

BÁO CÁO CHI TIẾT: CẤU TRÚC FILE IQTCP.H5

1. TỔNG QUAN

File iqtcp.h5 là file HDF5 chứa dữ liệu IQ (In-phase và Quadrature) từ Narrowband TCP. File này được đọc bởi module Python `reader_iqtcp_h5.py` dựa trên code MATLAB `read_iqtcp_h5_verge2.m`.

Thông tin cơ bản:

- **Định dạng**: HDF5 (Hierarchical Data Format version 5)
- **Loại dữ liệu**: IQ samples (Narrowband TCP)
- **Số sessions**: 46,072 sessions
- **Cấu trúc chính**:
 - `/attribute` : Chứa metadata (thông tin chung, DDC, request, ...)
 - `/session` : Chứa dữ liệu IQ của các sessions

2. CẤU TRÚC FILE H5

2.1. Cấu trúc tổng thể

```
iqtcp.h5
├── /attribute                                # Group chứa metadata
│   ├── Attributes (trực tiếp)                # → global_info
│   ├── /ddc                                    # → ddc_info
│   │   └── Attributes
│   ├── /request                               # → request_info
│   │   └── Attributes
│   └── ... (các group con khác)
└── /session                                   # Group chứa dữ liệu IQ
    ├── /000000000000000000000000
    │   ├── /i                                     # Dataset: In-phase samples
    │   └── /q                                     # Dataset: Quadrature samples
    ├── /000000000000000000000001
    │   ├── /i
```

```
|   └─ /q  
└ ... (46,072 sessions)
```

2.2. Chi tiết các thành phần

A. `/attribute` Group

Group này chứa tất cả metadata của file:

1. **Attributes trực tiếp tại `/attribute`**:

- Được đọc vào `data['global_info']`
- Chứa thông tin chung như: `client_ip`, `frequency`, `bandwidth`, `channel`, `mission`, ...

2. **Sub-groups trong `/attribute`**:

- Mỗi sub-group được đọc vào `data['{name}_info']`
- Ví dụ: `/attribute/ddc` → `data['ddc_info']`
- Ví dụ: `/attribute/request` → `data['request_info']`
- Các sub-group có thể chứa:
 - Attributes (metadata)
 - Datasets (dữ liệu bổ sung, ví dụ: labels)

B. `/session` Group

Group này chứa dữ liệu IQ của tất cả sessions:

- Mỗi session có ID dạng: `00000000000000000000000000000000`, `00000000000000000000000000000001`, ...
- Mỗi session chứa 2 datasets:
 - `i`: In-phase samples (kiểu int32, shape: (512,))
 - `q`: Quadrature samples (kiểu int32, shape: (512,))

3. INPUT VÀ OUTPUT CỦA READER

3.1. Input

Hàm: `read_iqtcp_h5(filename: str)`

Tham số:

- `filename` (str): Đường dẫn đến file H5 cần đọc
- - Ví dụ: `'..../00_DATA_h5/iqtcp.h5'

