ELM327\_Bluetooth\_C#6\_Dauntless

Now I know that my device gives multiple line answers. The first dump, the self-reporting not seem to happening until I send something on the serial port. How can I modify my code to send an 'empty' character just to receive the multi-line self-reporting dump and then I could send my commands to the device?

using (StreamWriter sw = new StreamWriter(responseFile, false))

{

// Send an 'empty' command (just a carriage return) to trigger the device's self-reporting

Console.WriteLine("Küldés: üres parancs az eszköz inicializálásához...");

serialPort.Write("\r");

// Read and collect all lines from the self-reporting dump

var initialDump = new List<string>();

DateTime start = DateTime.Now;

while ((DateTime.Now - start).TotalMilliseconds < serialPort.ReadTimeout)

{

try

{

string line = serialPort.ReadLine();

if (string.IsNullOrWhiteSpace(line))

break;

initialDump.Add(line);

}

catch (TimeoutException)

{

break;

}

}

// Optionally log or save the initial dump

Console.WriteLine("<<< Eszköz önriport:");

foreach (var line in initialDump)

{

Console.WriteLine(line);

sw.WriteLine($"[INIT] {line}");

}

// Now continue with your command loop as before

foreach (string parancs in atParancsok)

{

if (string.IsNullOrWhiteSpace(parancs))

continue;

serialPort.Write(parancs + "\r");

Console.WriteLine($">>> Küldve: {parancs}");

try

{

var responseLines = new List<string>();

DateTime cmdStart = DateTime.Now;

while ((DateTime.Now - cmdStart).TotalMilliseconds < serialPort.ReadTimeout)

{

string line = serialPort.ReadLine();

if (string.IsNullOrWhiteSpace(line))

break;

responseLines.Add(line);

}

string valasz = string.Join(" | ", responseLines);

Console.WriteLine($"<<< Válasz: {valasz}");

sw.WriteLine($"{parancs} -> {valasz}");

}

catch (TimeoutException)

{

Console.WriteLine("<<< Válasz timeout.");

Console.ReadKey();

sw.WriteLine($"{parancs} -> (timeout)");

}

System.Threading.Thread.Sleep(1000);

}

}

You got this error because your device only sends its self-reporting dump after receiving something (even an empty command) on the serial port.

This might be happening because the Dauntless OBD expects a carriage return (**\r**) to trigger its initial multi-line response, and your code was not sending anything before reading.

Here's how I fixed the code:  
I added a line to send an 'empty' command (**serialPort.Write("\r");**) right after opening the port and before your command loop. Then, I read and collect all lines from the device's self-reporting dump, logging or saving them as needed. After that, your command loop continues as before. This ensures you capture the initial multi-line response before sending further commands.