

NFC W WINDOWS (PHONE) 8

MICHAŁ JANKOWSKI

0 MNIE



3 LATA DOŚWIADCZENIA Z WP

8 LAT Z TECHNOLOGIĄ .NET

Tworzenie aplikacji na WP.
Blog dotyczący .net oraz WP.

Kursy Walut:

- 30k ściągnięć,
- 1k ocen,
- średnia ocena 4,5,
- 3 miejsce w rankingu najczęściej pobieranych płatnych aplikacji.

Ponad 300k ściągnięć aplikacji.

PLAN WYSTĄPIENIA



CO TO JEST NFC

JAK TO OBSŁUGIWAĆ

W 2015 roku już 42,3% płatności mobilnych odbywać się będzie przy pomocy technologii NFC.

CO TO JEST NFC



NFC (Near Field Communication) – jest to komunikacja bezprzewodowa niewielkiego zasięgu. Umożliwia ona przenoszenie informacji między telefonem, a innym urządzeniem NFC (telefon, tag, urządzenia płatnicze).

Tagi NFC – małe, pasywne urządzenia z zaprogramowanymi informacjami.

NFC VS BLUETOOTH

PORÓWNANIE

KRYTERIUM	NFC	BLUETOOTH
Typ sieci	Peer-to-peer	Point-to-multipoint
Zasięg	< 0,1 m	10 m (max 100 m)
Tryb	Aktywny / Pasywny	Aktywny
Prędkość	424 kbit/s	40 Mbit/s
Bezpieczeństwo	Wysokie	Dobre
Czas nawiązywania połączenia	<0,1 s	<6 s

NFC W TELEFONACH

IOS



BRAK

ANDROID



WYBRANE MODELE

WINDOWS PHONE



WYBRANE MODELE

WYKORZYSTANIE NFC W WINDOWS PHONE 8



Najczęściej NFC w telefonie jest wykorzystywane do parowania urządzeń, przesyłania danych pomiędzy urządzeniami oraz odczytywania danych zapisanych na tagach NFC.

Zobaczmy jak to działa

KRÓTKA DEMONSTRACJA

PŁATNOŚCI ZBLIŻENIOWE

Problem zbyt wielu kart w portfelu rozwiąże Windows Phone!



Coś co mogło być wielkim sukcesem, a wyszło jak zwykle.

W Polsce ta funkcjonalność nie działa pomimo, że jest promowana.

Wczytaj kody pobieraj rozkłady!



QR Kod zgodny z większością czytników w telefonach komórkowych.

NFC Kod zbliżeniowy, obsługiwany przez telefony z funkcją NFC.

Wrocław miasto spotkań

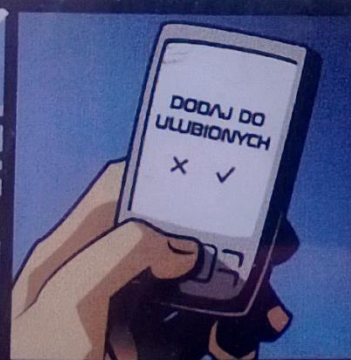
Przystanek Świeradowska

146	14:37	2 min.
BARTOSZOWICE - GAJ - pętla		
K	14:41	6 min.
KAMIENSKIEGO (Pętla) - GAJ - pętla		
136	14:48	13 min.
KOZANÓW - TARNOGAJ		
146	14:52N	17 min.
BARTOSZOWICE - GAJ - pętla		
K	14:58	23 min.
KAMIENSKIEGO (Pętla) - GAJ - pętla		
136	15:03N	28 min.
KOZANÓW - TARNOGAJ		
146	15:07	32 min.
BARTOSZOWICE - GAJ - pętla		
K	15:13N	38 min.
KAMIENSKIEGO (Pętla) - GAJ - pętla		
136	15:18	43 min.
KOZANÓW - TARNOGAJ		
	15:22N	47 min.