3.2. Output

Kiểu trả về: `Dict[str, Any]`

Cấu trúc output:

```
{  
    'global_info': {  
        # Dictionary chứa tất cả attributes từ /attribute  
        # Ví dụ:  
        # 'client_ip': '192.168.1.100',  
        # 'frequency': 1000000000.0,    # Hz  
        # 'bandwidth': 20000000.0,     # Hz  
        # 'channel': 1,  
        # 'mission': 'surveillance',  
        # ...  
    },  
  
    'ddc_info': {  
        # Dictionary chứa attributes từ /attribute/ddc  
        # Ví dụ:  
        # 'channelIndex': 0,  
        # 'frequency': 950000000.0,  
        # 'deviceId': 'device_001',  
        # ...  
    },  
  
    'request_info': {  
        # Dictionary chứa attributes từ /attribute/request  
        # Ví dụ:  
        # 'channel': 1,  
        # 'frequency': 1000000000.0,  
        # 'bandwidth': 20000000.0,  
        # 'mission': 'surveillance',  
        # ...  
    }  
}
```

```

        # Dictionary chứa attributes từ /attribute/request
        # Ví dụ:
        # 'fileName': 'data_20240113.h5',
        # 'duration': 3600.0, # seconds
        # 'checkpoint': True,
        # ...
    },

    # Các info khác (nếu có):
    # 'label_info': {...},
    # 'other_info': {...},

    'sessions': [
        {
            'id': '000000000000000000000000', # Session ID (string)
            'i': np.array([...]),           # In-phase samples (numpy array, int32)
            'q': np.array([...]),           # Quadrature samples (numpy array,
int32)
            'iq': np.array([...])           # Complex IQ = I + j*Q (numpy array,
complex128)
        },
        {
            'id': '000000000000000000000001',
            'i': np.array([...]),
            'q': np.array([...]),
            'iq': np.array([...])
        },
        # ... (46,072 sessions)
    ]
}

```

4. HƯỚNG DẪN SỬ DỤNG READER

4.1. Cài đặt

Yêu cầu:

- Python 3.6+
- Thư viện: `h5py`, `numpy`

Cài đặt dependencies:

```
pip install h5py numpy
```

4.2. Cách sử dụng cơ bản

```
from reader_iqtcp_h5 import read_iqtcp_h5

# Đọc file
filename = 'path/to/iqtcp.h5'
data = read_iqtcp_h5(filename)

# Truy cập thông tin
print(f"Số sessions: {len(data['sessions'])}")
print(f"Frequency: {data['global_info']['frequency']} Hz")
```

4.3. Cách lấy các trường thông tin output

A. Lấy thông tin chung (Global Info)

```
# Lấy toàn bộ global_info
global_info = data['global_info']

# Lấy từng trường cụ thể
client_ip = data['global_info']['client_ip']
frequency = data['global_info']['frequency'] # Hz
bandwidth = data['global_info']['bandwidth'] # Hz
channel = data['global_info']['channel']
mission = data['global_info']['mission']

# In ra tất cả các trường
for key, value in data['global_info'].items():
    print(f"{key}: {value}")
```

B. Lấy thông tin DDC

```
# Lấy toàn bộ ddc_info
ddc_info = data['ddc_info']

# Lấy từng trường cụ thể
channel_index = data['ddc_info']['channelIndex']
ddc_frequency = data['ddc_info']['frequency']
device_id = data['ddc_info']['deviceId']
```

C. Lấy thông tin Request

```
# Lấy toàn bộ request_info
request_info = data['request_info']

# Lấy từng trường cụ thể
file_name = data['request_info']['fileName']
duration = data['request_info']['duration'] # seconds
checkpoint = data['request_info']['checkpoint']
```

D. Lấy dữ liệu IQ từ Sessions

****1. Lấy một session cụ thể:****

```
# Lấy session đầu tiên
session_0 = data['sessions'][0]

# Lấy session theo index
session_idx = 100
session_100 = data['sessions'][session_idx]

# Lấy session theo ID
target_id = '0000000000000000100'
session = next((s for s in data['sessions'] if s['id'] == target_id), None)
```

