

Piotr Korbel

Projektowanie i programowanie systemów beprzewodowych Korzystanie z interfejsów komunikacyjnych

Zadanie nr 14 – Studia podyplomowe „Bezprzewodowe systemy nadzoru i monitorowania”



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna
współfinansowana przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
w projekcie

*„Innowacyjna dydaktyka bez ograniczeń
– zintegrowany rozwój Politechniki Łódzkiej –
zarządzanie Uczelnią,
nowoczesna oferta edukacyjna
i wzmacniania zdolności do zatrudniania
osób niepełnosprawnych”*



Politechnika Łódzka
Instytut Elektroniki

90-924 Łódź, ul. Żeromskiego 116,
tel. 042 631 28 83
www.kapitalludzki.p.lodz.pl



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Cele wykładu

- Zapoznanie słuchaczy z interfejsami komunikacyjnymi urządzeń działających pod kontrolą systemu Windows Mobile
- Zapoznanie ze sposobami programowania interfejsów komunikacyjnych urządzeń mobilnych



Politechnika Łódzka
Instytut Elektroniki

Projektowanie i programowanie systemów bezprzewodowych



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Plan wykładu

- Przegląd interfejsów komunikacyjnych urządzeń mobilnych
- Przykłady programów korzystających z wybranych interfejsów
 - IrDA
 - Bluetooth
 - Port szeregowy
 - Wysyłanie wiadomości tekstowych (SMS)
 - Wi-Fi/GPRS/EDGE/HSPA – System.Net.Sockets (TCP/IP)



Politechnika Łódzka
Instytut Elektroniki

Projektowanie i programowanie systemów bezprzewodowych



Interfejsy komunikacyjne urządzeń mobilnych

- Komunikacja bliskiego zasięgu
 - IrDA
 - Port szeregowy / USB
 - Bluetooth / Bluetooth po porcie szeregowym (Bluetooth w trybie serial data)
- Komunikacja dalekiego zasięgu
 - Sieć Wi-Fi
 - Telefonía komórkowa – GSM/SMS/GPRS/EDGE/UMTS/HSPA
- Sposoby komunikacji sieciowej
 - Sockets
 - Web services





IrDA

- Wymaga zapewnienia widzialności optycznej łączonych urządzeń
- Łatwe procedury nawiązywania łączności (nie jest wymagane parowanie urządzeń jak w przypadku systemu Bluetooth)
- Podstawowe etapy nawiązywania łączności:
 - Ustawienie portów w zasięgu widzialności optycznej
 - Wykrywanie urządzeń pozostających w zasięgu
 - Przesłanie danych pomiędzy urządzeniami





IrDA

- System.Net.IrDA.dll (System.Net, System.Net.Sockets)
- Klasy biblioteki System.Net.IrDA
 - IrDAEndPoint
 - IrDAListener (odbiór danych)
 - IrDACLient (wysyłanie danych)
 - IrDADeviceInfo
- Wymagane zdefiniowanie takiej samej wartości parametru SERVICE_NAME w obu łączonych urządzeniach (IrDAListener, IrDACLient)





IrDA – wykrywanie urządzeń

- IrDACLient
- Metoda DiscoverDevices()

```
private IrDAListener irListen;  
private IrDACLient irClient;  
private IrDAEndPoint irEndP;  
private IrDADeviceInfo[] irDevices;
```

```
String device;  
Int ID;
```

```
irDevices = irClient.DiscoverDevices(2);  
ID = BitConverter.ToInt32(irDevice.DeviceID, 0);  
device = ID.ToString() + " " + irDevice.DeviceName + " "  
        + irDevice.CharacterSet + " " + irDevice.Hints;
```





IrDA – odczyt danych

- IrDAListener
- IrDAClient

```
irEndP = new IrDAEndPoint(irDevices[i].DeviceID, irServiceName);  
irListen = new IrDAListener(irEndP);  
irListen.Start();
```

```
irClient = irListen.AcceptIrDAClient();
```

```
if (irListen.Pending() == true)  
    statusBar1.Text = "Pending ...";
```

```
Stream baseStream = irClient.GetStream();
```

```
irListen.Stop();  
irClient.Close();
```