http://mobi.wroclaw.pl/przystanek_21133.html



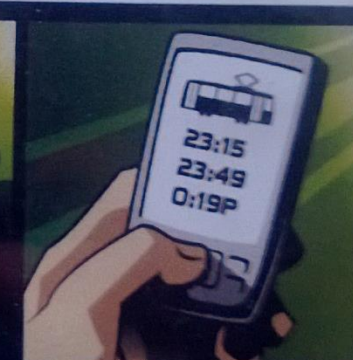
hom w komórce
k. Odczytaj aparatem



W ten sposób pobrałeś
adres strony z rozkładami



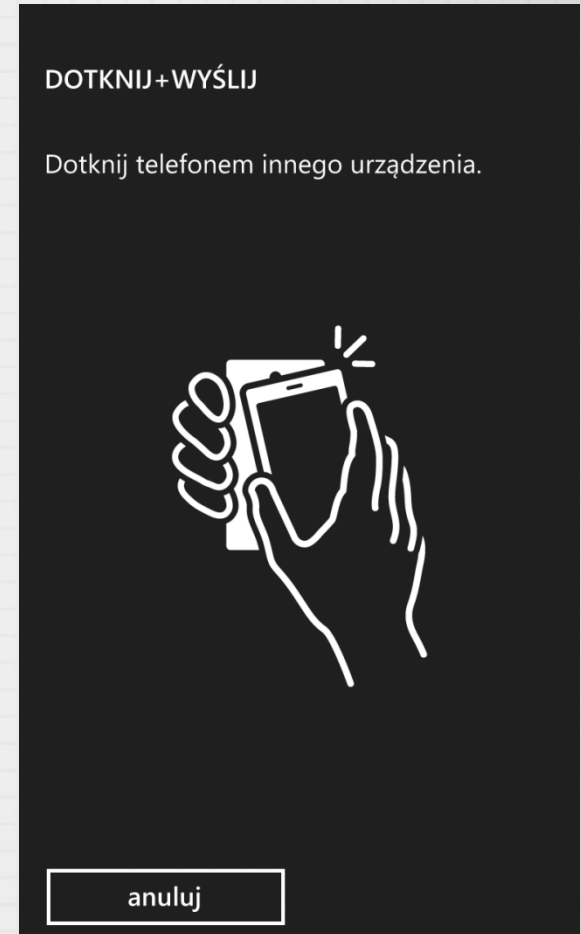
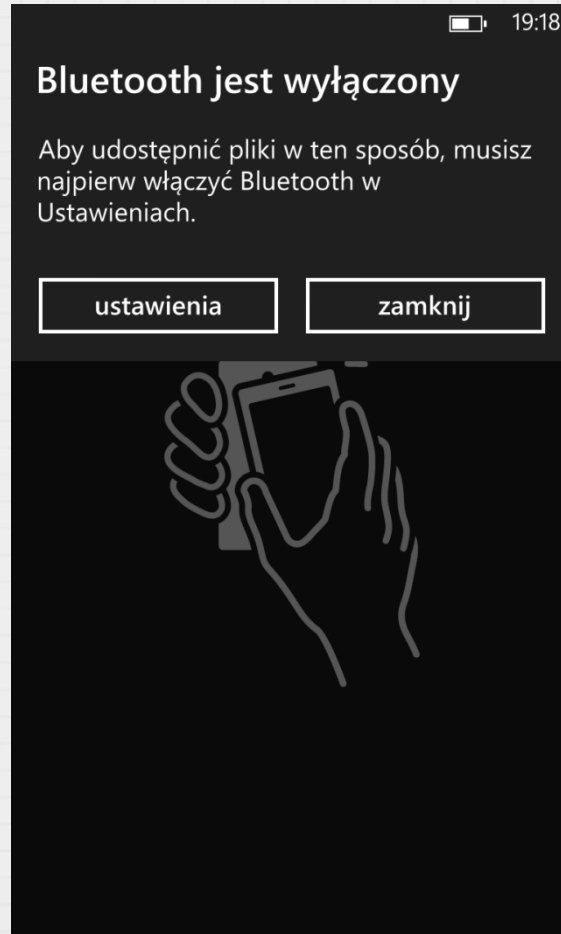
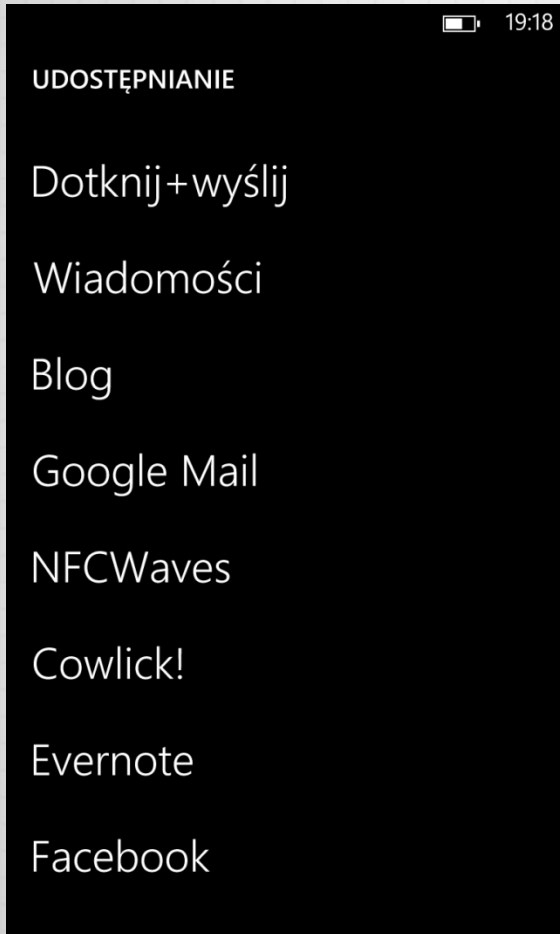
Za każdym razem, gdy
zarzysz na tę stronę



