



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI

Program Rozwojowy  
Politechniki Warszawskiej

EFES  
Europejski Fundusz Społeczny



## Zadania realizowane przez Wydział Elektryczny PW

### Komputerowe systemy wbudowane, sprzętowe ćwiczenie wprowadzające

#### Wstęp

Celem wstępnego laboratorium sprzętowego jest przygotowanie grupy studenckiej do samodzielnego korzystania z zasobów sprzętowych używanych w dalszej, regularnej części laboratorium w dedykowanych ćwiczeniach problemowych. Korzystanie to dotyczy również przygotowania studenta do realizacji, testowania i uruchamiania w komputerze wbudowanym prostego oprogramowania roboczego przy wykorzystaniu środowiska programowego Visual Studio 2005. Proste oprogramowanie obejmuje organizację komunikacji komputera wbudowanego z otoczeniem poprzez łącza RS i sieciowe (TCP/IP), parsowanie odebranych stringów i wykorzystanie grafiki funkcyjnej do zobrazowania danych. Istotny fragment nabywanej wiedzy dotyczy strony funkcjonalno-programowej urządzeń peryferyjnych do których należą: standardowe przyrządy pomiarowe wykorzystujące komunikaty SCPI, odbiornik GPS wykorzystujący protokoły NMEA i SiRF, oraz terminal GSM ze wsparciem sprzętowym dla transferów TCP/IP i UDP.

Dużą wagę zwrócono na możliwość realizacji pracy zdalnej (testowanie i uruchamianie aplikacji użytkownika) na platformach wbudowanych z wykorzystaniem pojedynczych komputerów sterujących wydzielonymi stanowiskami, oraz centralnej stacji serwerowej Windows 2003 z opcją wielodostępu. Praca zdalna pozwala, przy założeniu dostępu grupy studenckiej do sieci VPN PW EE, na korzystanie ze sprzętu w godzinach pozalaboratoryjnych, w tym w warunkach domowych. Stosowna konfiguracja i terminy udostępniania stanowisk oraz ustalanie zasad zdalnego dostępu wymaga uzgodnienia z osobą prowadzącą przedmiot. Zdalna praca jest szczególnie przydatna przy realizacji projektów, stanowiących opcję zaawansowanej kontynuacji ćwiczeń laboratoryjnych.

Cwiczenie zapoznaje studenta ze sprzętem i oprogramowaniem umożliwiającym przygotowywanie i uruchamianie własnych prostych aplikacji na emulatorach i sprzęcie wbudowanym dostępnym w laboratorium. Omawiane i testowane są różne metody integracji komputerów obiektowych ze środowiskiem uruchomieniowym, w szczególności w warunkach dostępu obydwu maszyn do łącz sieciowych. Używane w ramach tego etapu oprogramowanie dla platformy wbudowanej ma charakter wyłącznie testowy i nie korzysta z jej interfejsów, tym samym służy wyłącznie do testowania funkcjonowania środowiska uruchomieniowego. Celem tego etapu jest poznanie ograniczeń i możliwości organizacji



KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



PROGRAM ROZWOJOWY  
POLITECHNIKI WARSZAWSKIEJ

UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



dostępu do komputera wbudowanego w różnych wariantach konfiguracyjnych środowiska uruchomieniowego:

- jednostanowiskowy system bez Visual Studio (pkt. 1.1),
- jednostanowiskowy system z Visual Studio (pkt. 1.2),
- wielostanowiskowy (serwerowy) system z Visual Studio (pkt. 1.3).

Ćwiczenie realizowane jest w dwóch środowiskach:

- jednostanowiskowym – komputer XP z oprogramowaniem VS2005 na zespół dwuosobowy,
- wielostanowiskowym – wielodostęp do Windows Serwer 2003 z zainstalowanym VS2005 i zdalną pracą (RDC) zespołów dwu- lub jednoosobowych.

## 1. Środowisko uruchamiania oprogramowania nie korzystającego z łącz komunikacyjnych

W pierwszej fazie ćwiczenia realizowany jest dostęp i praca zdalna na wirtualnych i realnych komputerach Windows CE. Uruchamiana jest prosta aplikacja okienkowa i konsolowa (jeśli środowisko na to pozwala), bez elementów wykorzystania zewnętrznych łącz komunikacyjnych platformy wbudowanej. Celem tego etapu laboratorium jest poznanie środowiska w zakresie obsługi i możliwości dostępu do sprzętu i dostępnych emulatorów.

