

Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

Podstawy Projektowania Przyrządów Wirtualnych Wykład 7: Arduino, zaawansowane operacje na plikach, obsługa błędów, dystrybucja aplikacji, tips & tricks

Co-funded by the Erasmus+ Programme of the European Union



Arduino

- myDAQ dobrze sprawdza się do akwizycji danych, jednak możliwości generowania sygnałów są ograniczone np. brak wyjścia PWM.
- Gdy potrzebujemy generować sygnały i obsługiwać pojedyncze sensory lepiej sprawdzi się mikrokontroler np. arduino.
- Aby skorzystać z arduino pod LabVIEW należy zainstalować jedną z dwóch dostępnych bibliotek LIFA lub LINX (wykorzystywaną na laboratoriach z PPPW) korzystając VI Packages Manager.
- Więcej na temat Arduino (ale z wykorzystaniem C++) na przedmiocie: **Podstawy Systemów Wbudowanych**.

Biblioteka LINX

- Biblioteka LINX może być użyta do komunikacji z Arduino, Raspberry Pi, myRIO oraz chipKIT.
- Biblioteka ta została stworzona przez LabVIEW MakerHub, organizację wspierającą środowisko makers przy użyciu LabVIEW. Więcej na stronie: https://www.labviewmakerhub.com/.
- Biblioteka LINX zawiera podstawowe funkcji takie jak:
 - **Open Serial** funkcja, która rozpoczyna połączenie z urządzeniem na wybranym porcie szeregowym.
 - Close funkcja, która kończy połączenie z danym urządzeniem.
 - oraz funkcje zgrupowane w katalogach:
 - $Peripherals \rightarrow Analog$ do obsługi wejść analogowych.
 - $Peripherals \rightarrow Digital$ do obsługi wejść/wyjść cyfrowych.
 - Peripherals $\rightarrow PWM$ do obsługi wyjść generujących sygnał PWM.
 - Peripherals \rightarrow 12C do obsługi magistrali 12C.
 - Peripherals \rightarrow SPI do obsługi magistrali SPI.

Aby wgrać firmware do Arduino należy po uruchomieniu LabVIEW wybrać: Tools → MakerHub → LINX → LINX firmware Wizard... a następnie wybrać odpowiedni typ płytki Arduino oraz port szeregowy.



- Biblioteka LINX umożliwia odczytywanie i wysyłanie sygnałów cyfrowych.
- Pierwszy przykładowy program ma na celu zapalenie/zgaszenie diody LED.



• Program ma na celu zapalenie diody LED po wykryciu ruchu przez czujnik PIR.



 Program odczytuje napięcie zwracane przez czujnik temperatury LM35. Napięcie te jest proporcjonalne do temperatury.



- Biblioteka LINX umożliwia również generowanie sygnałów PWM za pomocą funkcji *PWM Set Duty Cycle*.
- Funkcja ta przyjmuje na wejście numer pinu, na którym ma wygenerować sygnał PWM oraz wypełnienie (wartość numeryczna z zakresu 0-1 przy czym 1 oznacza 100%).



Więcej przykładów w książce: "LabVIEW and Open Source Solutions" Dariusz Tefelski, Angelika Tefelska.

Zaawansowane operacje na plikach

Tworzenie plików i folderów

Metody tworzenia ścieżek do plików i folderów

- Poprzez wpisanie na "sztywno" ścieżki:
 - Metoda użyteczna przy tworzeniu prototypów.
 - Nie zalecana do aplikacji.
- File Dialog (wyskakuje gdy ścieżka do funkcji zapisu danych nie jest podana):
 - Umożliwia użytkownikowi wybór ścieżki do pliku lub katalogu.
- Poprzez utworzenie zależnej ścieżki korzystając ze stałych w LabVIEW:
 - Tworzenie wzorców nazw plików i rozszerzeń np. test_001.txt, test 002 txt itd.
 - Logiczne grupowanie plików w katalogach.

Tworzenie zależnej ścieżki

- Tworzenie zależnych ścieżek i folderów umożliwia ominięcie problemu z różnymi ścieżkami dla różnych kont użytkowników pod różnymi systemami operacyjnymi.
- Przydatne funkcje:
 - Application Directory VI zwraca lokalizację projektu.
 - Get System Directory VI zwraca ścieżkę do katalogu głównego.
 - Build Path tworzy ścieżkę.



