

Mateusz Krawczak 241318

Karol Jaskółka 241306

Grupa: Pon P 17:00

Data wykonania ćwiczenia: 13.01.2020

# Urządzenia Peryferyjne

## Ćwiczenie 7 – GPS

# 1. Wstęp

GPS ( Global Positioning System ) -jest systemem nawigacji satelitarnej, który został stworzony przez Departament Obrony Stanów Zjednoczonych. System ten obejmuje swoim zasięgiem całą kulę ziemską. Działanie GPS polega na mierzeniu czasu dotarcia sygnału radiowego z satelitów do odbiornika. Znając prędkość fali elektromagnetycznej oraz dokładny czas wysłania danego sygnału, można obliczyć odległość odbiornika od satelitów. Sygnał dociera do użytkownika na dwóch możliwych częstotliwościach nośnych:

- L1 = 1575,42 MHz (długość fali 19,029 cm)
- L2 = 1227,6 MHz (długość fali 24,421 cm)

Identyfikacja satelitów oparta jest na metodzie podziału kodu CDMA (Code Division Multiple Access). Oznacza to, że wszystkie satelity emitują na tych samych częstotliwościach, jednak nadawane sygnały są modulowane innymi kodami. Aby określić pozycję w przestrzeni i czasie, konieczny jest jednoczesny odbiór z co najmniej czterech satelitów. Odbiornik oblicza trzy pseudo-odległości do satelitów oraz odchyłki czasu (różnicy między mało dokładnym dokładnym wzorcem kwarcowym zainstalowanym na odbiorniku oraz bardzo precyzyjnym zegarem atomowym na satelicie). Satelita transmituje w depeszy nawigacyjnej m.in. czas, almanach (stan konstelacji satelitów) oraz efemerydy (parametry lotu satelity). Dzięki tym danym odbiornik GPS jest w stanie określić dokładne współrzędne satelity, w momencie nadania sygnału, co przekłada się następnie na ,przy pomocy pseudo-odległości, na obliczenie własnej pozycji.

Tab. 3. Zawartość wiadomości GGA			
Numer pola	Nazwa	Przykład	Format/Opis
1	ID wiadomości	\$GPGGA	nagłówek wiadomości GGA
2	Czas UTC	092842.094	hhmmss.sss – godziny, minuty, sekundy, ułamkowe części sekundy
3	Szerokość geograficzna	5215.2078	ddmm.mmmm – stopnie, minuty, ułamkowe części minuty
4	Wskaźnik półkuli N/S	N	N – północna S – południowa
5	Długość geograficzna	02054.3681	dddmm.mmmm – stopnie, minuty, ułamkowe części minuty
6	Wskaźnik półkuli E/W	E	E – wschodnia W – zachodnia
7	Wskaźnik rodzaju rozwiązania nawigacyjnego	1	0 – rozwiązanie niedostępne lub niepoprawne 1 – rozwiązanie dostępne (SPS) 2 – rozwiązanie dostępne (SPS) z wykorzystaniem poprawek DGPS 3 – rozwiązanie dostępne (PPS)
8	liczba użytych satelitów	06	liczba satelitów użytych podczas pozycjonowania (w badanym odbiorniku liczba z zakresu 0–12)
9	HDOP	1.7	współczynnik „rozmycia” dokładności położenia w płaszczyźnie poziomej (horyzontalnej)
10	Wysokość MSL	138.5	wysokość nad średnim poziomem morza MSL (w badanej wersji odbiornika nie zaimplementowano korekcji geoidy i jest to w rzeczywistości wysokość nad ziemską elipsoidą odniesienia)
11	Jednostki	M	jednostki, w których wyraża się wysokość (metry)
12	Separacja geoidy		pole puste ze względu na brak korekcji geoidy w tym modelu odbiornika
13	Jednostki		j.w.
14	Wiek poprawek różnicowych		pole puste ze względu na niedostępność poprawek różnicowych DGPS podczas badania odbiornika
15	ID stacji ref. DGPS	0000	numer identyfikacyjny stacji DGPS (same zera ze względu na niedostępność poprawek DGPS podczas badania odbiornika)
16	Suma kontrolna	09	suma XOR wszystkich bajtów pomiędzy „\$” a „*”