To samo proste zadanie realizowane jest w dwóch środowiskach programowych:

- jedno stanowiskowego (lokalnego) środowiska Windows XP/VS 2005,
- środowiska wielostanowiskowego Windows Serwer 2003/VS 2005.

W/w wariantowa realizacja ćwiczenia ma zorientować wykonujących laboratorium w ograniczeniach środowisk podnoszonych na relatywnie silnych klasycznych maszynach PC i przygotować studenta do samodzielnego zdalnego korzystania ze środowiska serwerowego.

Każde ze środowisk korzysta z Visual Studio 2005 i dostępu do zainstalowanych w nim emulatorów, oraz sprzętu wbudowanego w postaci:

1. komputera przemysłowego ADAM 6501/ Windows CE 4.2 (Advantech),
2. komputera przemysłowego Viper/Windows CE 5.0 (Arcom),
3. palmtopa Pocket LOOX T810/Windows CE 5.0 (Fujitsu Siemens).

W/w sprzęt wpięty jest w środowiska uruchomieniowe poprzez łącza sieciowe. Dwa pierwsze rozwiązania sprzętu wbudowanego umożliwiają jego w pełni zdalne wykorzystanie, z uwagi na udostępnione przez wytwórców oprogramowanie zdalnego dostępu (remote control) do ich sprzętu dla komputerów/stacji roboczych z s.o. Windows. Pocket LOOX nie daje takiej możliwości zdalnej pracy, ale może być zarządzany przez oprogramowanie Activesync, oraz ma dobrą alternatywę w postaci emulatorów.

O ile wyraźnie nie zaznaczono, omawiana w instrukcji realizacja ćwiczenia dotyczy pracy lokalnej, jedno stanowiskowej, która zapewnia większą (niż w wydaniu serwerowym) elastyczność w korzystaniu ze sprzętu i emulatorów. Zaletą drugiej opcji (serwerowej) jest łatwiejsza możliwość udostępniania możliwości pracy zdalnej, realizowanej spoza laboratorium w przypadku zakupienia stosownych licencji dostępowych.

- Lokalna praca odbywa się na równolegle na 10 komputerach laboratoryjnych z systemem Windows XP (i Visual Studio), dostępnych do wyłącznej pracy dla zespołów dwuosobowych. Komputery te selektywnie mogą wykorzystywać (po uzgodnieniu z prowadzącym) sprzęt wbudowany wpinany do nich poprzez łącza przewodowe (USB, RS) lub sieciowe. Nie istnieją podobne ograniczenia przy pracy z emulatorami, stąd zespoły opracowując aplikację dla platformy wbudowanej do końca pracują z emulatorem, korzystając (chwilowo) ze sprzętu tylko na finalnym etapie realizacji zadania, w celu weryfikacji poprawności działania aplikacji w warunkach „bojowych”.

- Praca wielostanowiskowa realizowana jest na odległym serwerze, a komputery laboratoryjne są tylko jego terminalami RDC (remote desktop connection) – wejście na serwer jest możliwe z każdego komputera, w tym domowego. Ten tryb pracy dedykowany jest raczej pod pracę wyłącznie z emulatorami z uwagi na to, że w odróżnieniu od trybu „lokalnego” wymaga uruchomienia tylko jednego komputera, który na dodatek może pracować bez nadzoru. Pozwoli to w przyszłości wykorzystać serwer do przygotowywania projektów i samodzielnych opracowań realizowanych poza Uczelnią w trybie dostępu „round the clock”.

## **1.1 Jednostanowiskowy system lokalny, bez Visual Studio**

### *1.1.1 Praca w systemie lokalnym, bez Visual Studio - dostęp do emulatora*

Windows XP pozwala na uruchomienie emulatora Windows CE v. 5.0 jako samodzielnej maszyny wirtualnej bez konieczności posiadania Visual Studio. Maszyna taka pozwala eksperymentować na środowisku CE w minimalnym środowisku programowym „gołego” XP.