Tworzenie zależnej ścieżki

- Tworzenie zależnych ścieżek i folderów umożliwia ominięcie problemu z różnymi ścieżkami dla różnych kont użytkowników pod różnymi systemami operacyjnymi.
- Przydatne funkcje:
 - Application Directory VI zwraca lokalizację projektu.
 - Get System Directory VI zwraca ścieżkę do katalogu głównego.
 - Build Path tworzy ścieżkę.



User Home ✓ User Desktop User Documents User Application Data User Preferences User Temporary Public Documents Public Documents Public Application Data Public Preferences Public Cache System Core Libraries System Installed Libraries Application Files

Tworzenie folderu

- Funkcja umożliwiająca utworzenie folderu to Create Folder.
- Dobrą praktyką jest sprawdzenie czy dany folder już istnieje. W przeciwnym wypadku możemy uzyskać błąd przy próbie nadpisania istniejącego folderu.



Dynamiczne tworzenie ścieżek i nazw plików



Dynamiczne tworzenie ścieżek i nazw plików



Format Codes for the Time Format String

The Format Date/Time String function calculates the date and time and formats the output string according to the following format codes.

Format Code	Value
< 8 a>	abbreviated weekday name (for example Wed)
<%A>	full weekday name (for example Wednesday)
<%b>	abbreviated month name (for example Jun)
<%B>	full month name (for example June)
<%c>	locale-specific default date and time
<%d>	day of month (01-31)
<%H>	hour (24-hour clock) (00-23)
<%I>	hour (12-hour clock) (01-12)
<%j>	day number of the year (001-366)
<%m>	month number (01-12)
<%M>	minute (00-59)
<%p>	AM or PM flag
<\$S>	second (00-59)
<% <digit>u></digit>	fractional seconds with <digit> precision</digit>
<%U>	week number of the year (00-53), with the first Sunday as the first day of week one; 00 represents the first week
<\$w>	weekday as a decimal number (0-6), with 0 representing Sunday
<%W>	week number of the year (00-53), with the first Monday as the first day of week one; 00 represents the first week
<%x>	locale-specific date
<%.1x>	long date format
<8.2x>	abbreviated long date format
<\$X>	locale-specific time
<\$y>	year within century (00-99); when you scan the numbers, numbers (00-68) represent years in the twenty-first century (2000-2068) and numbers (69-99) represent years in the twentieth century (1969-1999)
<%Y>	year, including the century (for example, 1997)

Zapis wielu kolumn danych

 Przed zapisem danych do plików ważne jest obrócenie tablicy z danymi. W przeciwnym razie kolejne elementy kolumny nie będą zapisane pionowo lecz poziomo.



Odczyt wielu kolumn danych



Zapis danych do pliku binarnego



Odczyt danych z pliku binarnego



Czas na podsumowanie tej części wykładu! Zamiast tekstu, seria pytań!

Podsumowanie

Który z rodzajów zapisu danych wygeneruje plik o najmniejszym rozmiarze?



Podsumowanie

Który z rodzajów zapisu danych wygeneruje plik o najmniejszym rozmiarze?



Obsługa błędów

Obsługa błędów

- Błędy i ostrzeżenia są przekazywane w formie klastra, który składa się z trzech elementów:
 - **status** wartość logiczna, która przyjmuje wartość TRUE w przypadku błędu. Ostrzeżenia mają status FALSE.
 - code 32-bit signed integer.
 - source string, opis pomagający zidentyfikować źródło błędu.
- Ostrzeżenia posiadają *status*=FALSE oraz *code* ≠0.

Jak używać klastra błędów?

- Korzystanie z klastra błędów jest istotnym elementem profesjonalnych aplikacji.
- Przekazuj klaster błędów pomiędzy funkcjami, które posiadają terminale error in/error out.
- Pomiędzy kolejnymi iteracjami pętli for/while przekazuj klaster błędów korzystając z rejestru przesuwnego.
- Łącz klastry błędów za pomocą funkcji Merge Errors, co umożliwi obsługę wszystkich powstałych błędów.
- Jeśli dany fragment kodu może nie działać poprawnie gdy użytkownik wprowadzi nieprawidłowe dane, stwórz samemu błąd i przekaż do klastra błędów odpowiednią informacje.

Jak używać klastra błędów?

- Korzystaj z funkcji umożliwiających wyświetlenie okienka z informacjami o błędzie.
- Takie narzędzia są szczególnie istotne gdy aplikacja będzie udostępniana innym jako plik wykonywalny (exe). Przy odtwarzaniu pliku wynikowego użytkownik nie ma dostępu do listy błędów (*Error List*).



