

```
MACAddress.css.cs
Assembly-CSharp
using System;
using System.Linq;
using UnityEngine;

Ссылка: 0
public class NewEmptyCSharpScript
{
    Ссылка: 3
    public class MACAddress
    {
        private readonly byte[] _addressBytes = new byte[6];
        private byte[] bytes;

        Ссылка: 1
        public MACAddress(byte[] bytes)
        {
            this.bytes = bytes;
        }

        Ссылка: 0
        public static MACAddress Parse(string address)
        {
            var parts = address.Split(':');
            return new MACAddress(parts.Select(p => Convert.ToByte(p, 16)).ToArray());
        }

        Ссылка: 0
        public bool IsBroadcast() => _addressBytes.All(b => b == 0xFF);
    }
}
```

```
EthernetFrame.cs
Assembly-CSharp
using System; // Для DateTime
using System.IO; // Для MemoryStream
using System.Linq; // Для LINQ
using UnityEngine;

Ссылка: 6
public class EthernetFrame
{
    public readonly byte[] Preamble = { 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA, 0xAA };
    Ссылка: 3
    public MACAddress DestinationAddress { get; }
    Ссылка: 3
    public byte[] Payload { get; }

    Ссылка: 1
    public EthernetFrame(MACAddress dest, MACAddress src, ushort type, byte[] payload)
    {
        DestinationAddress = dest ?? throw new ArgumentNullException(nameof(dest));
        Payload = payload ?? throw new ArgumentNullException(nameof(payload));
    }

    Ссылка: 0
    public byte[] Serialize()
    {
        using (var ms = new MemoryStream())
        using (var writer = new BinaryWriter(ms))
        {
            writer.Write(Preamble);
            writer.Write(DestinationAddress.ToByteArray());
            writer.Write(Payload);
            return ms.ToArray();
        }
    }
}
```

```
MACAddress.cs* X EnternetFrame.cs MACAddress.css.cs
Assembly-CSharp MACAddress

using System; // Добавлено для Convert
using System.Linq; // Добавлено для LINQ-методов
using UnityEngine; // Добавлено для Debug

Ссылка: 9
public class MACAddress
{
    private readonly byte[] _addressBytes = new byte[6]; // Хранит 6 байт адреса
    // Конструктор из массива байт
    Ссылка: 1
    public MACAddress(byte[] bytes)
    {
        if (bytes == null || bytes.Length != 6)
            throw new ArgumentException("MAC must be 6 bytes");
        Buffer.BlockCopy(bytes, 0, _addressBytes, 0, 6);
    }

    Ссылка: 0
    public static MACAddress Parse(string address)
    {
        try
        {
            var parts = address.Split(':', '-');
            if (parts.Length != 6)
                throw new FormatException("Invalid MAC format");

            var bytes = parts.Select(p => Convert.ToByte(p, 16)).ToArray();
            return new MACAddress(bytes);
        }
        catch (Exception ex)
        {
            Debug.LogError($"MAC parse error: {ex.Message}");
        }
    }
}
```

```
Ссылка: 0
public static MACAddress Parse(string address)
{
    try
    {
        var parts = address.Split(':', '-');
        if (parts.Length != 6)
            throw new FormatException("Invalid MAC format");

        var bytes = parts.Select(p => Convert.ToByte(p, 16)).ToArray();
        return new MACAddress(bytes);
    }
    catch (Exception ex)
    {
        Debug.LogError($"MAC parse error: {ex.Message}");
        throw;
    }
}

Ссылка: 1
public byte[] ToByteArray() => (_addressBytes).Clone();

Ссылка: 1
internal bool IsBroadcast()
{
    throw new NotImplementedException();
}
}
```