Każdy przystanek ma własny
kod tap kody przystanków

mobi.wroclaw.pl/przystanek

UDOSTĘPNIANIE

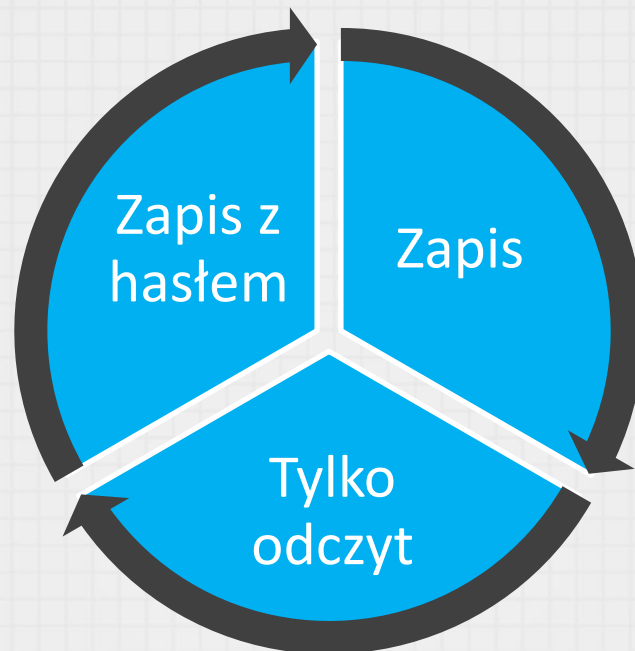


PRZYKŁADOWE TAGI

NAZWA	PAMIĘĆ [BAJTÓW]	DŁUGOŚĆ URL [ZNAKÓW]	DŁUGOŚĆ TEKSTU [ZNAKÓW]	KLASA NFC
Ultralight (UL)	48	41	39	Typ 2
Ultralight C (ULC)	148	132	130	Typ 2
NTAG203	144	132	130	Typ 2
MiFare 1K	1024	256	709	Classic (1)
DESFire EV1 2k	2048			Typ 4
DESFire EV1 4k	4096			Typ 4
DESFire EV1 8k	8192			Typ 4
Sony FeliCa 4K	4096			Typ 3
Sony FeliCa Lite	224			Typ 3
Topaz	512			Typ 1

Telefony z WP wspierają tagi sformatowane w standardzie NDEF (NFC Data Exchange Format). Należy o tym pamiętać przy ich kupnie.

SPOSOBY ZAPISU INFORMACJI



Windows Phone 8 jest w stanie odczytać każdy z tych tagów.

W przypadku zapisu nie ma możliwości zabezpieczenia taga przed zmianą. Zaleca się zamówić od razu odpowiednio zaprogramowany i zabezpieczony pakiet tagów ze sklepu.

OGRANICZENIE W WINDOWS PHONE 8

- Brak możliwości zabezpieczenia przed zapisem tagów.
- Brak możliwości sformatowania tagów do NDEF. Aby działały z naszym telefonem należy kupić sformatowane tagi lub użyć urządzenia wspierającego formatowanie (np. Android). W przypadku uszkodzenia taga też nie jesteśmy w stanie go naprawić.
- Tagi mogą zawierać tylko wiadomość w formacie NDEF.
- Brak możliwości użycia całej dostępnej pamięci taga – znaczniki, korekcja błędów, ...
- API nie wspiera *zapisywania* wiadomości w formacie NDEF domyślnie. Możemy to zrobić przygotowując wiadomość w formacie binarnym lub posłużyć się zewnętrzną biblioteką.
- Urządzenie domyślnie nie jest w stanie odczytać wszystkich wiadomości zapisanych na tagu. System wspiera odczyt tylko pierwszej wiadomości. Dodatkowe mogą być odczytane za pomocą własnej aplikacji.
- Nie ma możliwości odczytania taga przez aplikację działającą w tle.

