

Szymon KONDRATOWICZ<sup>1</sup>  
Bartłomiej OSZCZAK<sup>2</sup>

### **WYKORZYSTANIE JĘZYKA PROGRAMOWANIA PYTHON DO BEZPOŚREDNIEJ TELETRANSMISJI DANYCH Z ODBIORNIKA GPS NA ZDALNY SERWER FTP**

*Celem projektu było stworzenie systemu komunikacji odbiornika GPS z serwerem FTP poprzez GPRS. W tym celu wykorzystano programowalny odbiornik GPS firmy Telit. Do odbiornika został wgrany skrypt, którego to skompilowaną wersję odbiornik uruchamia po każdorazowym włączeniu. Zadaniem skryptu było przestanie za pomocą komunikatów NMEA danych nawigacyjnych odbiornika. Plik z danymi w czasie rzeczywistym jest tworzony na wybranym serwerze FTP. Następnym etapem jest odpowiednie sformatowanie odebranego pliku i umieszczenie go w tabeli bazy danych (stworzonej na serwerze). Wykonuje to aplikacja uruchamiana w dowolnej przeglądarce internetowej. Po tym etapie użytkownik może wyświetlić aktualne dane nawigacyjne odbiornika w postaci strony internetowej, wybierając interesujące go charakterystyki. Stworzona została również aplikacja dokonująca konwersji wyników pomiarów na format CSV. Umożliwia ona także wybór z pliku potrzebnych danych.*

### **THE USE OF PYTHON PROGRAMMING LANGUAGE FOR DIRECT TELETRANSMISSION OF DATA FROM GPS RECEIVER TO FTP SERVER**

*The main goal of the experiment was to create a communication system between a GPS receiver and FTP server (via GPRS). For this purpose, there was used the programmable GPS receiver Telit, to which the Python script was uploaded, and its compiled version was run by the receiver on each start. The main task of the script was to send data, using NMEA sentences, to the file made in the real time on the selected FTP server. In the next step, the received file was appropriately formatted and transferred to the database on the server, which may be performed by any internet browser. After this stage, the user can see up-to-date navigational data on a website and choose the needed angle of characteristics. Software was also made for choosing and converting data to the CSV format, which helps to find the needed information in a file.*

---

<sup>1</sup> OPEGIEKA Sp z. o. o. e-mail: szymonk@opegieka.pl

<sup>2</sup> Uniwersytet Warmińsko-Mazurski w Olsztynie, Wydział Geodezji i Gospodarki Przestrzennej, Katedra Geodezji Satelitarnej i Nawigacji, ul Heweliusza 5, 10-957 Olsztyn, tel: + 48 89 523-34-81, Fax: + 48 89 523-47-23, email: bartlomiej.oszczak@kgsin.pl, www.kgsin.pl

## 1. WSTĘP

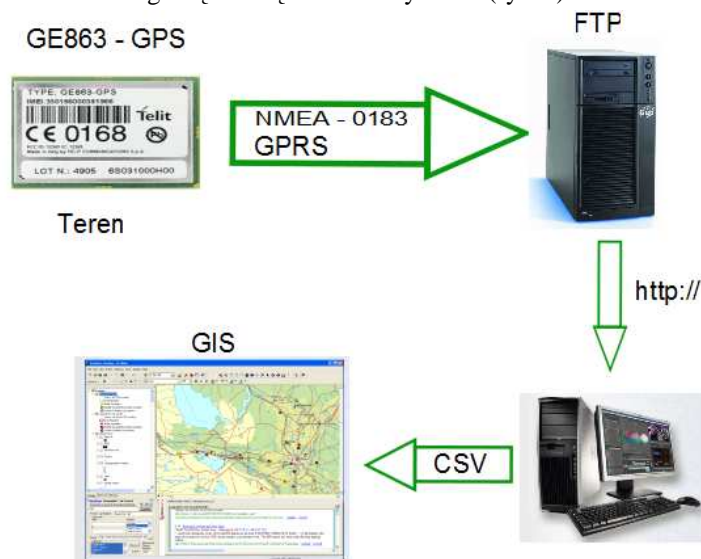
Zrealizowany system może być podstawą mobilnego systemu GIS. Pierwszym etapem działania systemu jest przesyłanie danych nawigacyjnych przez GPRS [1] z terenu na serwer FTP. Przesłany plik zawiera w swojej treści sentencje NMEA z zaprogramowanego mikroprocesora modemu GSM/GPS firmy Telit. Stworzono bazę danych mającą na celu gromadzenie rejestrowanych pomiarów z interwałem jednosekundowym. Wynikami pomiarów są wybrane elementy sentencji NMEA. Jej poszczególne składowe to nazwy kolumn tabeli. Użytkownik może wyświetlić wyniki pomiarów w przeglądarce internetowej (łączącej się z bazą danych) za pomocą GUI. Użytkownik może obejrzeć wyniki z dowolnego komputera, (podłączonego do sieci internet), w dowolnym miejscu. W związku z tym system jest mobilny po dwóch stronach: urządzenia pomiarowego i odbiorcy danych. Użytkownik może również wybrać pary interesujących go elementów sentencji NMEA, np. czas i liczbę satelitów, lub współczynnik hdop.