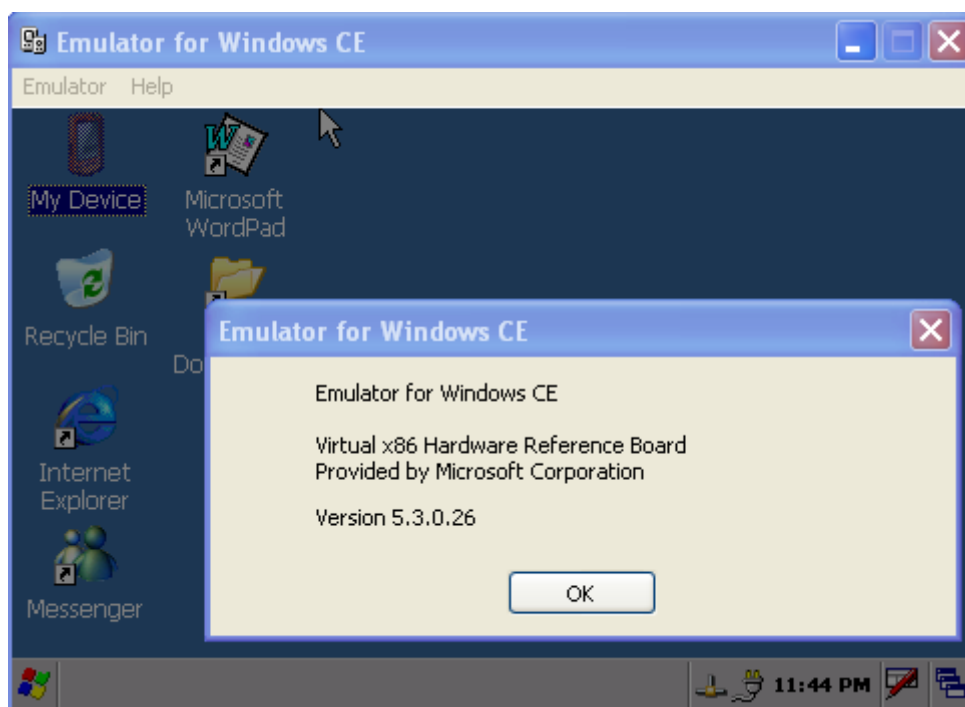
Uwaga: nie można tego samego zrobić (korzystać z emulatora) w zdalnie udostępnionej maszynie – zazwyczaj zwirtualizowana maszyna nie zawsze prawidłowo obsługuje odległą myszkę i klawiaturę.

Sprawdź jakie środowiska wirtualne masz zainstalowane w swojej stacji (Control Panel>Dodaj/usuń programy...). Poczekaj chwilę na wyniki przeglądania systemu, jeśli widzisz wpis Microsoft Windows CE 5.0 Emulator to masz szczęście – stosowny emulator jest zainstalowany. Jeśli nie jest tak dokonaj instalacji:

*Zaloguj się jako administrator (jest to niezbędne przy instalacji w systemie nowego oprogramowania).*

*Zainstalujesz za chwilę emulator w swoim PC korzystając ze ścieżki na serwerze FTP wskazanej przez prowadzącego. Zawiera ona małorozmiarowy setup (kilka MB). Przenieś go na swoją ścieżkę roboczą i uruchom. Po instalacji (trwa kilka minut) sprawdź możliwość uruchomienia emulatora (Start>Programy...). Wejdź w Control Panel do instalatora/deinstalatora programów i zauważ, że nowy emulator ma podobnie zorganizowane środowisko jak to, które „przyszło” z Visual Studio instalowanym wcześniej na stacji.*

Uruchom emulator.



Wykorzystaj w nim opcję udostępniania folderu komputera macierzystego (Emulator>Folder sharing...) i zorganizuj dostęp do swojego katalogu roboczego (w Windows XP). Ściągnij do emulatora aplikację da1\_ppc2003\_cf1.exe (serwer sierra.../ksw/) i uruchom ją. Uruchom tę samą aplikację w XP. Zauważ, że mimo różnej technologii sprzętu uruchomieniowego (żywy (XP) bądź emulowany (emulator) procesor x86) i sprzętu dla którego była przygotowywana (PPC003 - ARM4) aplikacja wykonuje się poprawnie w dwóch środowiskach (XP i CE) procesora x86. Oznacza to, że plik .exe generowany przez Visual Studio nie jest bezpośrednio ładowany do pamięci komputera, który go wykonuje. Plik ten opisuje program w metajęzyku CIL, wspólnym dla wielu języków programowania środowiska .NET, ulokowanym nieco powyżej asemblera. W momencie wykonywania instrukcje CIL w procesie kompilacji JIT (just in time), realizowanej współbieżnie z wykonywaniem programu zastępowane są wywołaniami konkretnego języka maszynowego. W ten sposób kod pośredni CIL jest wspólny nie tylko dla języków wysokiego poziomu, ale i dla platform na których jest wykonywany. Jeśli ten problem Cię interesuje głębiej poczytaj o CIL, JIT oraz managed/unmanaged code w Internecie.