Udostępnij zdjęcie

DEMO 1

KOD – SHARE MEDIA TASK

```
CameraCaptureTask cameraCaptureTask = new CameraCaptureTask();
cameraCaptureTask.Completed += (senderObject, photoResult) =>
{
    if (photoResult.TaskResult == TaskResult.OK)
    {
        ShareMediaTask shareMediaTask = new ShareMediaTask();
        shareMediaTask.FilePath = photoResult.OriginalFileName;
        shareMediaTask.Show();
    }
};

cameraCaptureTask.Show();
```




Wykrywanie urządzenia NFC

DEMO 2

ZANIM ZACZNIEMY - WMAPPMANIFEST.XML

The image shows two screenshots of the Windows Phone app manifest designer. The top screenshot shows the 'Capabilities' tab with a list of capabilities. The bottom screenshot shows the 'Requirements' tab with hardware requirements.

Capabilities Tab:

- ID_CAP_APPOINTMENTS
- ID_CAP_CONTACTS
- ID_CAP_IDENTITY_DEVICE
- ID_CAP_IDENTITY_USER
- ID_CAP_ISV_CAMERA
- ID_CAP_LOCATION
- ID_CAP_MICROPHONE
- ID_CAP_NETWORKING
- ID_CAP_PHONEDIALER
- ID_CAP_PROXIMITY

Requirements Tab:

Description
Provides access to Near Field Communication (NFC) services.
[More Info...](#)

Use this designer to set or modify some of the properties in the Windows Phone app manifest.

Application UI | Capabilities | **Requirements** | Packaging

Use this page to specify the hardware requirements of your application.

Hardware Requirements

- ID_REQ_NFC
- ID_REQ_FRONTCAMERA

Description
App requires a phone with a chip that e...
Selecting this option prevents the app f...
[More Info](#)

Należy wskazać funkcjonalności, które nasza aplikacja będzie wykorzystywać – NFC to IP_CAP_PROXIMITY. Często od razu można włączyć IP_CAP_NETWORKING.

Należy określić wymagania, które ma spełniać telefon – ID_REQ_NFC . Telefony bez NFC nie będą widziały tej aplikacji w Windows Store.

KOD – ZDARZENIA DLA NFC

```
ProximityDevice proximityDevice = ProximityDevice.GetDefault();  
if (proximityDevice != null)  
{  
    // Urządzenie ma moduł NFC  
    proximityDevice.DeviceArrived += proximityDevice_DeviceArrived;  
    proximityDevice.DeviceDeparted += proximityDevice_DeviceDeparted;  
}  
else  
{  
    // Brak modułu NFC  
}
```



Odczyt taga NFC

DEMO 3

KOD – ODCZYT TAGA

```
ProximityDevice proximityDevice = ProximityDevice.GetDefault();  
if (proximityDevice != null)  
{  
    long subscribedMessageId  
    = proximityDevice.SubscribeForMessage("WriteableTag", Handler);  
}  
  
void Handler(ProximityDevice device, ProximityMessage message)  
{  
    int writableSize = BitConverter.ToInt32(message.Data.ToArray(), 0);  
    this.ShowMessage("MessageType: " + message.MessageType + "\n"  
        + "DataAsString:" + message.DataAsString + "\n"  
        + "SubscriptionId: " + message.SubscriptionId + "\n"  
        + "Size of tag: " + writableSize + "\n");  
}  
  
proximityDevice.StopSubscribingForMessage(subscribedMessageId);
```