General Error Handler



Dialog box

- Gotowe funkcje do wyświetlania błędów są niezwykle przydatne.
- Jednak często chcielibyśmy przechwycić błąd i zapytać użytkownika co zrobić dalej: powtórzyć daną operacje czy może zamknąć program. Do tego przydają się dialog box.



One button dialog







Three button dialog

One button dialog



One button dialog



Two button dialog



Dodawanie własnego błędu do klastra

- Dodawanie własnych kodów błędów jest korzystne gdyż jako twórcy oprogramowania możemy przewidzieć więcej sytuacji powodujących nieprawidłowe działanie programu i im przeciwdziałać.
- LabVIEW zapewnia nam kilka zakresów wartości kodów, które możemy wykorzystać:
 - (-8999;-8000)
 - (5000, 9999)
 - (500 000, 599 999)
- Pozostałe wartości kodów są zarezerwowane przez inne funkcje. Dokładna dokumentacja kodów błędów jest dostępna na stronie

Dodawanie własnego błędu do klastra





Dodawanie własnego błędu do klastra - przykład



Dodawanie własnego błędu do klastra - przykład



Dodawanie własnego błędu do klastra - przykład


Dodawanie własnego błędu do klastra - przykład



Dodawanie własnego błędu do klastra - przykład



Dodawanie własnego błędu do klastra - przykład



Udoskonalanie istniejącego kodu

Refaktoryzacja

- Refaktoryzacja to proces przeprojektowywania aplikacji aby stała się bardziej czytelna i łatwiejsza w rozbudowie w przyszłości.
- Kiedy zabrać się za refaktoryzacje?
 - Wtedy gdy dodanie nowej funkcjonalności zabiera zbyt dużo czasu i wymaga znaczących zmian w istniejącym kodzie.
 - Kiedy aplikacja nie spełnia Twoich wymagań w satysfakcjonujący sposób.



Schemat refaktoryzacji

Problemy do rozwiązania:

- Brak organizacji
- Zbyt słabe nazewnictwo
- Zbyt skomplikowany kod
- Powtarzające się kawałki kodu
- Brak kontroli nad przepływem danych.



- Brak organizacji przemyśl wybór wzorca projektowego. Najczęściej stosuje się maszynę stanów lecz może w tym przypadku wzorzec Producent/Consument będzie lepszy.
- Zbyt słabe nazewnictwo zmień nazwy kontrolek i indykatorów tak by było oczywiste do czego służą.
- Zbyt skomplikowany kod zastanów się czy nie można danej operacji wykonać w sposób prostszy. Przykład: obsługa silnika krokowego.



 Powtarzające się kawałki kodu - zamknij je w subVI, który dobrze udokumentuj (zmień ikonę z domyślnej na sugerującą działanie, dodaj opis do pomocy kontekstowej, dobież odpowiednio nazwy terminali).



 Brak kontroli nad przepływem danych - zastąp Sequence structures maszyną stanów, jeśli to możliwe usuń zmienne lokalne w zamian przekazując wartości bezpośrednio z kontrolek lub stosując kolejkę/FGV.



Przygotowanie plików

- Główny VI zmodyfikuj korzystając z VI Properties jako Top-level application.
- Wykorzystaj properties nodes i invoke nodes do zmiany wyglądu kontrolek/okna. Wyłącz niepotrzebne paski narzędzi (toolbar), suwaki itd.

Category	Window Appearance
Window title	
Untitled 1	Same as VI name
Tan laud antication window	
Dialog	File Edit Operate Teels
Default	2231 @
Custom	
Customize	



Przygotowanie ścieżek

- Przygotuj ścieżki zależne od miejsca uruchomienia aplikacji a nie jako stałe.
- Pamiętaj, że ścieżki folderów mogą się różnić w zależności od uruchomionego systemu operacyjnego.



Zamknięcie aplikacji

 Upewnij się, że w przypadku wystąpienia błędów aplikacja zostanie zamknięta.