Sprawdź czy masz dostęp do Internetu z emulowanej maszyny. Jeśli nie to zmień jej nastawy sieciowe Control Panel>Network&Dial-up... Dostęp do Internetu daje Ci możliwość sięgania po aplikacje dostępne w sieci www. Ma to znaczną wartość użyteczną. Jedną z takich aplikacji może być klient FTP, umożliwiający ładowanie plików do komputera wbudowanego, inną oprogramowanie wirtualizacji portów szeregowych, jeszcze inną moduły ConmanClient2 i CMAaccept służące do połączenia z środowiskiem VS.

Zauważ, że przeglądarka emulatora nie pozwala skorzystać z FTP –możliwość przenoszenia plików poprzez wspólny folder jest więc bardzo przydatna.

Przed chwilą pracowałeś w emulatorze i zorganizowałeś dostęp do jego zasobów i realnej pracy na najprostszym zasadzie: przygotuj aplikację, załaduj i uruchom. W podobny sposób wykorzystasz wkrótce platformy fizyczne.

Do przygotowywania aplikacji na sprzęt wbudowany (moduł .exe) możesz wykorzystać w zasadzie dowolne środowisko programistyczne nastawione na generację kodu na komputer wbudowany. Jeśli zamykasz się w ofercie Microsoft'u to najczęściej będzie to Visual Studio (licencjonowane), a rzadziej Embedded Visual C++ (darmowe). Komfort pracy z Visual Studio i darmowy akademicki dostęp do niego powodują, że EVC nie jest omawiane w niniejszej instrukcji.

### **1.1.2 Praca w systemie lokalnym, bez Visual Studio - dostęp do platform sprzętowych**

#### *Pocket LOOX*

Dostęp do Palmtopa Pocket Loox nie jest tu omawiany z uwagi na zazwyczaj bezproblemowe fizyczne i logiczne połączenie maszyn łączem USB. ActiveSync (o ile jest zainstalowany) sam odzyska komputer mobilny i zestawia połączenie pozwalające przenosić pliki pomiędzy platformą uruchomieniową i uruchamianą. Jeśli nie ma go w XP to pobierz go i zainstaluj z sieci (najlepiej w wersji 4.5, jest najbardziej otwarta na współpracę ze sprzętem).

#### *ADAM6501 - praca zdalna*

Wkomponowanie ADAM6501 w stację roboczą wymaga wywołania firmowej aplikacji kontrolującej uScope.exe (Google: adam 6500 uscope.exe):

[http://support.advantech.com.tw/support/DownloadSRDetail.aspx?SR\\_ID=1-LMTHZ](http://support.advantech.com.tw/support/DownloadSRDetail.aspx?SR_ID=1-LMTHZ)  
<http://support.elmark.com.pl/advantech/ia/ADAM6000/ADAM-6500/>

uScope jest samodzielny wykonywalny plikiem .exe i jako taki nie wymaga przeprowadzania instalacji. Przy problemach z dostępem do adresu IP Adam'a 6501 przy wywołaniu z Windows, program wywołać należy z linii poleceń, podając explicite IP ADAMa.

- pytanie o pomoc:

uscope ?

- wywołanie:

uscope 194.29.147.xxx

Wywołanie uscope jest bezskuteczne o ile na ADAMie otwarta jest inna sesja dostępu zdalnego. Sprawdź to kontaktując się z równolegle pracującymi kolegami.

#### *ADAM6501 – ładowanie aplikacji*



Z natury rzeczy nawiązanie połączenia z ADAM-em oznacza, że jest on wkomponowany w sieć i może mieć dostęp do Internetu (o ile są zrealizowane stosowne ustawienia DNS – sprawdź to: My Computer>Control Panel>Network and Dial up..>DM9CE1 itd.). Sprawdź czy działa przeglądarka i czy ADAM widzi sieć.