PROTOKOŁY

WINDOWS	The message contains binary data.
WINDOWSURI	The message data contains an URI (UTF-16LE encoded string). No subtype.
WINDOWSMIME	The message data is of a specific MIME type. For example, if the message data is a JPEG image, the message type is <code>WindowsMime.image/jpeg</code> .
WINDOWS:WRITETAG	Same as the Windows protocol, except that the content is intended to be written to a static tag. The message is not transmitted to any device except a writable static tag. This protocol is only valid for publications.
WINDOWSURI:WRITETAG	Same as the WindowsUri protocol, except that the content is intended to be written to a static tag. The message is not transmitted to any device except a writable static tag. This protocol is only valid for publications. No subtype.
LAUNCHAPP:WRITETAG	Write a tag that can be used to launch a specific app with optional launch parameters. If you publish a <code>LaunchApp:WriteTag</code> message to a tag the result when that tag is tapped to a computer is the same as calling the <code>PeerFinder.Start</code> method and passing your launch parameters.
WRITEABLETAG	When subscribing for this message protocol, if a writable tag is brought in to proximity, a proximity message is received that contains an <code>int32</code> (little endian) indicating the maximum writable size of the tag. This protocol is only valid for subscriptions.
NDEF	The message contents are properly formatted NDEF records. The underlying type of the content for a publication using NDEF as the message type is contained in the NDEF records. A subscription for the NDEF type subscribes to all NDEF formatted messages.
NDEF:EXT	The message data is application defined NDEF records (TNF field value of <code>0x04</code>). This protocol only applies to subscriptions; when publishing NDEF content, use NDEF.
NDEF:MIME	The message data is a properly formatted NDEF mime message (TNF field value of <code>0x02</code>). For example, <code>NDEF:MIME.image/jpeg</code> . This protocol only applies to subscriptions; when publishing NDEF content, use NDEF.
NDEF:URI	The message data is a properly formatted NDEF message of a type defined by a URI namespace (TNF field value of <code>0x03</code>). This protocol only applies to subscriptions; when publishing NDEF content, use NDEF. This means that the data format is identified by the specified URI.
NDEF:WKT	The message data is a properly formatted NDEF message of a type defined by the NFC forum (TNF field value of <code>0x01</code>). An example of this type is <code>NDEF:wkt.U</code> for the well known type of URI. This protocol only applies to subscriptions; when publishing NDEF content, use NDEF.
NDEF:WRITETAG	The message data should be written to an NFC forum standard static tag. The message data must be in a proper NDEF format. This protocol is only valid for publications.
NDEF:UNKNOWN	The message data is an untyped NDEF message (TNF field value of <code>0x05</code>). This protocol only applies to subscriptions; when publishing NDEF content, use NDEF.

Publikacja taga NFC

DEMO 4

KOD – ZAPIS TAGA

```
this.proximityDevice.PublishMessage("Windows.SampleMessage", "Hello World",  
                                     Handler);  
  
this.proximityDevice.PublishUriMessage(new System.Uri("http://www.onet.pl",  
                                                       Handler));  
  
this.proximityDevice.PublishBinaryMessage("Windows:WriteTag.mySubType",  
                                           Encoding.Unicode.GetBytes("Hello World").AsBuffer(), Handler);  
  
Uri uri = new Uri("http://www.wp.pl");  
byte[] buffer = Encoding.Unicode.GetBytes(uri.ToString());  
proximityDevice.PublishBinaryMessage("WindowsUri:WriteTag",  
                                     buffer.AsBuffer(), Handler);  
  
private void Handler(ProximityDevice sender, long messageId)  
{  
    this.proximityDevice.StopPublishingMessage(messageId);  
    this.ShowMessage("Wysłano / Zapisano!");  
}
```


KOD – ODCZYT TAGA

```
this.subscribedMessageId = this.proximityDevice.SubscribeForMessage(  
    "Windows.mySubType", Handler2);  
  
private void Handler2(ProximityDevice device,  
    ProximityMessage message)  
{  
    this.proximityDevice.StopSubscribingForMessage(  
        this.subscribedMessageId);  
  
    byte[] buf = message.Data.ToArray();  
    this.ShowMessage("MessageType: " + message.MessageType + "\n"  
        + "Data:" + Encoding.Unicode.GetString(buf, 0, buf.Length) + "\n"  
        + "SubscriptionId: " + message.SubscriptionId + "\n");  
}
```



Uruchomienie aplikacji

DEMO 5

KOD – ZAPIS TAGA

```
Uri uri = new Uri("ms-drive-to:?  
    destination.latitude=52.199506&  
    destination.longitude=21.041737&  
    destination.name=4Developers");  
  
byte[] buffer = Encoding.Unicode.GetBytes(uri.ToString());  
proximityDevice.PublishBinaryMessage("WindowsUri:WriteTag",  
    buffer.AsBuffer(), MessageTransmittedHandler);
```

