.mobilefish.com/developer/iota/iota_quickguide_raspi_mam.html
*/

```
const Mam = require('./lib/mam.client.js');
const IOTA = require('iota.lib.js');
const iota = new IOTA({ provider: 'https://nodes.testnet.iota.org:443' });

const MODE = 'restricted'; // public, private or restricted
const SIDEKEY = 'mysecret'; // Enter only ASCII characters. Used only in restricted mode

let root;
let key;

// Check the arguments
const args = process.argv;
if(args.length !== 3) {
    console.log('Missing root as argument: node mam_receive.js <root>');
    process.exit();
} else if(!iota.valid.isAddress(args[2])){
    console.log('You have entered an invalid root: ' + args[2]);
    process.exit();
} else {
    root = args[2];
},
```

```
pi@raspberrypi:~/dht11-raspi3 $ nano mam_sensor.js
```

```
GNU nano 3.2
```

```
mam_sensor.js
```

```
/*
Author: Robert Lie (mobilefish.com)

The mam_sensor.js file publishes DHT11 sensor data (temperature and humidity) on the tangle using MAM.
This file only works on the Raspberry Pi.
The published data can be viewed using the mam_receive.js file or
https://www.mobilefish.com/services/cryptocurrency/mam.html (Select option: Data receiver)
```

Usage:

- 1) Connect DHT11 sensor to Raspberry Pi.
- 2) Do not forget to type: npm install
- 3) You can change the default settings: MODE, SIDEKEY, SECURITYLEVEL or TIMEINTERVAL
If you do, make the same changes in mam_receive.js file.
- 4) Start the app: node mam_sensor.js

More information:

https://www.mobilefish.com/developer/iota/iota_quickguide_raspi_mam.html
*/

```
const sensor = require('node-dht-sensor');
const Mam = require('./lib/mam.client.js');
const IOTA = require('iota.lib.js');
const moment = require('moment');


const iota = new IOTA({ provider: 'https://nodes.testnet.iota.org:443' });

const MODE = 'restricted'; // public, private or restricted
const SIDEKEY = 'mysecret'; // Enter only ASCII characters. Used only in restricted mode
const SECURITYLEVEL = 3; // 1, 2 or 3
const TIMEINTERVAL = 30; // seconds
const SENSORTYPE = 11; // 11=DHT11, 22=DHT22
const GPIOPIN = 4; // The Raspi gpio pin where data from the DHT11 is read

// Initialise MAM State
let mamState = Mam.init(iota, undefined, SECURITYLEVEL);

// Set channel mode
if (MODE === 'restricted') {
    const key = iota.utils.toTrytes(SIDEKEY);
    mamState = Mam.changeMode(mamState, MODE, key);
} else {
    mamState = Mam.changeMode(mamState, MODE);
}
```


edit 14 to 9 by searching File

```
var _ref5 = _asyncToGenerator( /*# PURE */regeneratorRuntime.mark(function _callee5(trytes, root) {  
  var depth = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : 6;  
  var mwm = arguments.length > 3 && arguments[3] !== undefined ? arguments[3] : 9;   
  var transfers, objs;
```