Dodatkową funkcjonalnością jest konwersja odebranego pliku na plik w formacie CSV. Plik w takim formacie obsługiwany jest przez programy GISowe. Celem przerobienia pliku przesłanego na serwer na plik CSV jest zlokalizowanie odbiornika Telit. Docelowo można bowiem wgrać plik wyjściowy z programu do programu z rodziny GIS (np. ArcGIS) i zlokalizować aktualne położenie odbiornika. Program oferuje także generowanie raportów z surowych danych przesłanych przez odbiornik. Raport może zawierać wybrane przez użytkownika elementy sentencji NMEA.

Do komunikacji wykorzystano programy RsTerm i CDM. Wgranie skryptu (wykonywanego po stronie odbiornika) wymagało ustawienie odpowiedniej prędkości połączenia i portu COM.

## 2. DIAGRAM DZIAŁANIA SYSTEMU

Poniżej zaprezentowano ogólną zasadę działania systemu (rys.1.):



Rys. 1. Zasada działania zbudowanego systemu

## 2.1 PROGRAMOWALNY ODBIORNIK GPS (Telit GE863)

W systemie jako zestaw badawczy wykorzystano płytę główną [2] (zestaw badawczy-rys.2.), do której za pomocą adapterów można podłączyć każdy moduł firmy Telit. Wykorzystanym modulem w projekcie jest GSM/GPRS GE863-GPS [3]. Według specyfikacji producenta odbiornik wbudowany GPS ma dokładność poniżej 2.5m. Można podłączyć do EVK dwuczęstotliwościowy odbiornik geodezyjny za pomocą wejścia RS232 wtedy dokładność będzie adekwatnie wyższa. Wykorzystany moduł obsługuje serwisy WAAS, EGNOS i umożliwia pozycjonowania wewnątrz budynków. Istotną składową modułu jest interpreter języka Python (umożliwiający wgranie dowolnej aplikacji wykonywanej po stronie modułu). Moduł zawiera: silnik interpretera skryptów Python'a, około 3Mb pamięci NVM na skrypty użytkownika, około 1.5 Mb pamięci na użytek interpretera skryptów [4,5,6].



Rys. 2. Zestaw badawczy do testów+moduł Telit 862-GPS

## 3. SKŁADOWE SYSTEMU

Stworzony system nawiązuje komunikację: komputer PC - moduł GSM/GPRS GE863-GPS (przez aplikację Rsterm i CDM 2.04.14). Następnie wykorzystany jest skrypt napisany w języku Python [4,5,6], wgrany do modułu. Po włączeniu EVK w terenie skrypt zostaje automatycznie uruchomiony. Określa on pozycję (można wybrać też inne parametry nawigacyjne np.: prędkość; w zależności od implementacji interfejsu GPS) w układzie WGS84. Zapisuje następnie parametry do pliku (co określony interwał czasu), kolejno otwiera połączenie z wybranym serwerem ftp, przesyła plik z parametrami nawigacyjnymi, zamyka połączenie ftp, kasuje plik z pamięci.

