**Viết chương trình kiểm tra tín hiệu mạng, lấy địa chỉ MAC wifi bằng C#**

- Đầu tiên các bạn cần imports thư viện vào:

using System.IO;

using System.Media;

- Sau đó các bạn viết một function scan sau: để get dữ liệu vào list box:

private void Scan()

{

string output;

string line;

int BSSIDNumber = 0;

int NetworkIndex = -1;

string[,] Networks = new string[100, 9];

listView1.SmallImageList = imageSignalLevel;

Process proc = new Process();

proc.StartInfo.CreateNoWindow = true;

proc.StartInfo.FileName = "netsh";

proc.StartInfo.Arguments = "wlan show networks mode=bssid";

proc.StartInfo.RedirectStandardOutput = true;

proc.StartInfo.UseShellExecute = false; // required for the Redirect setting above Process.Start(proc);

proc.Start();

output = proc.StandardOutput.ReadToEnd();

proc.WaitForExit();

StringReader sr = new StringReader(output.ToString());

line = null;

while ((line = sr.ReadLine()) != null)

{

if (line.StartsWith("General Failure"))

{

// Wifi disconnected or not installed

break;

}

if (line.StartsWith("SSID"))

{

NetworkIndex++;

for (int i = 0; i < 9; i++)

{

Networks[NetworkIndex, i] = " "; // prevent exception finding null on search

}

Networks[NetworkIndex, 3] = "0%"; // prevent exception for trim

BSSIDNumber = 0;// reset the BSSID number

Networks[NetworkIndex, 1] = line.Substring(line.IndexOf(":") + 1).TrimEnd(' ').TrimStart(' ');

continue;

}

if (line.IndexOf("Network type") > 0)

{

if (line.EndsWith("Infrastructure"))

{

Networks[NetworkIndex, 7] = "AP";

continue;

}

else

{

Networks[NetworkIndex, 7] = line.Substring(line.IndexOf(":") + 1); //"Ad-hoc";

}

}

if (line.IndexOf("Authentication") > 0)

{

Networks[NetworkIndex, 4] = line.Substring(line.IndexOf(":") + 1).TrimStart(' ').TrimEnd(' ');

continue;

}

if (line.IndexOf("Encryption") > 0)

{

Networks[NetworkIndex, 5] = line.Substring(line.IndexOf(":") + 1).TrimStart(' ').TrimEnd(' ');

continue;

}

if (line.IndexOf("BSSID") > 0)

{

if ((Convert.ToInt32(line.IndexOf("BSSID" + 6)) > BSSIDNumber))

{

BSSIDNumber = Convert.ToInt32(line.IndexOf("BSSID" + 6));

NetworkIndex++;

Networks[NetworkIndex, 1] = Networks[NetworkIndex - 1, 1]; // same SSID

Networks[NetworkIndex, 7] = Networks[NetworkIndex - 1, 7]; // same Network Type

Networks[NetworkIndex, 4] = Networks[NetworkIndex - 1, 4]; // Same authorization

Networks[NetworkIndex, 5] = Networks[NetworkIndex - 1, 5]; // same encryption

}

Networks[NetworkIndex, 0] = line.Substring(line.IndexOf(":") + 1);

continue;

}

if (line.IndexOf("Signal") > 0)

{

Networks[NetworkIndex, 3] = line.Substring(line.IndexOf(":") + 1);

continue;

}

if (line.IndexOf("Radio Type") > 0)

{

Networks[NetworkIndex, 6] = line.Substring(line.IndexOf(":") + 1);

continue;

}

if (line.IndexOf("Channel") > 0)

{

Networks[NetworkIndex, 2] = line.Substring(line.IndexOf(":") + 1);

continue;

}

if (line.IndexOf("Basic Rates") > 0)

{

//Networks[NetworkIndex, 8] = line.Substring(line.Length - 2, 2);

Networks[NetworkIndex, 8] = line.Substring(line.IndexOf(":"));

if (Networks[NetworkIndex, 8] == ":") { Networks[NetworkIndex, 8] = "not shown"; continue; }

Networks[NetworkIndex, 8] = Networks[NetworkIndex, 8].TrimStart(':').TrimStart(' ').TrimEnd(' ');

for (int i = Networks[NetworkIndex, 8].Length - 1; i > 0; i--)

{

if (Networks[NetworkIndex, 8].Substring(i, 1) == " ")

{

Networks[NetworkIndex, 8] = Networks[NetworkIndex, 8].Substring(i + 1, Networks[NetworkIndex, 8].Length - 1 - i);

break;

}

}

}

if (line.IndexOf("Other Rates") > 0)