****2. Lấy dữ liệu I, Q, và IQ:****

```
# Lấy session
session = data['sessions'][0]

# Lấy Session ID
session_id = session['id']
print(f"Session ID: {session_id}")

# Lấy dữ liệu I (In-phase)
i_data = session['i'] # numpy array, shape: (512,), dtype: int32
print(f"I data: shape={i_data.shape}, dtype={i_data.dtype}")
print(f"I samples: {i_data[:5]}") # 5 giá trị đầu

# Lấy dữ liệu Q (Quadrature)
q_data = session['q'] # numpy array, shape: (512,), dtype: int32
print(f"Q data: shape={q_data.shape}, dtype={q_data.dtype}")
print(f"Q samples: {q_data[:5]}") # 5 giá trị đầu
```

```

# Lấy dữ liệu IQ phức (I + j*Q)
iq_data = session['iq'] # numpy array, shape: (512,), dtype: complex128
print(f"IQ data: shape={iq_data.shape}, dtype={iq_data.dtype}")
print(f"IQ samples: {iq_data[:5]}") # 5 giá trị đầu

# Tính toán từ IQ phức
import numpy as np

# Biên độ (Magnitude)
magnitude = np.abs(iq_data)
print(f"Magnitude: min={magnitude.min():.2f}, max={magnitude.max():.2f}")

# Phase (Góc pha)
phase = np.angle(iq_data)
print(f"Phase: min={phase.min():.3f}, max={phase.max():.3f}")

# Power
power = np.abs(iq_data) ** 2
print(f"Power: mean={power.mean():.2f}")

```

3. Duyệt qua tất cả sessions:

```

# Duyệt qua tất cả sessions
for i, session in enumerate(data['sessions']):
    session_id = session['id']
    i_data = session['i']
    q_data = session['q']
    iq_data = session['iq']

    # Xử lý dữ liệu...
    magnitude = np.abs(iq_data)
    phase = np.angle(iq_data)

    # Ví dụ: chỉ xử lý 10 sessions đầu
    if i >= 10:
        break

```

4. Lọc sessions có dữ liệu:

```

# Chỉ lấy các sessions có dữ liệu I và Q
valid_sessions = [
    s for s in data['sessions']
    if s['i'] is not None and s['q'] is not None
    and len(s['i']) > 0 and len(s['q']) > 0
]

print(f"Số sessions có dữ liệu: {len(valid_sessions)}")

```

5. Lấy thống kê của một session:

```
session = data['sessions'][0]

if session['i'] is not None and len(session['i']) > 0:
    i_stats = {
        'min': session['i'].min(),
        'max': session['i'].max(),
        'mean': session['i'].mean(),
        'std': session['i'].std(),
        'size': len(session['i'])
    }
    print(f"I statistics: {i_stats}")

if session['q'] is not None and len(session['q']) > 0:
    q_stats = {
        'min': session['q'].min(),
        'max': session['q'].max(),
        'mean': session['q'].mean(),
        'std': session['q'].std(),
        'size': len(session['q'])
    }
    print(f"Q statistics: {q_stats}")

if session['iq'] is not None and len(session['iq']) > 0:
    iq_mag = np.abs(session['iq'])
    iq_stats = {
        'magnitude_min': iq_mag.min(),
        'magnitude_max': iq_mag.max(),
        'magnitude_mean': iq_mag.mean(),
        'phase_min': np.angle(session['iq']).min(),
        'phase_max': np.angle(session['iq']).max(),
        'size': len(session['iq'])
    }
    print(f"IQ statistics: {iq_stats}")
```