Poniżej znajduje się fragment kodu źródłowego dla odbiornika GPS w języku Python:

```
import MDM
import MOD
import SER
import sys
import GPS
```

```

SER.set_speed('115200','8N1')
net1 = MDM.send('AT+CGDCONT=1,"IP","www.plusgsm.pl", 0) #podaje nazwe sieci
net2 = MDM.sendbyte(0x0d, 0) #znak <cr><lf>
net3 = MDM.receive(20)
MOD.sleep(200)
net4 = MDM.send('AT#GPRS=1', 0) #podaje nazwe sieci plusgsm.pl
net5= MDM.sendbyte(0x0d, 0) #znak <cr>
net6= MDM.receive(20)
# print "Jestem w GPRS: %s"%net6
MOD.sleep(30)
res=MDM.send('AT#FTPOPEN="nazwa.serwer.pl","login","password",0\r',0)
res=MDM.receive(100)
res=MDM.send('AT#FTPPUT="try1_1.txt"\r',0)
res=MDM.receive(100)
def printPosition():
    i = 0
    while i<50:
        poz=GPS.getActualPosition()
        pozycja = poz+'\n'
        SER.send(pozycja)
        res=MDM.send(pozycja,0)
        res=MDM.receive(20)
        i=i+1
printPosition()
res=MDM.send('+++ ',100)
res=MDM.receive(100)
res=MDM.send('AT#FTPCLOSE\r',0)
res=MDM.receive(20)

```

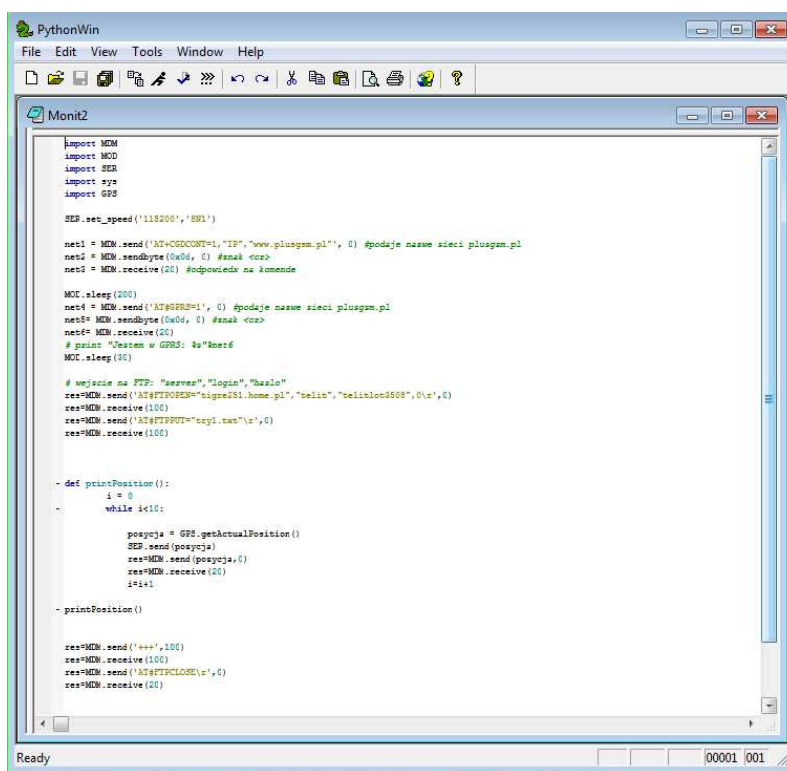
W pierwszej kolejności skrypt odbiornika GPS uruchamia wbudowane biblioteki (interfejsy urządzenia). Biblioteka MDM odpowiada za sterowanie i konfigurację komendami modemowymi AT. Interfejs MOD służy do sterowania różnymi procesami np. posiada możliwość zarządzania urządzeniem na podstawie aktualnego czasu GPS, np. potrafi zatrzymać wykonywanie skryptu na określony moment (przykładowo można zatrzymać skrypt aby w pierwszej kolejności odebrać odpowiedź na zapytanie modemu). Interfejs SER odpowiada za sterowanie i konfigurację portu szeregowego RS-232. Interfejs SYS odpowiada aplikacji za debugowanie i wyświetlanie aktywnych procesów programu. Biblioteka GPS aktywuje i odpowiada za konfigurację odbiornika GPS.

