

Paweł Twardawa 235072

Data: 6.11.2018 r.

Aleksandra Wieczorkiewicz 234980

Urządzenia peryferyjne

Ćwiczenie 7 GPS

Prowadzący:
Dr inż. Jacek Mazurkiewicz

1. Wstęp

Celem zadania było zapoznanie się z obsługą GPS, oraz zrozumienie w jaki sposób pozyskuje dane o pozycji geograficznej. Program został napisany w języku C# wersji okienkowej.

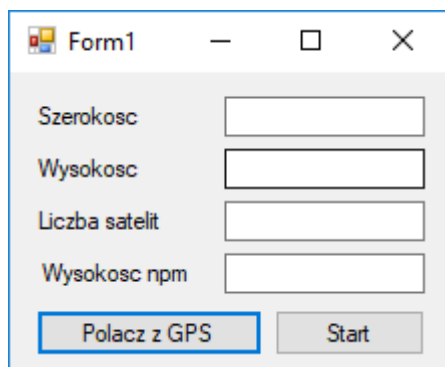
2. Zadania do wykonania

- 2.1. Zapoznać się z zestawem GPS oraz podłączyć via Bluetooth
- 2.2. W ramach testu podłączyć GPS na ustawieniach testowych oraz utworzyć połączenie przy użyciu HyperTerminala.
- 2.3. Odczytać uzyskane komendy oraz podzielić je wg typów wiadomości.
- 2.4. Sprawdzić ważność uzyskanych danych i przedyskutować wynik.
- 2.5. Napisać program w dowolnym środowisku obiektowym, który będzie obsługiwał transmisję szeregową oraz pozwoli na czytelne przedstawienie uzyskanych danych.
- 2.6. Napisać program, który na podstawie samodzielnie uzyskanych danych lub od prowadzącego (plik tekstowy, format NMEA) zlokalizuje na mapie świata (np. z Google Map) punkty, w których znajdowało się urządzenie.

3. Opis programu

3.1. Wygląd aplikacji

Aplikacja wyświetla podstawowe informacje tj. wysokość, szerokość geograficzna, liczba satelit od których odbiera sygnał oraz wysokość nad poziomem morza. Aby rozpocząć pracę aplikacji należy połączyć się z GPS (lewym dolnym przyciskiem). Po poprawnym nawiązaniu komunikacji należy nacisnąć przycisk Start który rozpoczyna odczytywanie danych z GPS.



3.2. Komunikacja z GPS

Do komunikacji z modułem GPS został użyty port szeregowy. Na Listingu 1 przedstawione jest stworzenie portu wraz z podaniem odpowiednich parametrów.

Listing 1.

```
SerialPort port = new SerialPort("COM9", 9600, Parity.None, 8, StopBits.One);
```

3.3. Otwieranie połączenia

Po naciśnięciu przycisku „Połącz z GPS” zostaje wywołana metoda `polacz_Click(object, EventArgs)`. Metoda rozpoczyna próbę otwarcia portu. Jeżeli port zostanie poprawnie otwarty, na ekranie wyświetli się komunikat „port otwarty”, w przeciwnym razie komunikat będzie zawierał informację na temat błędu.

Listing 2.

```
private void polacz_Click(object sender, EventArgs e)
{
    try
    {
        port.Open();
        if(port.IsOpen)
        {
            MessageBox.Show("port otwarty");
        }
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

3.4. Odczytywanie danych

Po wciśnięciu przycisku Start zostanie wywołana metoda start_Click(objct, EventArgs). Metoda ta odpowiada za wystartowanie Timera, dzięki któremu wyświetlana lokalizacja jest zawsze aktualna, ponieważ działający w tle timer stale aktualizuje dane o lokalizacji. W metodzie timer1_Tick(object, EventArgs) odczytywane są dane z portu szeregowego następnie przetwarzane i wyświetlane w zrozumiałym formacie.

Listing 4.

```
private void button2_Click(object sender, EventArgs e)
{
    if (port.IsOpen)
    {
        if (!timer1.Enabled)
        {
            timer1.Enabled = true;
            timer1.Start();
            button2.Text = "Stop";
        }
        else
        {
            timer1.Enabled = false;
            timer1.Stop();
            button2.Text = "Start";
        }
    }
    else
    {
        MessageBox.Show("Port nie zostal otwarty");
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    String st = ((char)176).ToString();

    String data;
    if (port.IsOpen)
    {
        data = port.ReadExisting();
    }
}
```

```

String[] dataArr = data.Split('$');

for (int i = 0; i < dataArr.Length; i++)
{
    String tmp = dataArr[i];

    String[] tmpData = tmp.Split(',');

    if (tmpData[0] == "GPGGA" && tmpData[2] != "" && tmpData[4] != "")
    {
        String[] sze = tmpData[2].Split('.');
        String[] wys = tmpData[4].Split('.');

        textBox1.Text = (sze[0].Length == 5 ? sze[0].Substring(0, 3) :
sze[0].Substring(0, 2)) + st + (sze[0].Length == 5 ? sze[0].Substring(3, 2) :
sze[0].Substring(2, 2)) + "\" + sze[1].Substring(0, 2) + \",\" + sze[1].Substring(2, 2)
+ "\" + tmpData[3];
        textBox2.Text = (wys[0].Length == 5 ? wys[0].Substring(0, 3) :
wys[0].Substring(0, 2)) + st + (wys[0].Length == 5 ? wys[0].Substring(3, 2) :
wys[0].Substring(2, 2)) + "\" + wys[1].Substring(0, 2) + \",\" + wys[1].Substring(2, 2)
+ "\" + tmpData[5];

        satelity.Text = tmpData[7];
        wysokosc.Text = tmpData[9];
    }
}
}
}

```

4. Wnioski

GPS jest bardzo przydatnym wynalazkiem który w dzisiejszych czasach jest implementowany w wielu urządzeniach tj. smartfony, samochody, rowery itp. Zaletą GPS jest możliwość zwrócenia danych geograficznych z dokładnością do kilku metrów. Dane zwracane przez GPS nie są trudne do zinterpretowania oraz wyświetlenia ich w zrozumiały sposób.