IrDA – wysyłanie danych

- IrDACLient

```
irClient = new IrDACLient(irServiceName);
Stream baseStream = irClient.GetStream();

...
// Rozmiar przesyłanego pliku, zapis do strumienia
byte[] length = BitConverter.GetBytes((int)fileStream.Length);
baseStream.Write(length, 0, length.Length);
// Bufor danych odczytywanych z pliku
byte[] buffer = new byte[bufferSize];
while (fileLength > 0)
{
    int numRead = fileStream.Read(buffer, 0, buffer.Length);
    baseStream.Write(buffer, 0, numRead);
    fileLength -= numRead;
}
baseStream.Close();
irClient.Close();
```





Bluetooth

- Nie wymaga warunków widzialności optycznej łączonych urządzeń (LOS)
- Zasięg – od kilku do kilkadziesiąt metrów
- Zwykle wymaga parowania urządzeń
- Brak zarządzanych klas .NET CF do obsługi komunikacji
- Profil Serial Port umożliwia obsługę tak jak w przypadku zwykłego portu szeregowego
 - WM z WM
 - Zewnętrzne odbiorniki GPS
 - ...





Bluetooth – konfiguracja

- Konfiguracja urządzeń
- Należy skonfigurować usługi połączeń Bluetooth
- Ustawienia -> Połączenia -> Bluetooth -> Usługi -> Port szeregowy -> Zaawansowane...
 - Komunikaty przychodzące – COM7
 - Komunikaty wychodzące – COM8





Port szeregowy

- System.IO.SerialPort
- Utworzenie instancji obiektu SerialPort
 - `SerialPort sp = new SerialPort();`
- Konfiguracja paramterów pracy portu
 - `sp.PortName = "COM7";`
 - `sp.BaudRate = 4800;`
 - `sp.Parity = Parity.None;`
 - `sp.DataBits = 8;`
 - `sp.StopBits = StopBits.One;`
 - `ReadTimeout = 500;`
 - `WriteTimeout = 500;`





Port szeregowy

//Zapełnienie kontrolki combobox nazwami dostępnych portów (COM1, COM2, ...)

```
comboBoxPort.Items.Clear();  
string[] ports = SerialPort.GetPortNames();  
foreach (string prt in ports)  
{  
    comboBoxPort.Items.Add(prt);  
}
```





Port szeregowy – zapis danych

- Otwarcie portu
 - `sp.Open();`
- Zapis danych do portu
 - `Write, WriteLine`
- Zapis jednej linii danych tekstowych
 - `sp.WriteLine(str);`
- Zamknięcie portu
 - `sp.Close();`





Port szeregowy – odczyt danych

- Otwarcie portu
 - `sp.Open();`
- Odczyt danych
 - `Read, ReadByte, ReadChar, ReadLine, ReadExisting, ReadTo`
- Odczyt jednej linii danych tekstowych z portu
 - `str = sp.ReadLine();`
- Zamknięcie portu
 - `sp.Close();`





Port szeregowy – obsługa zdarzenia DataReceived

- Odbiór danych w momencie zasygnalizowania zdarzenia DataReceived
 - `SerialPort sp = new SerialPort();`
 - `sp.DataReceived += new SerialDataReceivedEventHandler(sp_DataReceived);`
 - `void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)`
 - `{`
 - `str = sp.ReadExisting();`
 - `}`





SMS – wysyłanie krótkich wiadomości tekstowych

- Microsoft.WindowsMobile.dll
- Microsoft.WindowsMobile.PocketOutlook.dll
- using Microsoft.WindowsMobile.PocketOutlook;

```
SmsMessage sms = new SmsMessage();
```

```
sms.Body = textBoxMsg.Text;
```

```
sms.To.Add(new Recipient(textBoxNo.Text));
```

```
sms.RequestDeliveryReport = true;
```

```
sms.Send();
```

```
statusBar1.Text = "Message sent!";
```





SMS – odbieranie (przechwytywanie) wiadomości tekstowych

- Microsoft.WindowsMobile.dll
- Microsoft.WindowsMobile.PocketOutlook.dll
- using Microsoft.WindowsMobile.PocketOutlook;
- using Microsoft.WindowsMobile.PocketOutlook.MessageInterception;
- Obsługa zdarzenia MessageReceived





SMS – odbieranie (przechwytywanie) wiadomości tekstowych

```
MessageInterceptor msgInterceptor;
```

```
private void Form1_Load(object sender, EventArgs e)
{
    msgInterceptor = new MessageInterceptor(
        InterceptionAction.NotifyAndDelete, true);
    msgInterceptor.MessageReceived += new
        MessageInterceptorEventHandler(
            msgInterceptor_MessageReceived);
}
```