PRZYKŁADOWE ADRESY DO WYKORZYSTANIA

HTTP:[URL]	Launches the web browser and navigates to the specified URL.
MAILTO:[EMAIL ADDRESS]	Launches the email app and creates a new message with the specified email address on the To line. Note that the email is not sent until the user taps send.
MS-SETTINGS-ACCOUNTS:	Launches the Account Settings app.
MS-SETTINGS-AIRPLANEMODE:	Launches the Airplane Mode Settings app.
MS-SETTINGS-BLUETOOTH:	Launches the Bluetooth Settings app.
MS-SETTINGS-CELLULAR:	Launches the Cellular Settings app.
MS-SETTINGS-EMAILANDACCOUNTS:	Launches the email and accounts settings app.
MS-SETTINGS-LOCATION:	Launches the Location Settings app.
MS-SETTINGS-LOCK:	Launches the Lock Screen settings app.
MS-SETTINGS-WIFI:	Launches the Wi-Fi Settings app.
ZUNE:NAVIGATE?APPID=[APP ID]	Launches the Windows Phone Store and shows the details page for the specified app.
ZUNE:REVIEWAPP	Launches the Store and shows the review page for the calling app.
ZUNE:REVIEWAPP?APPID=APP[APP ID]	Launches the Store and shows the review page for the specified app. Note that you must prepend “app” to the ID for the specified app. For example, the URI for reviewing an app with the ID fdf05477-814e-41d4-86cd-25d5a50ab2d8 would be zune:reviewapp?appid=appfdf05477-814e-41d4-86cd-25d5a50ab2d8
ZUNE:SEARCH?KEYWORD=[SEARCH KEYWORD]&CONTENTTYPE=APP	Launches the Store and searches for apps by keyword.
ZUNE:SEARCH?PUBLISHER=[PUBLISHER NAME]	Launches the Store and searches for items by publisher name.

Dodanie obsługi własnego protokołu do aplikacji

DEMO 6

KONFIGURACJA – WMAPPMANIFEST.XML

```
...  
</Tokens>  
<Extensions>  
  <Protocol Name="developers" NavUriFragment="encodedLaunchUri=%s"  
    TaskID="_default" />  
</Extensions>  
<ScreenResolutions>  
...
```

KOD – URI MAPPER

```
class AssociationUriMapper : UriMapperBase
{
    private string tempUri;
    public override Uri MapUri(Uri uri)
    {
        this.tempUri = System.Net.HttpUtility.UrlDecode(uri.ToString());
        if (this.tempUri.Contains("developers:MainPage?Param="))
        {
            int categoryIndex = this.tempUri.IndexOf("Param=") + 6;
            string param = this.tempUri.Substring(categoryIndex);
            return new Uri("/MainPage.xaml?Param=" + param, UriKind.Relative);
        }
        // Zwykle uruchomienie naszej aplikacji
        return uri;
    }
}

// Trzeba dodać go do App.cs:
RootFrame = new PhoneApplicationFrame();
RootFrame.UriMapper = new AssociationUriMapper();
```