5. VÍ DỤ CODE HOÀN CHỈNH

```
#!/usr/bin/env python3
"""
Ví dụ sử dụng reader_iqtcp_h5.py
"""
```

```

import numpy as np
from reader_iqtcp_h5 import read_iqtcp_h5

# 1. Đọc file
filename = '../../../../../00_DATA_h5/iqtcp.h5'
print(f"Đang đọc file: {filename}")
data = read_iqtcp_h5(filename)

# 2. Hiển thị thông tin chung
print("\n==== THÔNG TIN CHUNG ===")
print(f"Số sessions: {len(data['sessions'])}")

if 'global_info' in data:
    print("\nGlobal Info:")
    for key, value in data['global_info'].items():
        print(f"  {key}: {value}")

if 'ddc_info' in data:
    print("\nDDC Info:")
    for key, value in data['ddc_info'].items():
        print(f"  {key}: {value}")

if 'request_info' in data:
    print("\nRequest Info:")
    for key, value in data['request_info'].items():
        print(f"  {key}: {value}")

# 3. Xử lý dữ liệu từ session đầu tiên
print("\n==== XỬ LÝ SESSION ĐẦU TIÊN ===")
if data['sessions']:
    session = data['sessions'][0]
    print(f"Session ID: {session['id']}")

    if session['i'] is not None:
        print(f"I: shape={session['i'].shape}, dtype={session['i'].dtype}")
        print(f"  Min={session['i'].min()}, Max={session['i'].max()},
Mean={session['i'].mean():.2f}")

    if session['q'] is not None:
        print(f"Q: shape={session['q'].shape}, dtype={session['q'].dtype}")
        print(f"  Min={session['q'].min()}, Max={session['q'].max()},
Mean={session['q'].mean():.2f}")

    if session['iq'] is not None:
        iq_mag = np.abs(session['iq'])
        print(f"IQ: shape={session['iq'].shape}, dtype={session['iq'].dtype}")
        print(f"  Magnitude: Min={iq_mag.min():.2f}, Max={iq_mag.max():.2f},
Mean={iq_mag.mean():.2f}")
        print(f"  Phase: Min={np.angle(session['iq']).min():.3f},
Max={np.angle(session['iq']).max():.3f}")

# 4. Duyệt qua một số sessions
print("\n==== DUYỆT QUA 5 SESSIONS ĐẦU ===")
for i in range(min(5, len(data['sessions']))):
    session = data['sessions'][i]

```

```

if session['iq'] is not None and len(session['iq']) > 0:
    iq_mag = np.abs(session['iq'])
    print(f"Session {i} (ID: {session['id']}): "
          f"Magnitude mean={iq_mag.mean():.2f}, "
          f"Size={len(session['iq'])}")

```

6. BẢNG TÓM TẮT CẤU TRÚC OUTPUT

Trường	Kiểu dữ liệu	Mô tả	Ví dụ truy cập
`global_info`	`dict`	Attributes từ `/attribute`	`data['global_info']['frequency']`
`ddc_info`	`dict`	Attributes từ `/attribute/ddc`	`data['ddc_info']['channelIndex']`
`request_info`	`dict`	Attributes từ `/attribute/request`	`data['request_info']['fileName']`
`sessions`	`list`	Danh sách các sessions	`data['sessions'][0]`
`sessions[i]['id']`	`str`	ID của session thứ i	`data['sessions'][0]['id']`
`sessions[i]['i']`	`np.ndarray`	In-phase samples (int32)	`data['sessions'][0]['i']`
`sessions[i]['q']`	`np.ndarray`	Quadrature samples (int32)	`data['sessions'][0]['q']`
`sessions[i]['iq']`	`np.ndarray`	Complex IQ = I + j*Q (complex128)	`data['sessions'][0]['iq']`

7. LƯU Ý QUAN TRỌNG

3. **Kiểu dữ liệu**:

- `i` và `q` là `int32` (số nguyên 32-bit)
- `iq` là `complex128` (số phức 128-bit, được tính từ I và Q)

4. **Kích thước**:

- Mỗi session có 512 samples (I và Q)
- File có 46,072 sessions

5. **Xử lý dữ liệu**:

- - Luôn kiểm tra `session['i']` và `session['q']` có `None` hoặc rỗng không trước khi sử dụng
- - Sử dụng `numpy` để xử lý các phép toán trên mảng

6. **Hiệu năng**:

- - Đọc toàn bộ file có thể mất thời gian (46,072 sessions)
 - - Có thể tối ưu bằng cách chỉ đọc một số sessions cần thiết
-

8. TÀI LIỆU THAM KHÁO

- **Code MATLAB gốc**: `read_iqtcp_h5_verge2.m`
 - **Code Python**: `reader_iqtcp_h5.py`
 - **File test**: `test_reader_iqtcp.py`
 - **HDF5 Documentation**: <https://www.hdfgroup.org/solutions/hdf5/>
-

Ngày tạo báo cáo: 2025-01-XX

Phiên bản reader: 1.0

Tương thích với: MATLAB `read_iqtcp_h5_verge2.m`