{

// overwrite the basic rates if this entry is present

Networks[NetworkIndex, 8] = line.Substring(line.IndexOf(":"));

if (Networks[NetworkIndex, 8] == ":") { Networks[NetworkIndex, 8] = "not shown"; continue; }

Networks[NetworkIndex, 8] = Networks[NetworkIndex, 8].TrimStart(':').TrimStart(' ').TrimEnd(' ');

for (int i = Networks[NetworkIndex, 8].Length - 1; i >= 0; i--)

{

if (Networks[NetworkIndex, 8].Substring(i, 1) == " ")

{

Networks[NetworkIndex, 8] = Networks[NetworkIndex, 8].Substring(i + 1, Networks[NetworkIndex, 8].Length - 1 - i);

break;

}

}

}

}

for (int i = 0; i < listView1.Items.Count; i++)

{

// set signal to zero on all items in list

listView1.Items[i].SubItems[3].Text = "0%";

listView1.Items[i].ImageIndex = 5;

}

for (int i = 0; i < NetworkIndex + 1; i++)

{

ListViewItem SearchItem = new ListViewItem();

if (Networks[i, 0] == " ") continue; // don't search if no valid MAC Address !

SearchItem = listView1.FindItemWithText(Networks[i, 0]);

if (SearchItem == null)

{

// New discovery - add it to the list

SystemSounds.Hand.Play();

listView1.Items.Add(Networks[i,0]); // MAC Address

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 1]); // SSID

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 2]); // Channel

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 3]); // Signal

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 4]); // Authenticatiopn

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 5]); // Encryption

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 6]); // Radio Type

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 7]); // Network Type

listView1.Items[listView1.Items.Count-1].SubItems.Add( Networks[i, 8]); // Speed

int SignalInt = Convert.ToInt32(Networks[i, 3].TrimEnd(' ').TrimEnd('%'));

if (SignalInt > 50) listView1.Items[listView1.Items.Count - 1].ImageIndex = 0;

else if (SignalInt > 40) listView1.Items[listView1.Items.Count - 1].ImageIndex = 1;

else if (SignalInt > 30) listView1.Items[listView1.Items.Count - 1].ImageIndex = 2;

else if (SignalInt > 20) listView1.Items[listView1.Items.Count - 1].ImageIndex = 3;

else if (SignalInt > 0) listView1.Items[listView1.Items.Count - 1].ImageIndex = 4;

if ((Networks[i, 4] == "Open") & (Networks[i, 5] == "None")) listView1.Items[listView1.Items.Count - 1].BackColor = Color.PaleGreen;

listView1.Items[listView1.Items.Count - 1].EnsureVisible();

}

else

{

// Already in list - update Signal and other details that may change

listView1.Items[SearchItem.Index].SubItems[3].Text = Networks[i, 3]; // Signal

// Don't change any details if blank

if (Networks[i,1] != null) listView1.Items[SearchItem.Index].SubItems[1].Text = Networks[i, 1]; // SSID

if (Networks[i, 4] != null) listView1.Items[SearchItem.Index].SubItems[4].Text = Networks[i, 4]; // Authenticatiopn

if (Networks[i, 5] != null) listView1.Items[SearchItem.Index].SubItems[5].Text = Networks[i, 5]; // Encryption

if (Networks[i, 6] != null) listView1.Items[SearchItem.Index].SubItems[6].Text = Networks[i, 6]; // Radio Type

if (Networks[i, 7] != null) listView1.Items[SearchItem.Index].SubItems[7].Text = Networks[i, 7]; // Network Type

if (Networks[i, 8] != null) listView1.Items[SearchItem.Index].SubItems[8].Text = Networks[i, 8]; // Speed

int SignalInt = Convert.ToInt32(Networks[i, 3].TrimEnd(' ').TrimEnd('%'));

if (SignalInt > 50) listView1.Items[SearchItem.Index].ImageIndex = 0;

else if (SignalInt > 40) listView1.Items[SearchItem.Index].ImageIndex = 1;

else if (SignalInt > 30) listView1.Items[SearchItem.Index].ImageIndex = 2;

else if (SignalInt > 20) listView1.Items[SearchItem.Index].ImageIndex = 3;

else if (SignalInt > 0 ) listView1.Items[SearchItem.Index].ImageIndex = 4;

else if (SignalInt == 0) listView1.Items[SearchItem.Index].ImageIndex = 5;

}

}

}

- Sau đó các bạn gọi hàm Scan(), khi nhấn vào button dò tìm:

private void btnScan\_Click(object sender, EventArgs e)

{

Scan();

}