Dostęp taki (do sieci) jest istotny, gdyż pozwala Ci łączyć aplikacje dla platform mobilnych dostępne w sieci www. Niestety przeglądarka mobilna 6501 nie daje dostępu do ftp (sprawdź że tak jest to zapamiętasz). Nie masz również dostępu do zasobów lokalnych sieci - innych komputerów Windows pracujących w sieci lokalnej. W tej sytuacji korzystne jest załadowanie klienta FTP jako pierwszej aplikacji użytkowej. Masz to już zrealizowane – My computer/Flash/zs/e\_ftp

Poprzez FTP możesz wprowadzać pliki wykonawcze i uruchamiać je pracując zdalnie na ADAM6501. Korzystając ze wsparcia prowadzącego zajęcia uruchom w ten sposób poprzednią aplikację (da1\_ppc2003\_cf1.exe) na ADAMie.

Zauważ, że dzięki możliwości wstawienia karty Flash do ADAMa możesz pracować na tym komputerze w sposób klasyczny, fizycznie przenosząc nośnik z uruchamianym oprogramowaniem – jest to niewygodne i lokalne, ale tworzy realną alternatywę dla dostępu sieciowego.

Uwaga:

O ile potwierdziłeś sugestię, że w przeglądarce Adam'a 6501 nie działa wywołanie FTP to jesteś w błędzie. Problemem skutecznego wywołania jest składnia komendy. System przyjmuje wywołanie w tylko w szczególnej postaci:

<ftp://user:password@serwer ftp>,

a dodatkowo, w przypadku password podawanego jako adres mailowy (co najczęściej ma miejsce w odpytaniach anonimowych), w adresie moj@adres musisz zamienić „@” na %40 (znak specjalny @ w formacie hex). Klasyczny „at”: @, jest separatorem przed nazwą serwera i tylko w tym miejscu powinien być użyty. Oznacza to, że wywołanie sierry wymaga podania komendy:

<ftp://anonymous:moj%40adres@sierra.iem.pw.edu.pl>

Stosuj taką składnię wywołania ftp we wszystkich przeglądarkach/systemach gdy masz problemy dostępu do serwera ftp. Czasami Ci się uda...

Sprawdź to wywołanie i, jak poprzednio, dokonaj stosownego uruchomienia programu. Wywołanie ftp j.w. stosuj zawsze w przeglądarkach mobilnych – działa też w komputerze Viper co za chwilę sprawdzisz.

### *Viper 5.0 – praca zdalna*

W odróżnieniu od firmowego oprogramowania do pracy zdalnej używanego przez ADAM 6501, komputer Viper 5.0 podnosi server nasłuchowy VNC (Virtual Network Connection). Jest to słabsza, standardowa alternatywa dla windowsowego RDC (Remote Desktop Connection), używanego przy zdalnej pracy PC/PC. Popularność i wolny dostęp do VNC powoduje, że istnieje wiele wersji tego oprogramowania pracującego w trybie klient-serwer z serwerem uruchamianym po stronie maszyny udostępnianej. Server VNC zainstalowany w



Viper jest w wydaniu Tight, w związku z czym na stacji roboczej należy zainstalować klienta Tight VNC. Doświadczenie wskazuje, że najlepiej w ramach Tight korzystać z tej samej wersji oprogramowania klienta i serwera (tutaj v. 1.2.9) - rozbieżne wersje mogą stwarzać problemy komunikacji. Ściągnij i zainstaluj stosownego klienta i uruchom go po uzgodnieniu adresu IP Viper'a z prowadzącym. Sprawdź, że podobnie jak w przypadku ADAM6501, na Viper'rze może pracować zdalnie tylko jeden użytkownik.

### *Viper 5.0 – ładowanie aplikacji*

Tym razem rozsądnie jest zapewnić dostęp Viper'a do Internetu i sprawdzić działanie FTP w ramach przeglądarki. Użyć należy składni poznanej uprzednio (z %40), a skutek powinien być pozytywny. Korzystając z FTP przeglądarki dokonaj transferu i uruchomienia testowej aplikacji na komputerze obiektowym.

### *Podsumowanie: Praca w systemie lokalnym bez Visual Studio*

Na tym etapie realizacji ćwiczenia dysponujesz dwoma platformami sprzętowymi, które możesz wykorzystać do zdalnego uruchomienia modułów wykonawczych ładowanych z serwera FTP. Dysponujesz równocześnie lokalnie udostępnionym palmtopem Pocket LOOX T810 i emulatorem. Na wszystkich czterech maszynach powinieneś samodzielnie umieć uruchomić aplikację testową przygotowaną dla platformy docelowej jako programowy moduł wykonawczy (\*.exe). Uruchamianą aplikacją był moduł da1\_ppc2003\_cf1.exe. Długa nazwa modułu informuje, że przygotowywany był w Visual Studio jako Device application dla komputera PPC2003 (ARM) z Compact Framework 1.0. W rozdziale kolejnym (pkt. 1.2) nauczysz się wytwarzać własny moduł wykonawczy korzystając ze środowiska Visual Studio 2005.