Po ustaleniu prędkości portu szeregowego następuje ustawienie prędkości portu komunikacyjnego RS232. Następnie urządzenie działa w ten sposób, że jego modem czeka na pobranie dynamicznego adresu internetowego IP i po jego uzyskaniu odbiornik GPS otwiera połączenie FTP ze zdalnym serwerem. W następnej kolejności urządzenie 863-GPS zakłada plik zdalnie na serwerze FTP i wykonywana jest funkcja Pythona odpowiedzialna za wysyłanie wybranych sentencji formatu NMEA w każdej sekundzie na port komunikacyjny. Dane GPS z wewnętrznego portu zapisywane są bezpośrednio do pliku

zdalnego FTP. W następnym etapie zamykane jest połączenie z serwerem FTP i utworzony plik wykorzystywany jest w aplikacji przeglądarki internetowej.

Skrypt napisany w języku Python, wgrany do modułu (rys.3.). Po włączeniu EVK w terenie skrypt zostaje automatycznie uruchomiony. Skrypt działa następująco:

- określa pozycję (można wybrać też inne parametry nawigacyjne np.: prędkość; w zależności od implementacji interfejsu GPS) w układzie WGS84,
- zapisuje parametry do pliku (co określony interwał czasu),
- otwiera połączenie z wybranym serwerem ftp,
- przesyła plik z parametrami nawigacyjnymi,
- zamyka połączenie ftp,
- kasuje plik z pamięci.



```
PythonWin
File Edit View Tools Window Help

Monit2

import MCM
import MCD
import SEP
import SYS
import GPS

SEP.set_speed('115200','8N1')

net1 = MCM.send('AT+CGDCONT=1,"IP","www.plusgem.pl",0) #podaje nazwe sieci plusgem.pl
net2 = MCM.sendbyte(0x00,0) #znak <cr>
net3 = MCM.receive(20) #odpowiedz na komende

MCD.sleep(200)
net4 = MCM.send('AT+SPRS=1',0) #podaje nazwe sieci plusgem.pl
net5 = MCM.sendbyte(0x00,0) #znak <cr>
net6 = MCM.receive(20)
# print "Wzrost w GPS: %s"%net6
MCD.sleep(50)

# wejście na FTP: "server","login","haslo"
res=MCM.send('AT#FTTPSER="lugre551.home.pl","celis","celisloc508",0/0',0)
res=MCM.receive(100)
res=MCM.send('AT#FTTPUF="wyj1.txt"\r',0)
res=MCM.receive(100)

- def printPosition():
    i = 0
    while i<10:

        pozycja = GPS.getActualPosition()
        SEP.send(pozycja)
        res=MCM.send(pozycja,0)
        res=MCM.receive(20)
        i+=1

- printPosition()

res=MCM.send('+++',100)
res=MCM.receive(100)
res=MCM.send('AT#FTPCLOSE\r',0)
res=MCM.receive(20)

Ready 00001 001
```

Rys. 3 Środowisko PythonWin, w którym napisano i skompilowano skrypt

W rezultacie, bezpośrednio z obornika GPS na serwer FTP wysyłane i zapisywane są dane o następującej treści (fragment):

```
5345.6370N,02027.4624E,0.8,118.7,3,85.73,0.32,0.17,140410,09113612.000,5345.637
0N,02027.4625E,0.8,118.8,3,109.57,0.32,0.17,140410,09113614.000,5345.6368N,0202
7.4626E,0.8,118.8,3,350.22,0.18,0.09,140410,09113616.000,5345.6368N,02027.4626E
```

,0.8,118.8,3,342.87,0.25,0.13,140410,09113618.000,5345.6367N,02027.4627E,0.8,119.0,3,19.58,0.14,0.07,140410

Funkcja `GPS.getActualPosition()` zawarta w pętli jest odpowiedzialna za pobranie danych nawigacyjnych. Jest odpowiednikiem komendy NMEA `AT$GPSACP` (komendy modemowej). W wyniku wykorzystania takiej funkcji pobierana jest linijka danych:

113612.000,5345.6370N,02027.4625E,0.8,118.8,3,109.57,0.32,0.17,140410,09  
gdzie:

- 113612.000 czas UTC,
- 5345.6370N szerokość geograficzna,
- 02027.4625E długość geograficzna,
- 0.8 hdop (*ang. Horizontal Dilution of Precision*),
- 118.8 wysokość odniesiona do geoidy [metry],
- 3 3D fix (możliwe wartości: 1 Invalid fix 2 2D fix 3 3D fix),
- 109.57 cog (*ang. course over ground*) [stopnie],
- 0.32 spkm (*ang. speed over ground*) [km/h],
- 0.17 spkn (*ang. speed over ground*) [węzły],
- 140410 data,
- 09 liczba satelitów.