Terminal 1

```
pi@raspberrypi:~/dht11-raspi3 $ node mam_publish.js  
json= { data: 47, dateTime: '13/05/2022 04:16:35' }  
Root: MO9VICICCHWVCDGLPNJGDWNPQBQCHHKCCAWLUBMNGHEFEUYZGVWTVPNBDMYAW9ZXKDNAIRCGQCFDXEUU  
Address: IEBYDOAWFUYQVIINZCTDNDVUSEP9JDAGRHTJOE9TLRDDOFMWEENPWPQ9OMMRUNVNDQYCEGKUYDJBKJNM  
failed to attach message:  
Error: Invalid Response: Error: getaddrinfo ENOTFOUND nodes.testnet.iota.org nodes.testnet.iota.org:443  
  at GetAddrInfoReqWrap.onlookup [as oncomplete] (dns.js:56:26)  
  at Object.invalidResponse (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/errors/requestErrors.js:5:12)  
  at makeRequest.prepareResult (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:285:24)  
  at exports.XMLHttpRequest.request.onreadystatechange (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:71:25)  
  at exports.XMLHttpRequest.dispatchEvent (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:591:25)  
  at setState (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:610:14)  
  at exports.XMLHttpRequest.handleError (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:532:5)  
  at ClientRequest.errorHandler (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:459:14)  
  at ClientRequest.emit (events.js:198:13)  
  at TLSSocket.socketErrorListener (http_client.js:401:9)  
  at TLSSocket.emit (events.js:198:13)  
dateTime: 13/05/2022 04:16:35, data: 47, root: MO9VICICCHWVCDGLPNJGDWNPQBQCHHKCCAWLUBMNGHEFEUYZGVWTVPNBDMYAW9ZXKDNAIRCGQCFDXEUU  
json= { data: 43, dateTime: '13/05/2022 04:17:07' }  
Root: YYBOUGQISNMUVTL09KBXFQDEPQLBT9GDTIYIZDIEKXLSTW9YCGLJANFXZGZFAJOTGIEBXE9YMODNIKZRB  
Address: EOSCNCXZRSYKHBVFNUVONDGXRQRIIVXGVQP9PARKU9PZMINLEKNNBTTHNNDKAPLU9YEVPCGYZXKBYLZB  
failed to attach message:  
Error: Invalid Response: Error: getaddrinfo ENOTFOUND nodes.testnet.iota.org nodes.testnet.iota.org:443  
  at GetAddrInfoReqWrap.onlookup [as oncomplete] (dns.js:56:26)  
  at Object.invalidResponse (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/errors/requestErrors.js:5:12)  
  at makeRequest.prepareResult (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:285:24)  
  at exports.XMLHttpRequest.request.onreadystatechange (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:71:25)  
  at exports.XMLHttpRequest.dispatchEvent (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:591:25)  
  at setState (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:610:14)  
  at exports.XMLHttpRequest.handleError (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:532:5)  
  at ClientRequest.errorHandler (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:459:14)  
  at ClientRequest.emit (events.js:198:13)  
  at TLSSocket.socketErrorListener (http_client.js:401:9)  
  at TLSSocket.emit (events.js:198:13)  
dateTime: 13/05/2022 04:17:07, data: 43, root: YYBOUGQISNMUVTL09KBXFQDEPQLBT9GDTIYIZDIEKXLSTW9YCGLJANFXZGZFAJOTGIEBXE9YMODNIKZRB  
json= { data: 83, dateTime: '13/05/2022 04:17:37' }  
Root: VLXATWKOPIYDBZPCYYBKV9GDFKHYHQHFKJSEOWLLNLHQ9SBLMCTAAVCGYDIAGZ9PNHNNNOAMREYFEXR  
Address: NIWCTTUGKMQQCCXUDQONKFY9DRWDZRPGAOUJA9TDEBDZSDUHKVCPNREQSPGAF9SBKBJFKRWUYGPHTTQE9  
failed to attach message:  
Error: Invalid Response: Error: getaddrinfo ENOTFOUND nodes.testnet.iota.org nodes.testnet.iota.org:443  
  at GetAddrInfoReqWrap.onlookup [as oncomplete] (dns.js:56:26)  
  at Object.invalidResponse (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/errors/requestErrors.js:5:12)  
  at makeRequest.prepareResult (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:285:24)  
  at exports.XMLHttpRequest.request.onreadystatechange (/home/pi/dht11-raspi3/node_modules/iota.lib.js/lib/utlis/makeRequest.js:71:25)  
  at exports.XMLHttpRequest.dispatchEvent (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:591:25)  
  at setState (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:610:14)  
  at exports.XMLHttpRequest.handleError (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:532:5)  
  at ClientRequest.errorHandler (/home/pi/dht11-raspi3/node_modules/xmlhttprequest/lib/XMLHttpRequest.js:459:14)  
  at ClientRequest.emit (events.js:198:13)  
  at TLSSocket.socketErrorListener (http_client.js:401:9)  
  at TLSSocket.emit (events.js:198:13)
```

Terminal 2

```
pi@raspberrypi:~/dht11-raspi3 $ node mam_receive.js MO9VICICCHWVCDGLPNJGDWNPQBQC  
HHKCCAWLUBMNGHEFEUYZGVWTVPNBDMYAW9ZXKDNAIRCGQCFDXEUU
```