## **1.2 Jednostanowiskowy system lokalny, praca z Visual Studio**

### *1.2.1 Praca w systemie lokalnym, z Visual Studio - dostęp do emulatorów zintegrowanych ze środowiskiem*

Tworzenie własnej aplikacji dla Windows CE w VS najlepiej zacząć od projektu Smart Device>PocketPC2003>DeviceApplication(1.0). Projekt taki jest uniwersalny gdyż:

- wymaga wsparcia starym Compact Framework 1.0, co oznacza, że daje się uruchomić na nowych i starych wersjach Windows CE/Mobile,
- jest dedykowany pod starszą wersję Windows CE (4.2), co nie stwarza problemów kompatybilności z nowszymi wersjami,
- jest projektem okienkowym co pozwala uruchamiać go na większości obiektów wyposażonych w ekran (nie wymaga konsoli).

Zrealizuj taki program (prosty przycisk i pole tekstowe), zbuduj moduł wykonawczy (Build>Build Solution) , odszukaj gdzie jest w systemie (w oknie Solution Explorer kliknij prawym przyciskiem myszy na dowolnym programie źródłowym (\*.cs) – powinieneś zobaczyć ścieżkę na której zapisywany jest twój projekt).



Uwaga :

Odszukanie miejsca lokowania programu jest istotne, gdyż VS pamięta środowisko ostatnio używane i zawycza tam (tzn. na ścieżkach poprzedniego użytkownika) zapisuje nowe projekty.

Uruchom program w emulatorze:

Uruchomienie Visual Studio i wybranie projektu dla Smart devices automatycznie udostępnia dostęp środowiska do emulatorów PPC2003 i Smatphone2003, oraz innych o ile zostały wgrane stosowne SDK. Emulatory te są silnie zintegrowane z VS i nie ma potrzeby korzystania z emulatora zewnętrznego "stand alone" który wykorzystywałeś dotychczas. Wybierz i podłącz emulator (najniższa lista narzędziowa górnej części ekranu, strona lewa trzy pierwsze pola). Podłączenie realizujesz korzystając z ikonki z kabelkiem. Powinien „podnieść się” emulator. Pobaw się nim chwilę i sprawdź czy widzi on Internet. Jeśli nie to zmień ustawienia interfejsu sieciowego ☹. Dostęp emulatora do Internetu w zasadzie nie jest konieczny, ale jest dostępną opcją i otwiera go na świat... Emulator jest mocno wpięty w środowisko Visual Studio i nie potrzebuje używania ftp do przeciągnięcia aplikacji pomiędzy komputerami.

Uruchomienie programu wymaga wywołania Debug>Start without debugging>Deploy

Aplikacja chwilę się ładuje (zwłaszcza przy pierwszym uruchomieniu, gdy „ciągnie” za sobą Compact Frameworka, o ile nie był wcześniej wgrany do małej maszyny).

Po załadowaniu (Deploy succeed) przejdź do ekranu emulatora i sprawdź jak działa twoja aplikacja. Zamknij ją, zrób małą zmianę widoczną dla użytkownika, (np. przesunij button, zmień komunikat...) i spróbuj ponownie ją przekompilować i uruchomić.

Uwaga:

Zazwyczaj przy pierwszej próbie uruchomienia programu masz dwa błędy:

Jeden polega na braku w systemie XP stosownego Frameworka (1.1) kompatybilnego z CF1.0 którego używa twój projekt. Framework ten, instalowany (przez Ciebie) na maszynie uruchomieniowej (XP) współpracuje z CF1.0 który instalowany jest automatycznie przez VS na maszynie obiektowej. Mimo tego, że zazwyczaj są zainstalowane w środowisku XP nowsze Frameworks, musisz doinstalować ten właśnie (wygooglaj go i zainstaluj, trwa to chwilę) by korzystać z CF1.0. Jeśli brak „dużego” Frameworka nie jest zgłaszany to masz szczęście.