Category	Information
Information Source Files Destinations Source File Settings Icon Advanced Additional Exclusions Version Information Windows Security Shared Variable Deployment Run-Time Languages Pre/Post Build Actions Preview	Build specification name My Application Target filename Histogram Data.exe Destination directory C:\perforce\custEd\Course\\InDev\LVCore2\2012\English\SolutionSource\Exercise 6-1\Executable Build specification description

My Application Properties				×
Category		Source Files		
information Source Files Destinations Source File Settings Icon Additional Exclusions Version Information Windows Security Shared Variable Deployment Run-Time Languages Pre/Post Build Actions Preview	Project Files Project Files Logged Data File D Module File D Module Histogram Main.vi File Consume Message.ett File Consume Message.ett File Consume Message.ett	51 	artup VIs	· · ·
• Always include	Always include a Startup VI.		lways Included	
 Do not include statically linked files. 		→		
 Include any dynamically linked files in Always Included section. 				Ŧ
		Build	d OK Cancel	Help

Catalana	
Information	Advanced
Source Files Destinations Source File Settings	Wat for debugging
icon	Copy error code files
Advanced	Use custom aliases file
Additional Exclusions Version Information	Aliases file in project
Windows Security	
Shared Variable Deployment	Use custom configuration file
Pre/Post Build Actions	Configuration file in project
Preview	
	Pass all command line arguments to application
	Fnable ActiveX server
	ActiveX server name
Engla dalarra	
Enable debug	gging to identify run-
time problem	s in deployed FXFs
une problem	
Enabling deb	unging increases the
	agging moreases me
EXE size.	
EXE size.	

My Application Properties Category Information Source Files Destinations Source File Settings Icon Advanced Additional Exclusions Version Information Windows Security Shared Variable Deployment Run-Time Languages Provident Build Actions Preview	Generate Preview Generated Hies Citemp1build3/Histogram\My Applica data Histogram Data.aliases Histogram Data.air Histogram Data.air	Preview	
 Preview the budirectory structure generate. Don't build yet project. 	hild to see the eture and files that t. First save the		Cancel Help



Build Status

Build status	
My Application	
Compiling	
Warnings	
Explore Done Cancel Help	
	Build status
	My Application
	The build is complete. You can locate the build at C:\perforce\custEd\Courses\InDev\LVCore2\2012\English\SolutionSource\ Exercise 6-1\Executable.
	Warnings
1	Done Cancel Help

Run EXE



Dlaczego warto tworzyć instalator?

- Do wykonywania się aplikacji w LabVIEW nie potrzeba tak naprawdę LabVIEW. Wystarczy darmowe oprogramowanie: LabVIEW Run-Time Engine (RTE).
- Jeśli aplikacja korzysta z pobranych sterowników, można wymusić ich instalację wraz z plikiem EXE.
- Instalator zapewnia skopiowanie niezbędnych plików do pożądanej lokalizacji.

Installer Properties

Category	Product Information	
Destinations	Build specification name	
Source Files	My Installer	
Source File Settings	Desident anno	
Additional Installers Westher Station		
Dialog Information	weather station	
Registry	Installer destination	
Hardware Configuration	C:\My Installer	
Windows Security		
Advanced		

Installer Properties

Category Product Information	Addition	al Installers
Destinations Source Files Source Files Shortcuts Additional Intellers Dialog Information Registry Hardware Configuration Windows Security Advanced	National Instruments Installers to Include Microsoft Silverlight 5.1 Microsoft Silverlight 5.1 Microsoft Silverlight 5.1 NI DataSocket 5.0 NI DataSocket 5.0 NI Distributed System Manager 2012 NI Uo Trace 3.0. NI LabVIEW 2011 SP1 Deployable License NI LabVIEW 2011 SP1 Deployable License NI LabVIEW Run-Time Engine 2010 SP1 Mit LabVIEW Run-Time Engine 2010 SP1 NI LabVIEW Run-Time Engine 2011 SP1 Mit LabVIEW Run-Time Engine 2012 V NI LabVIEW Run-Time Engine 2012 NI LabVIEW Run-Time Engine 2012 V NI LabVIEW Run-Time Engine 2012 NI LabVIEW 2012 Foror Reporting V NI LabVIEW 2012 Run-Time Engine Non-Engine NI LabVIEW 2012 Run-Time Engine Non-Engine V NI LabVIEW 2012 Web Server 12.0 NI N LabVIEW Run-Time Engine 8.2.1 V NU C2008/SN4s NI LabVIEW Run-Time Engine 8.2.1 V Copy selected installers to this computer before buildi Copy all future installers to this computer when you run	Install type Full Description Libraries and other files necessary to execute LabVIEW 2012-built applications and shared libraries. Includes NI Report, and a browser plug-in that allows clients to view and control front panels remotely using a browser. Distribution title Local Cache Installer source location C<\Program Files\National Instruments\Shared\ ProductCache ng unterm

Instalacja

gram Data
Destination Directory Select the prinsey installation directory.
All software will be installed in the following locations: To install software into a different locations, click the Browne button and neted another directory.
Directory for Histogram Data
C-Program Files/Histogram/ Browner.
Directory for National Instruments products



Tips & Tricks

Organizacja elementów na front panel-u

- Elementy na front panelu można oddzielać dekoracjami lub zgrupować w różnych zakładkach **Tab Control**.
- Tab control zwraca numer zakładki, który użytkownik wyświetla numerując od zera.