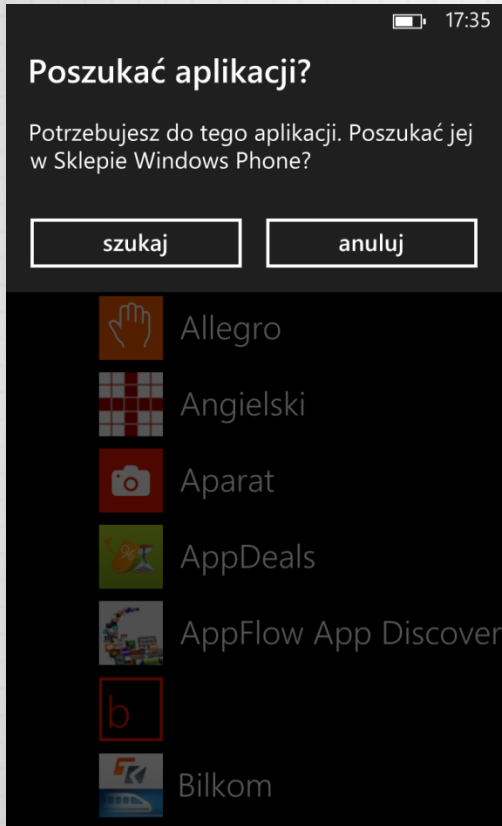
STRONA APLIKACJI – ONNAVIGATEDTO

```
// Odczyt parametru:
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    if (NavigationContext.QueryString.ContainsKey("Param"))
    {
        this.ShowMessage(NavigationContext.QueryString["Param"]);
    }
    base.OnNavigatedTo(e);
}

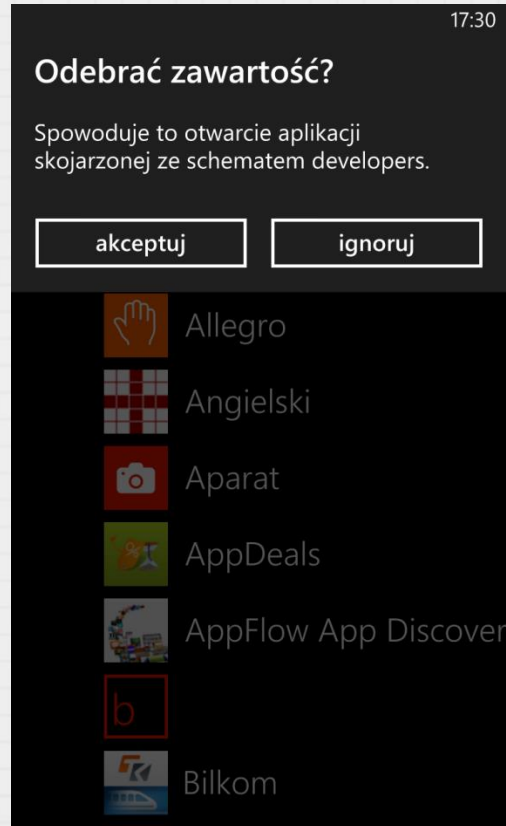
// Zapis to już znamy:
Uri uri = new Uri("developers:MainPage?Param=HelloWorld");
byte[] buffer = Encoding.Unicode.GetBytes(uri.ToString());
proximityDevice.PublishBinaryMessage("WindowsUri:WriteTag",
    buffer.AsBuffer(), MessageTransmittedHandler);
```


A TAK WYGLĄDA TO W TELEFONIE

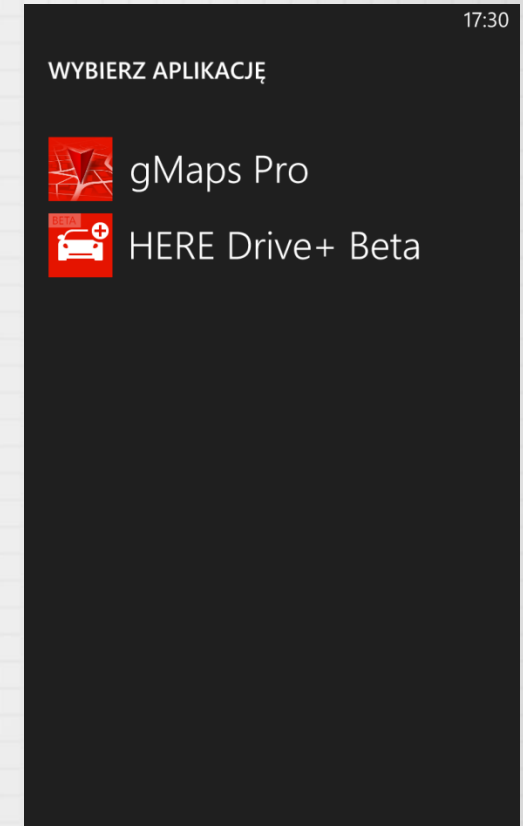
BRAK APLIKACJI



1 APLIKACJA



WIĘCEJ APLIKACJI



INNY SPOSÓB – LAUNCHAPPTAG

PARAMETRY

t
a
b

PLATFORMA 1

t
a
b

NAZWA APLIKACJI 1

t
a
b

PLATFORMA 2

t
a
b

NAZWA APLIKACJI 2

...

PRZYKŁAD:

PARAM=HELLOWORLD

t
a
b

WINDOWSPHONE

t
a
b

{27COEBCF-3BE5-416C-8668-B8BEB6FC2701}

...

STANDARD MIRCOSOFTU

DZIAŁA TYLKO NA WP8

Wadą jest dość duży rozmiar zajmowany przez dane na tagu.

Jednakże instaluje ściśle wybraną aplikację.