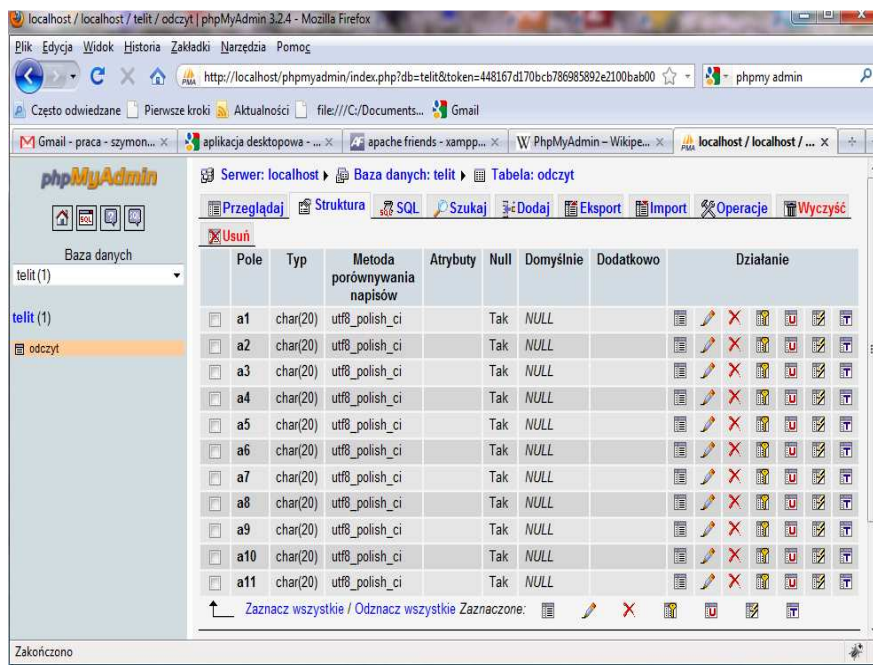
Skrypt po stronie serwera (php) działa w następujący sposób:

- dokonuje zapisu danych nawigacyjnych do bazy danych mysql
- wyświetla je w czytelnej postaci tabeli w stronie internetowej
- oferuje możliwość wyboru par parametrów

Aplikacja napisana w języku C++. Uruchamiana na komputerze PC. Co robi:

- Dokonuje formatowania pliku wysłanego przez moduł, do postaci pliku \*.csv, obsługiwanego przez większość platform GIS'owych (np. ArcGIS, Geomedia, czy też opensourc'owy: QuantumGIS)

Przed uruchomieniem skryptu konieczne było stworzenie bazy danych na użytek aplikacji w tym celu użyto narzędzia do zarządzania bazą danych MySQL – *phpMyAdmin* (rys.4):



Rys. 4. Struktura tabeli w bazie danych MySQL

Została ( w języku SQL ) stworzona tabela w bazie danych o kolumnach o nazwach: a1, a2, a3, a4, a5, a6, a7, a8, a8, a9, a10, a11 oznaczających dane przesłane przez odbiornik GPS. A więc odpowiednio: czas UTC, szerokość geograficzna, długość geograficzna, hdop (ang. *horizontal dilution of precision*), wysokość odniesiona do geoidy [metry], fix (możliwe wartości: 1 Invalid fix 2 2D fix 3 3D fix), cog (ang. *course over ground*) [stopnie], spkm (ang. *speed over ground*) [km/h], spkn (ang. *Speer over ground*) [węzły], data, liczba satelitów.

Kolejnym etapem jest uruchomienie skryptu do zgrania danych z pliku do tabeli bazy danych. Dokonuje się to poprzez uruchomienie pliku *doBazy.php* w przeglądarce internetowej.

