

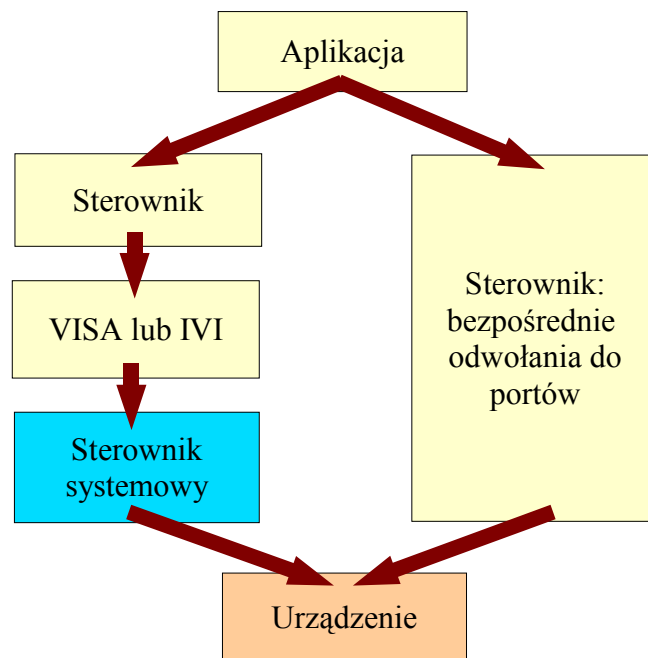
Akumulacja danych za pomocą kart pomiarowych UIA i UIB (CMA).

Wirtualny oscyloskop.

Wprowadzenie

Środowisko programistyczne LabView firmy National Instruments (NI) to dobry przykład systemu RAD (Rapid Application Development), czyli systemu szybkiego tworzenia aplikacji. LabView jest zatem podobne w filozofii tworzenia oprogramowania do takich środowisk jak MS Visual Basic czy Borland Delphi i JBuilder. Środowiska te różnią się jednak zasadniczo przeznaczeniem. LabView w przeciwieństwie do pozostałych wymienionych środowisk RAD jest przeznaczone przede wszystkim do tworzenia aplikacji obsługujących elektroniczny sprzęt pomiarowy i sterujący. Posiada ono również szereg funkcji umożliwiających programiście bardzo szybkie utworzenie oprogramowania służącego analizie danych zebranych na drodze pomiaru lub przeznaczonych do sterowania urządzeniami. Z tej przyczyny środowisko to znajduje zastosowanie przede wszystkim w laboratoriach i w przemyśle.

Komunikacja z zewnętrznymi urządzeniami pomiarowymi i sterującymi odbywa się w LabView za pośrednictwem odpowiednich sterowników. Przez sterowniki należy w tym przypadku rozumieć zestaw funkcji (tzw. „vi”), które pozwalają komunikować się z wybranym urządzeniem tj. zainicjować je, skonfigurować, zebrać dane, wykorzystać urządzenie do sterowania innym, a na koniec zamknąć połączenie z urządzeniem. Sterowniki będące integralną częścią LabView, albo pisane przez osoby trzecie z myślą o LabView, są najczęściej tworzone z wykorzystaniem jednego z dwóch zestawów funkcji wysokiego poziomu, pośredniczących w komunikacji z urządzeniem. Zestawy te to utworzony w 1993 roku min. przez NI standard VISA (Virtual Instrumentation



Rysunek 1. Komunikacja z urządzeniem w LabView. Żółte pola odpowiadają oprogramowaniu stworzonemu w LabView.

Software Architecture), oraz utworzony w 1998 roku również min. przez NI standard IVI (Interchangeable Virtual Instrument). Korzystając z funkcji VISA lub IVI należy pamiętać, że funkcje te pośredniczą w komunikacji pomiędzy funkcjami używanymi bezpośrednio w aplikacji napisanej w LabView, a funkcjami właściwego sterownika systemowego wybranego urządzenia. Nie znając zatem postaci funkcji sterownika systemowego, możemy użyć funkcji API (*application program interface*) VISA lub IVI i skomunikować się z urządzeniem. Strukturę zależności między poszczególnymi zbiorami funkcji przedstawia rysunek 1.