Wiadomość typu NDEF

DEMO 7

NDEF

Standard zapisu wiadomości na urządzeniach wykorzystujących NFC do komunikacji.

Specyfikacja dostępna na: http://www.nfc-forum.org/specs/spec_list/

REKORD NDEF

(URI, WIZYTÓWKA)

...

WIADOMOŚĆ NDEF

NFC DATA EXCHANGE FORMAT

Windows Phone potrafi domyślnie odczytać tylko pierwszy rekord z wiadomości.

Pozostałe rekordy mogą być odczytane przez dodatkowe programy.

NDEF I WINDOWS PHONE

```
while (i < message.Length)
{
    Debug.WriteLine("Parsing byte[] to NDEF message. New record starts at {0}", i);
    // Parse flags out of NDEF message header
    bool messageBegin = (message[i] & 0x80) != 0;
    bool messageEnd = (message[i] & 0x40) != 0;
    bool cf = (message[i] & 0x20) != 0;
    bool sr = (message[i] & 0x10) != 0;
    bool il = (message[i] & 0x08) != 0;
    var typeNameFormat = (NdefRecord.TypeNameFormatType)(message[i] & 0x07);

    Debug.WriteLine("ShortRecord: " + (sr ? "yes" : "no"));
}
```

Windows Phone pozwala na zapis wiadomości w formacie NDEF, ale w formacie binarnym. Zanim zapiszemy wiadomość musimy sami ją przygotować.

Alternatywnie można użyć biblioteki: NDEF Library (<http://ndef.codeplex.com/>).

KOD

```
// Zapis:
NdefUriRecord uriRecord =
    new NdefUriRecord { Uri = "http://4developers.org.pl" };
NdefMessage message = new NdefMessage { uriRecord };
ProximityDevice.GetDefault().PublishBinaryMessage("NDEF:WriteTag",
    message.ToArray().AsBuffer(), MessageTransmittedHandler);

// Odczyt
ProximityDevice.GetDefault().SubscribeForMessage("NDEF", MesReceivedHan);

void MesReceivedHan (ProximityDevice sender, ProximityMessage message)
{
    byte[] rawMsg = message.Data.ToArray();
    var ndefMessage = NdefMessage.FromByteArray(rawMsg);
    foreach (NdefRecord record in ndefMessage)
    {
        if (NdefUriRecord.IsRecordType(record))
        {
            var uriRecord = new NdefUriRecord(record);
            ShowMessage("URI: " + uriRecord.Uri);
        }
    }
}
```

Automatyczne nawiązywanie połączenia

DEMO 8

KOD

```
// Rozpoczęcie konfiguracji połączenia
PeerFinder.TriggeredConnectionStateChanged += OnTriggeredConnectionStateChanged;
PeerFinder.AllowBluetooth = true;
PeerFinder.AllowWiFiDirect = true;
PeerFinder.Start();

private void OnTriggeredConnectionStateChanged(object sender,
    TriggeredConnectionStateChangedEventArgs args)
{
    switch (args.State)
    {
        case TriggeredConnectState.Listening: // Łączenie jako host
            break;
        case TriggeredConnectState.PeerFound:
            // Zakończono używania NFC - połączenie wykorzystuje Wi-Fi, Bluetooth.
            // Można odsunąć urządzenia.
            break;
        case TriggeredConnectState.Connecting: // Łączenie...
            break;
        case TriggeredConnectState.Completed: // Połączenie ustanowiono
            streamSocket = args.Socket;
            break;
        case TriggeredConnectState.Canceled:
            break;
        case TriggeredConnectState.Failed: // Nastąpił błąd - trzeba jeszcze raz spróbować
            break;
    }
}
```


PODSUMOWANIE

WIELOPLATFORMOWOŚĆ

LINK WWW

WŁASNY
SCHEMAT
URI

LAUNCH
APPTAG

EXTERNAL
TYPE

WINDOWS
PROTOCOL

LINKI

- [How to use emulators to develop NFC features on WP8](#)
- [Proximity for Windows Phone 8 \(MSDN\)](#)
- [Use NFC tags with Windows Phone 8 \(Nokia\)](#)
- [NDEF Library for Proximity APIs \(NFC\)](#)

PYTANIA

?????

**DZIĘKUJĘ
ZA UWAGĘ**



MICHAŁ JANKOWSKI

WWW: WWW.JANKOWSKIMICHAL.PL

E-MAIL: JANKOWSKI.M@GMAIL.COM