EthernetInterface.cs    MACAddress.cs\*    EthernetFrame.cs    MACAddress.css.cs

Assembly-CSharp    EthernetInterface    Address

```
using System;
using UnityEngine;

Скрипт Unity | Ссылка: 3
public class EthernetInterface : MonoBehaviour {
    Ссылка: 2
    public MACAddress Address { get; private set; }
    public float SpeedMbps = 100f;

    // Инициализация интерфейса
    Ссылка: 0
    public void Initialize(MACAddress address) {
        Address = address;
    }

    // Отправка кадра
    Ссылка: 0
    public void SendFrame(MACAddress dest, ushort type, byte[] payload) {
        var frame = new EthernetFrame(dest, Address, type, payload);
        EthernetSimulator.Instance.TransmitFrame(frame, this);
    }

    Ссылка: 1
    internal void ReceiveFrame(EthernetFrame frame)
    {
        throw new NotImplementedException();
    }
}
```

```

EthernetSimulator.cs  EthernetInterface.cs  MACAddress.cs*  EthernetFrame.cs  MACAddress.css.cs
Assembly-CSharp  EthernetSimulator  Instance

using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;

Скрипт Unity | Ссылки: 2
public class EthernetSimulator : MonoBehaviour
{
    public static EthernetSimulator Instance;
    public List<EthernetInterface> Interfaces = new();

    Сообщение Unity | Ссылки: 0
    void Awake() => Instance = this;

    // Передача кадра между устройствами
    Ссылки: 1
    public void TransmitFrame(EthernetFrame frame, EthernetInterface sender)
    {
        foreach (var iface in Interfaces.Where(i => i != sender))
        {
            StartCoroutine(DeliverFrame(frame, iface));
        }
    }

    Ссылки: 1
    IEnumerator DeliverFrame(EthernetFrame frame, EthernetInterface receiver)
    {
        yield return new WaitForSeconds(0.1f); // Задержка передачи
        receiver.ReceiveFrame(frame);
    }
}

```

```

EthernetVisualizer.cs  EthernetSimulator.cs  EthernetInterface.cs  MACAddress.cs*  EthernetFrame.cs
Assembly-CSharp  EthernetVisualizer  _framePrefab

using UnityEngine;

Скрипт Unity | Ссылки: 0
public class EthernetVisualizer : MonoBehaviour
{
    [SerializeField] private GameObject _framePrefab;
    [SerializeField] private float _animationSpeed = 5f;
    [SerializeField] private Color _unicastColor = Color.green;
    [SerializeField] private Color _broadcastColor = Color.red;

    Ссылки: 0
    public void VisualizeTransmission(EthernetFrame frame, Vector3 start, Vector3 end)
    {
        var frameObj = Instantiate(_framePrefab, start, Quaternion.identity);
        var visual = frameObj.GetComponent<FrameVisual>();

        visual.Setup(
            frame.DestinationAddress.IsBroadcast() ? _broadcastColor : _unicastColor,
            frame.Payload.Length
        );

        StartCoroutine(visual.AnimateMove(end, _animationSpeed));
    }
}

```

```
FrameVisual.cs  X  EthernetVisualizer.cs  EthernetSimulator.cs  EthernetInterface.cs  MACAddress.cs*
Assembly-CSharp  FrameVisual  _renderer

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Скрипт Unity | Ссылка: 1
public class FrameVisual : MonoBehaviour
{
    [SerializeField] private Renderer _renderer;
    [SerializeField] private TextMesh _sizeLabel;

    Ссылка: 1
    public void Setup(Color frameColor, int payloadSize)
    {
        _renderer.material.color = frameColor;
        _renderer.transform.localScale = Vector3.one * (0.5f + payloadSize / 1000f);
        _sizeLabel.text = $"{payloadSize} bytes";
    }

    Ссылка: 1
    public IEnumerator AnimateMove(Vector3 target, float speed)
    {
        while (Vector3.Distance(transform.position, target) > 0.1f)
        {
            transform.position = Vector3.MoveTowards(
                transform.position,
                target,
                speed * Time.deltaTime
            );
            yield return null;
        }
        Destroy(gameObject);
    }
}
```

100 % Проблемы не найдены. Стр: 1 Симв: 1 Пробе