SMS – odbieranie (przechwytywanie) wiadomości tekstowych

```
void msgInterceptor_MessageReceived(object sender, EventArgs e)
{
    //Obsługa zdarzenia i wysłanie potwierdzenia
    textBoxRx.Text = smsReceived.Body;
    try
    {
        SmsMessage sms = new SmsMessage()
        {
            Body = "Message received."
        };
        sms.To.Add(smsReceived.From);
        sms.Send();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```





SMS – filtrowanie odbieranych wiadomości tekstowych

```
private void Form1_Load(object sender, EventArgs e)
{
    msgInterceptor = new MessageInterceptor(
        InterceptionAction.NotifyAndDelete, true);

    // Warunki filtracji
    msgInterceptor.MessageCondition =
        new MessageCondition(
            MessageProperty.Body,
            MessagePropertyComparisonType.StartsWith,
            „Dear John”, true);

    msgInterceptor.MessageReceived += new
        MessageInterceptorEventHandler(
            msgInterceptor_MessageReceived);
}

//StartsWith, EndsWith, Contains, Equal, NotEqual
```





GSM/GPRS

- Microsoft.WindowsMobile.Status
 - SystemState
 - PhoneRadioPresent
 - PhoneRadioOff
 - PhoneRingerOff
 - PhoneSignalStrength
 - PhoneGprsCoverage
 - PhoneTalkingCallerNumber
 - PhoneMissedCall
 - PhoneMissedCalls
 - ...





GSM/GPRS

- Microsoft.WindowsMobile.Telephony
 - Phone
 - Talk(String)
 - Wybierany numer jako String
 - Talk(String, Boolean)
 - Wybierany numer jako String
 - Zmienna logiczna potwierdzenia zgody na wykonanie połączenia





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sockets

- System.Net
- System.Net.Sockets
- TcpListener
 - AcceptTcpClient()
- TcpClient



Politechnika Łódzka
Instytut Elektroniki

Projektowanie i programowanie systemów bezprzewodowych



Sockets – odbieranie danych

```
const string SERVER_IP = "192.168.1.1";  
const int PORT_NO = 13000;  
IPAddress localAddress = IPAddress.Parse(SERVER_IP);  
TcpListener listener = new TcpListener(localAddress, PORT_NO);  
Byte[] bytes = new Byte[256]; String data = null;  
listener.Start();  
while (true)  
{  
    TcpClient client = listener.AcceptTcpClient();  
    NetworkStream stream = client.GetStream();  
    ...  
    client.Close();  
}
```





Sockets – wysyłanie danych

```
const string SERVER_IP = "192.168.1.1";  
const int PORT_NO = 13000;  
const string MESSAGE_TO_SEND = „Message_to_send”;  
Socket client = null;  
  
IPAddress remoteAddress = IPAddress.Parse(SERVER_IP);  
IPEndPoint endPoint = new IPEndPoint(remoteAddress, PORT_NO);  
  
client = new Socket(endPoint.AddressFamily, SocketType.Stream, ProtocolType.Tcp);  
client.Connect(endPoint);  
  
byte[] data = ASCIIEncoding.ASCII.GetBytes(MESSAGE_TO_SEND);  
client.Send(data);  
client.Close();
```





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Literatura

- <http://msdn.microsoft.com/>
- www.devx.com



Politechnika Łódzka
Instytut Elektroniki

Piotr Korbel

Projektowanie i programowanie systemów beprzewodowych Korzystanie z interfejsów komunikacyjnych

Zadanie nr 14 – Studia podyplomowe „Bezprzewodowe systemy nadzoru i monitorowania”



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Prezentacja multimedialna
współfinansowana przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
w projekcie

*„Innowacyjna dydaktyka bez ograniczeń
– zintegrowany rozwój Politechniki Łódzkiej –
zarządzanie Uczelnią,
nowoczesna oferta edukacyjna
i wzmacniania zdolności do zatrudniania
osób niepełnosprawnych”*



Politechnika Łódzka
Instytut Elektroniki

90-924 Łódź, ul. Żeromskiego 116,
tel. 042 631 28 83
www.kapitalludzki.p.lodz.pl