## 2. Program

Form1

Połącz

Odczyt

Pokaż na mapie

Status : Połączono

\$GPGGA,180521.000,5106.5130,N,01703.6415,E,0.0,,107.4,M,42.6,M,,\*48

Czas GMT0

18:05:21

Szerokosc

51.108550

Dlugosc

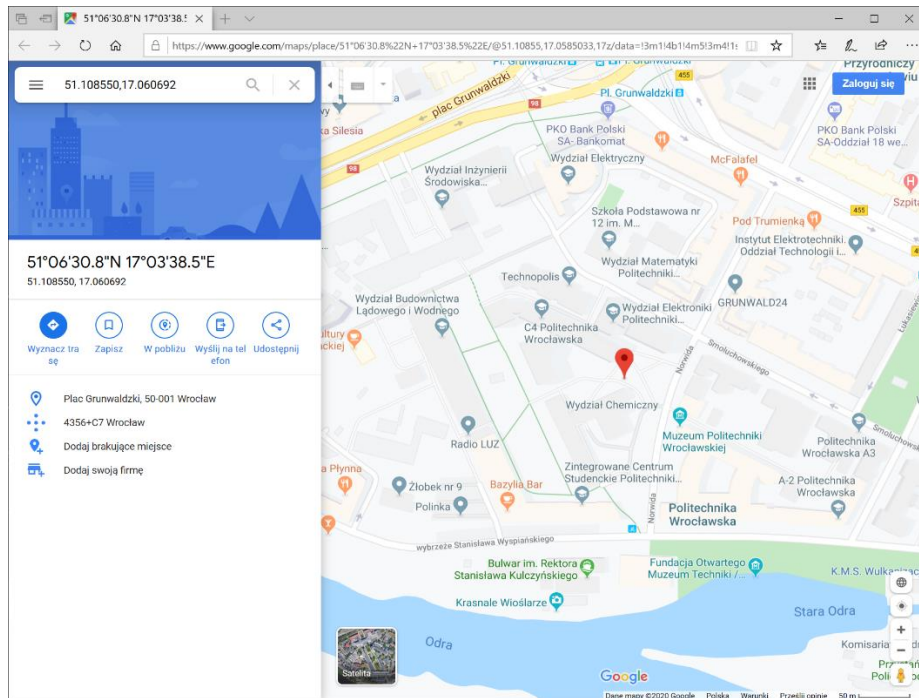
17.060692

Wysokosc

107.4 m

Satelitey

0



Cały program jest dostępny pod tym linkiem:

<https://github.com/matson19/UP/tree/master/Lab%206%20-%20gps>

```
public partial class Form1 : Form
{
    static SerialPort serialPort;
    string outputData;
    string latitude;
    string longitude;
    bool connected = false;
    char[] receivedData = new char[512];

    public Form1()
    {
        Thread.CurrentThread.CurrentCulture =
CultureInfo.CreateSpecificCulture("en-GB");
        InitializeComponent();
        serialPort = new SerialPort();
        textBoxMessage.Text = "";
    }

    private void buttonConnect_Click(object sender, EventArgs e)
    {
        if (!connected)
        {
            serialPort.PortName = "COM8";
            serialPort.BaudRate = 9600;
            serialPort.Open();
            connected = true;
            labelConnectionStatus.Text = "Status : Połączono";
            labelConnectionStatus.ForeColor = Color.Green;
            ReadData();
        }
    }
}
```

```

private void buttonRead_Click(object sender, EventArgs e)
{
    ClearData();
    if (connected)
    {
        ReadData();
    }
}

private void buttonShow_Click(object sender, EventArgs e)
{
    try
    {
        StringBuilder queryAddress = new StringBuilder();
        queryAddress.Append("http://maps.google.com/maps?q=");

        if (latitude != string.Empty)
        {
            queryAddress.Append(latitude + "%2C");
        }

        if (longitude != string.Empty)
        {
            queryAddress.Append(longitude);
        }

        System.Diagnostics.Process.Start(queryAddress.ToString());
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), "Error");
    }
}

private void ReadData()
{
    outputData = serialPort.ReadExisting();

    var splitedData = outputData.Split('$');

    foreach (var line in splitedData)
    {
        try
        {
            if (line.Contains("GPGGA"))
            {
                string fLatitude = "";
                string fLongitude = "";

                var info = line.Split(',');

                double longdec = double.Parse(info[4],
CultureInfo.InvariantCulture) / 100.0;
                double latdec = double.Parse(info[2],
CultureInfo.InvariantCulture) / 100.0;

                if (info[3] == "S")
                {
                    fLatitude = "-";
                }
            }
        }
    }
}

```

```

    }
    if (info[5] == "W")
    {
        fLongitude = "-";
    }

    var latSplitted = Convert.ToString(latdec).Split('.');
    var longSplitted = Convert.ToString(longdec).Split('.');

    longdec = Convert.ToDouble("0." + longSplitted[1],
CultureInfo.InvariantCulture) * 10 / 6;
    latdec = Convert.ToDouble("0." + latSplitted[1],
CultureInfo.InvariantCulture) * 10 / 6;

    textBoxLatitude.Text = fLatitude +
(Convert.ToDouble(latSplitted[0]) + latdec).ToString("F6");
    textBoxLongitude.Text = fLongitude +
(Convert.ToDouble(longSplitted[0]) + longdec).ToString("F6");

    latitude = fLatitude + (Convert.ToDouble(latSplitted[0]) +
latdec).ToString("F6");
    longitude = fLongitude + (Convert.ToDouble(longSplitted[0]) +
longdec).ToString("F6");

    textBoxTime.Text = info[1].Substring(0, 2) + ":" +
info[1].Substring(2,2) + ":" + info[1].Substring(4,2);

    textBoxMessage.Text += "$" + line;
    textBoxHigh.Text = info[9] + " m";
    textBoxSatellites.Text = info[7];

    }
}
catch (Exception)
{
    Console.WriteLine("ERROR");
}
}
}

private void ClearData()
{
    textBoxLatitude.Text = "";
    textBoxLongitude.Text = "";
    textBoxMessage.Text += "";
    textBoxHigh.Text = "";
    textBoxSatellites.Text = "";
}
}
}

```

### 3. Wnioski

Napisany przez nas na zajęciach program pozwolił nam na zapoznanie się z możliwościami i sposobami obsługi urządzeń GPS na komputerze. Poznaliśmy również działanie protokołu NMEA, powszechnie stosowanym w komunikacji urządzeń elektronicznych. W dalszych krokach ćwiczenia odkodowaliśmy dane wysyłane przez urządzenie GPS na długość i szerokość

geograficzną, a także aktualny czas. Udało się również poprawnie sformatować dane, aby wyświetlić aktualną pozycję GPS na Mapach Google.