Drugi błąd polega na niemożności zdeployowania kolejnej wersji twojej aplikacji na platformę mobilną. Nie „obchodź” takich błędów na siłę („continue”). Problem wynika z tego, że rutynowo zamknięcie okienka aplikacji często nie oznacza jej zakończenia, a tylko „odpina” ją od małego ekranu komputera obiektowego (namiastka minimalizacji z dużych Windows). Zakończenie zadania jest możliwe przez usługę systemu pozwalającą na usuwanie modułów załadowanych do pamięci lub mobilną wersję Task Managera – jeśli jest zainstalowany, miękki i twardy reset maszyny (soft/hard reset), lub ustawienie w opcjach aplikacji fałszu we właściwości Minimize twojego projektu. Ta ostatnia metoda jest najlepsza i musisz jej użyć.

Umiejętność przeglądania i kontrolowania zadań uruchomionych w systemie Windows CE jest przydatna i wymaga kolejno wywołania: settings>system>memory .

Sprawdź zdolność do szybkiej modyfikacji i uruchamiania twego, tym razem własnego (–masz przecież źródła), oprogramowania na emulatorze. Jak pracujesz sprawnie to uruchom „swój” program na jednym ze stanowisk sprzętowych, podobnie jak robiłeś to z da1\_ppc2003\_cf1.exe. (musisz odszukać i przenieść exec’a).



Łatwość uruchamiania oprogramowania na emulatorze dotyczy wyłącznie modułu kompilowanego w Visual Studio. W przypadku konieczności uruchomienia obcej aplikacji (moduł exe spoza VS) musisz postępować tak jak opisano w pkt 1.1.

### 1.2.2 Praca z Visual Studio - dostęp do maszyn realnych

W obecnym stanie dostępu do sprzętu najwygodniej pracować Ci z emulatorami zintegrowanymi z VS. Dostępne jest wówczas bezpośrednie ładowanie aplikacji (deploy) do komputera mobilnego, a nawet jej debugowanie podczas wykonywania na symulowanym sprzęcie (tego na razie nie robimy). Podobną funkcjonalność daje ActiveSync i oprogramowanie uruchamiane na Pocket LOOX. Poprzez łącze USB i ActiveSync palmtop równie dobrze integruje się z Visual Studio jak emulator.

Integracja platformy odległej (dostępnej przez sieć, jak ADAM6501 i Viper) z Visual Studio też jest możliwa poprzez uruchomienie na platformie docelowej dwóch programów: ConmanClient2.exe i CMaccept.exe zgodnie z opisem z:

<http://blogs.msdn.com/vsdteam/archive/2005/04/28/413304.aspx>

skróconą treść ww blogu zawiera wtrącenie poniżej:

Debugging on CE5.0 device without Activesync

This information ONLY applies to the Visual Studio 2005 Beta 2

Without the help of ActiveSync, VS 2005 does not automatically copy the connectivity binaries down to the device and hence debugging becomes problematic.

In order to use debugger on Windows CE 5.0 devices without active sync, you need to:

Step 1. Manually copy the following files down to the device

Clientsshutdown.exe  
ConmanClient2.exe  
CMaccept.exe  
eDbgTL.dll  
TcpConnectionA.dll

From the desktop folder:  
\\Program Files\\Common Files\\Microsoft Shared\\CoreCon\\1.0\\Target\\wce400\\<CPU>, to \\windows on device

Step 2. Manually launch the conmanclient2.exe

On the device side, open the command prompt and run "Conmanclient2.exe"

Step 3. Set the correct IP address

On the desktop side, open VS 2005  
Tools>options>device tools>devices



Choose Windows CE 5.0 device, click on "properties".

On the "Windows CE 5.0 device properties" dialog, click on "configure".

On the "Configure TCP/IP" Transport dialog, choose "use specific IP address" and type in the IP address of your windows CE 5.0 device.

Click OK.

Step 4. Enable the connection (You can skip this step if the security is already disabled on the CE device by setting "HLKM\System\CoreConOverrideSecurity = 1". But disabling security may expose your device to malicious attack)

Run cMaccept.exe

Connect to the device within 3 minutes after you run cMaccept.exe. (The 3 minutes window is for the first connection. As long as you establish the first connection within 3 minutes, the following deployment/debugging sessions using the same VS instance is not limited by this 3 minutes window)

You need to perform Step 4 again when you try to connect from another instance of VS.

Now debugger is ready to go, and you should be able to deploy and debug program(s) running on Windows CE 5.0 device now.