Następnie uruchamiając plik napisany w języku HTML mamy do wyboru sposób wyświetlania wyników pomiaru (rys.5,6):

### Wybierz parametry pomiaru

Wybierz parametry do wyszukania:

Wszystkie ▾

Wszystko ▾

Szukaj

Wszystkie  
 UTC  
 latitude  
 longitude  
 hdop  
 altitude  
 fix  
 cog  
 spkm  
 spkn  
 date  
 nsat

Rys.5. Wybór parametrów do wyświetlenia

Po wybraniu opcji *wszystko* (rys.6.) w co najmniej jednej liście rozwijalnej wyświetli się strona następującej treści:

### Rezultaty Pomiaru urządzeniem Telit

Szukaj ponownie

Ilość znalezionych pozycji: 531

Lp.	UTC	latitude	longitude	hdop	altitude	fix	cog	spkm	spkn	date	nsat
1	115248.000	5345.6367N	02027.4607E	1.2	124.6	3	3.30	0.50	0.27	140410	06
2	115250.000	5345.6370N	02027.4603E	1.4	123.8	3	338.31	0.75	0.40	140410	06
3	115253.000	5345.6371N	02027.4600E	1.4	124.0	3	78.96	0.43	0.23	140410	06
4	115255.000	5345.6371N	02027.4599E	1.4	124.8	3	38.40	0.21	0.11	140410	06
5	115257.000	5345.6370N	02027.4599E	1.0	125.6	3	12.96	0.14	0.07	140410	07
6	115259.000	5345.6369N	02027.4599E	1.0	126.5	3	48.41	0.10	0.05	140410	07
7	115301.000	5345.6369N	02027.4602E	1.0	126.4	3	326.40	0.50	0.27	140410	07
8	115303.000	5345.6364N	02027.4610E	1.0	127.8	3	79.82	0.18	0.09	140410	07
9	115305.000	5345.6360N	02027.4615E	1.0	129.0	3	18.14	0.21	0.11	140410	07
10	115307.000	5345.6358N	02027.4616E	1.0	130.7	3	354.39	0.10	0.05	140410	07
11	115310.000	5345.6357N	02027.4608E	0.8	135.9	3	310.49	0.14	0.07	140410	08
12	115312.000	5345.6359N	02027.4608E	0.8	137.9	3	309.50	0.54	0.29	140410	08
13	115314.000	5345.6360N	02027.4607E	0.8	139.6	3	291.93	0.14	0.07	140410	08
14	115316.000	5345.6360N	02027.4608E	0.8	141.1	3	232.48	0.10	0.05	140410	08

Rys.6. Rezultaty pomiaru



#### 4. WNIOSKI

Przewodowa i bezprzewodowa transmisja danych, programowanie, bazy danych na pewno ułatwią pracę współczesnemu geodecie. Techniki transmisji danych oprócz warstwy fizycznej obejmują również języki obsługi takie jak format NMEA-0183. Przydatnym przy zrozumieniu sentencji NMEA okazuje się być język komend modemowych AT. Sentencje NMEA wysyłane od lub do odbiornika GPS zawierają w sobie informacje, które można wykorzystać na wiele sposobów. Często jednak poprzedzić trzeba to odpowiednim ich sformatowaniem. Z pomocą przychodzą tu napisane w tym celu aplikacje. Umiejętne gromadzenie i administracja danych nawigacyjnych, geodezyjnych to konieczność w przypadku zbioru danych o wielkich rozmiarach.

#### 5. BIBLIOGRAFIA

- [1] Wesołowski K., *Systemy Radiokomunikacji ruchomej*, 2002
- [2] Telit *EVK2 Reference Guide*
- [3] Telit *GE863-Family Hardware User Guide*,  
Telit *Modules Software User Guide\_r5*,  
*AT Commands Reference Guide\_r9*, *Easy GPRS User Guide\_r9*  
*CMUX User Guide\_r3*, *Telit Easy Script Python\_r12*  
*SIM Access Profile User Guide\_r2* *Telit Battery Charger Application note\_*  
*Telit Audio Settings Application Note\_r4*,  
*Telit SIM Holder Design Guide Application Note\_r4*  
*Telit Antenna Detection Application Note\_r4*  
*Telit Digital Voice Interface Application Note\_r4*  
*Telit Event Monitor Application Note\_r2* *Telit Run AT Remotely Application Note\_*  
*r3*  
*Telit SIM Toolkit AT Application Note\_r0* *Telit PFM Application Note\_r8*  
*Telit PFM Certification Program Application Note\_r1*
- [4] G. van Rossum i inni: *Python Documentation*; <http://www.python.org/doc>; 1990–2009
- [5] G. van Rossum i inni: *Dokumentacja Pythona*; <http://docs.python.org.pl>; 1990–2003
- [6] M. Lutz, D. Ascher: *Python. Wprowadzenie*; wydanie III 2009, Helion, Gliwice