Organizacja elementów z wykorzystaniem dekoracji



Organizacja elementów z wykorzystaniem subVI

- Część parametrów można podawać bezpośrednio w wyskakującym subVI. W tym celu oszczędza się miejsce na front panelu.
- SubVI musi otwierać się w momencie uruchomienia. W tym celu należy wybrać odpowiednią opcję w VI properties.

VI Properties		×
Categ	ory Window Appearance	
Window title Untitled 4 C Top-level application windox Dialog Dialog Default C Custom Customize	Customize Window Appearance V Window has title bar Show wertcal scrol bar* Show korizontal scrol bar* Show korizontal scrol bar* Show toubar when running Show toubar when running Show Abort button Show Run Doutron Show Run Continuously button Show Run Continuously button Case afterwards for ginally dosed Conse afterwards for ginally dosed	Window Behavior Default Default Default Highlight Enter boolean Window runs transparently Allow user to reise window Allow user to reise window Allow user to reise window Allow user to minimize window Allow
		OK Cancel Help

Organizacja elementów z wykorzystaniem subVI



Ukrywanie menu



Front Panel na środku ekranu



Ukrywanie scrollbar-ów



Zmiana stylu kursora

- Czasami dobrze jest zmienić wygląd kursora np. na klepsydrę w czasie wczytywania/generowania dużych ilości danych.
- W tym celu należy skorzystać z odpowiednich funkcji znajdujących się w: Functions Palette \rightarrow Dialog & User Interface \rightarrow Cursor



Zmiana stylu kursora



Zmiana stylu kursora


Usuwanie obramowań klastrów

- Przetwarzanie danych w klastrze jest wygodniejsze lecz obramowanie klastra może wyglądać nieestetycznie.
- Obramowanie klastra z palety Classic można ukryć wybierając:
 - **1** View \rightarrow Tools Palette,
 - w narzędziu do zmiany koloru wybieramy kolor przezroczysty (T),
 - 3 zmieniamy kolor klastra,
 - ④ klikamy na klaster i wybieramy Visible ItemsightarrowLabel



Basic tricks

Zmiana stylu kontrolek/indykatorów

- Czasami standardowy wygląd kontrolki/indykatora nie jest zadowalający.
- Wszystkie kontrolki/indykatory z palety Classic można edytować. W tym celu:
 - Kliknij prawym przyciskiem na danym obiekcie i wybierz: $Advanced \rightarrow Customize$
 - 2 Następnie wybierz: *Window* \rightarrow *Show Parts Window*. W powstałym okienku wybierasz część danego obiektu, którą chcesz modyfikować.
 - 8 Kliknij prawym przyciskiem myszy na części obiektu, który chcesz edytować a rozwinie się lista opcji z których możesz skorzystać.



Kontrolki/Indykatory na obrazku





Basic tricks

Animacja za pomocą picture ring



Basic tricks

Podświetlanie elementów



Grafika 2D

- Grafikę 2D można tworzyć wykorzystując funkcję z: Graphics & Sound
 → Picture Functions.
- Przykład dodawania punktów do obrazka. Pamiętaj, że obrazek ma wymiary, które można odczytać korzystając z property nodes - bounds.



Grafika

Grafika 2D

Obiekty, które można narysować:

- punkt,
- linię,
- prostokąt,
- okrąg,
- tekst.



Angelika Tefelska

Grafika

Grafika 3D

- Grafikę 2D można tworzyć wykorzystując funkcję z: Graphics & Sound → 3D Picture Control.
- Przykład rysowania prostopadłościanu.



Grafika 3D



Grafika

Dostępne geometrie





Dziękuję za uwagę!

Wsparcie Komisji Europejskiej dla produkcji tej publikacji nie stanowi poparcia dla treści, które odzwierciedlają jedynie poglądy autorów, a Komisja nie może zostać pociagnięta do odpowiedzialności za jakiekolwiek wykorzystanie informacji w niej zawartych.

Wykład został opracowany w oparciu o materiały: "LabVIEW Core 1 Course Manual", "LabVIEW Core 2 Course Manual", pierwotną wersje wykładu: mgr. inż. Marcina Biedy oraz przykładowe egzaminy CLAD opublikowane na stronie www.ni.com .

Angelika Tefelska

PPPW 2022/2023