Published Thursday, April 28, 2005 9:09 PM by [MobileDevGroup](#)

Jak widać, wkomponowanie sprzętu obiektowego z dostępem do sieci w środowisko VS jest stosunkowo łatwe, ale wymaga obecności w nim kilku modułów programowych i ich właściwego wykorzystania (uruchomienia). Wgranie stosownych modułów do komputera obiektowego też nie jest trudne - potrafisz sam to zrobić. Źródła znajdziesz w katalogu:

Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\<CPU>

Zauważ, że w wyżej podanym katalogu Microsoft VS udostępnia całą gamę identycznych pakietów dla różnych platform wbudowanych – przy instalacji należy wybrać właściwy.

Nazwa instalacja jest użyta tu nieco na wyrost – chodzi o przegranie w/w modułów na ścieżkę roboczą w komputerze wbudowanym.

### *1.2.3 Integracja komputera obiektowego z Visual Studio (po zainstalowaniu oprogramowania)*

Odszukaj w komputerze wbudowanym ConmanClient2.exe i CMaccept.exe. Wejdź na stosowną ścieżkę i przygotuj się do wywołania tych programów. Sprawdź IP komputera mobilnego – będziesz go musiał podać w Visual Studio przy próbie nawiązania połączenia. Uruchom VS i w opcji (Tools>Options) wybierz sprzęt i w jego Properties ustaw stosowne IP w TCP/IP settings. Wywołaj połączenie: Connect to device (Tools>Connect to device) i w czasie szukania uruchom w wyżej podanej kolejności obydwie programy na komputerze obiektowym. Powinieneś uzyskać połączenie...

Po "wpięciu" komputera wbudowanego w środowisko VS możesz na nim pracować identycznie jak na emulatorze.

### 1.3 Podsumowanie

Na tym etapie ćwiczenia potrafisz skorzystać z Visual Studio do przygotowania aplikacji mobilnej przy wykorzystaniu emulatorów wbudowanych w środowisko, doń doinstalowanych i zewnętrznych. Emulatory zintegrowane z VS stanowią idealne narzędzie do tworzenia pierwszych roboczych wersji własnych aplikacji. Zasadniczy problem realizacji aplikacji o charakterze użytkowym stanowią ograniczenia emulatorów i ich oderwanie od świata rzeczywistego. Pracę ze środowiskiem Visual Studio i emulacją sprzętu wbudowanego umiesz realizować lokalnie (maszyny Windows XP, sala 505) lub logując się na serwerze (Windows Serwer 2003, pracuje w 413, XP 505 jako terminale RDC). Wiesz że moduł wykonawczy dla platformy wbudowanej (projekt Smart Device) dedykowany jest procesorowi ARM z ograniczeniami narzuconymi przez CF, ale może być uruchomiony w systemie XP. Stanowi to jego istotną wartość, gdyż aplikacja może być testowana w środowisku maszyny realnej (XP), a nie tylko wirtualnej (CE). Jak się wkrótce przekonasz, jest to dość ważne z praktycznego punktu widzenia.

Aplikacje użytkowe, zwłaszcza wbudowane, sterują sprzętem stąd nie mogą być zrealizowane i przetestowane skutecznie w środowisku symulacyjnym. Z uwagi na to poznałeś możliwości programowania trzech komputerów obiektowych: ADAM6501/CE 4.2, Viper/CE 5.0, wpiętych do sieci, palmtopa Pocket LOOX/CE 5.0 wpiętego lokalnie przez łącze USB/ActiveSync, oraz emulatora zewnętrznego Windows CE 5.0 (x86). Powinieneś umieć przygotować, załadować i uruchomić własną aplikację na każdej z czterech ww platform. Powinieneś rozumieć że najlepiej zintegrowane ze środowiskiem uruchomieniowym XP/VS są zasoby dostępne lokalnie (emulatory oraz Pocket LOOX). Umożliwiają one bezpośrednie ładowanie i debugowanie aplikacji z VS, oraz obsługę poprzez wirtualny panel dostępny na ekranie XP dla emulatorów i żywy ekran dotykowy palmtopa wpiętego w USB. Na dwóch pozostałych maszynach możesz pracować zdalnie (uScope i VNC), możesz ładować do nich aplikacje korzystając z łącz sieciowych i FTP i je zdalnie uruchamiać. Wiesz również, że aplikacje możesz przenosić do Viper'a przez pendriva USB, a do ADAMA przez kartę CF (ma wbudowany stosowny czytnik). Na Viper'ze możesz też pracować lokalnie (ma dostępny ekran dotykowy 240x320).