W przypadku gdy sterownik systemowy jest niedostępny użycie funkcji VISA czy IVI jest niemożliwe. Konieczne jest utworzenie własnego sterownika. Jeżeli programujemy w środowisku Windows 95, 98 lub Me, wówczas możliwe jest odwoływanie się bezpośrednio do portów urządzenia. Dysponując dostarczoną przez producenta urządzenia (np. karty pomiarowej) opisem rozkazów jakie należy przesłać do przyrządu na określone porty, możemy utworzyć z poziomu LabView zbiór funkcji, które posłużą komunikacji z przyrządem. Zbiór ten będzie naszym sterownikiem urządzenia. W przypadku Windows NT, 2000, Server i XP sytuacja może być bardziej złożona. Systemy te standardowo uniemożliwiają bezpośredni zapis i odczyt z portów urządzeń. Konieczne jest wówczas stworzenie własnego pliku sterownika systemowego, co okazuje się żmudnym i wymagającym dużej wiedzy procesem. Niewątpliwie jednak tak powinno się postępować, gdyż wówczas system jest obciążony odpowiedzialnością za synchronizację odwołań do urządzenia i unikanie konfliktów między tymi odwołaniami, które mogą pochodzić np. z dwóch instancji tej samej aplikacji. W przypadku LabView w wersji 7.0 i wyższej możliwe jest jednak bezpośrednio odwoływanie się do portów. Dzieje się tak, gdyż funkcje *In Port.vi* i *Out Port.vi* służące do obsługi portów uruchamiane są przez LabView w trybie jądra, tzn. są uprzywilejowane. Możliwe jest zatem podobnie jak w systemach Windows 95, 98 i Me utworzenie sterownika w LabView, bezpośrednio odwołującego się do urządzenia. Należy jednak pamiętać, że rozwiązanie to obciąża programistę odpowiedzialnością za ewentualne konflikty w odwołaniach.

Cel ćwiczenia

Celem ćwiczenia jest utworzenie pełnego sterownika karty pomiarowej UIA lub UIB poprzez uzupełnienie zbioru istniejących funkcji dostępu do tych kart o funkcję pozwalającą na odczyt wartości napięcia na wejściu kart. Korzystając z tak dokończonego sterownika wykonujący ćwiczenie stworzy w LabView aplikację spełniającą funkcję prostego oscyloskopu o pojedynczym kanale jakim będzie kanał pomiarowy użytej karty. W końcowym etapie ćwiczenia korzystając ze stworzonego przez siebie programu wykonujący ćwiczenie przeprowadzi pomiar indukcji siły elektromotorycznej w sposób w jaki się go wykonuje w doświadczeniu „Badanie prawa indukcji

Faraday'a”.

Aparatura

Tworząc sterownik i aplikację pomiarową (wirtualny oscyloskop) wykonujący ćwiczenie będzie korzystał z generatora sygnału i oscyloskopu. Możliwa jest również generacja sygnału na karcie dźwiękowej komputera. Podczas tworzenia oprogramowania mierzony sygnał może być dowolny np. sinusoidalny. W końcowym etapie ćwiczenia użyty zostanie zestaw eksperymentalny z doświadczenia „Badanie prawa indukcji Faraday'a”. Opis doświadczenia i użytej w nim aparatury znajduje się na pulpicie systemowym komputera. Karty pomiarowe, które należy oprogramować to karty UIA lub UIB zaprojektowane na Wydziale Dydaktyki Fizyki Uniwersytetu w Amsterdamie.

Oprogramowanie kart UIA i UIB

Szczegółowy opis karty pomiarowej UIA znajduje się w II części Podręcznika do programu IP-COACH. Karta UIB, która jest używana w doświadczeniu ma budowę bardzo zbliżoną do UIA. Obie mają cztery kanały pomiarowe. Pozwalają na pomiar napięcia w zakresie 0 .. 5V w kanałach 1 i 2, a 0 .. 1V w kanałach 3 i 4. Karta UIA koduje wartość napięcia w jednym słowie 8 bitowym, podczas gdy UIB koduje sygnał w 12 bitach, z których 8 bitów młodszych jest przekazywanych na inny port niż pozostałe 4 bity. Poniżej przedstawiono krótki opis kroków jakie należy podjąć, aby odczytać sygnał z kart. Na końcu wspomnianego podręcznika można znaleźć przykładowe procedury obsługi kart napisane w języku programowania *pascal*. Karty są już zainstalowane w komputerze i mają przydzielone odpowiednie przerwania i porty.

Zasadniczo pomiar napięcia odczytanego przez karty UIA i UIB wymaga wykonania tych samych czynności tj. przekazania karcie numeru kanału, z którego chcemy odczytać napięcie, a następnie jego odczytu. Komunikacja z kartami odbywa się przede wszystkim poprzez port \$308 (szesnastkowo). Zapis (w *pascalu*)

$$\text{Port}[\$308] := \text{numer kanału} + 3;$$

powoduje, że wartość napięcia na wybranym kanale (numerowanym od 1 .. 4) zostanie zapisana za pomocą odpowiedniej liczby bitów. Odczyt wartości tego napięcia odbywa się poprzez ten sam port. Zatem

$$U := \text{Port}[\$308];$$

spowoduje zapis wartości napięcia, zakodowanej odpowiednią liczbą bitów, w zmiennej U.

Przetworzenie sygnału analogowego na cyfrowy trwa 40 μ s toteż przed odczytem wartości napięcia należy odczekać, aż będzie ona dostępna. Z tego powodu odczyt napięcia nie może zostać sprowadzony wyłącznie do zapisu i odczytu z portu \$308. Funkcja odczytu wartości napięcia, ADCval, powinna mieć następującą postać (zapisaną w *pascalu*):

```
Function ADCval(kanal: Byte): Byte;
Begin
Port[$308] := kanal + 3;           // wybór kanału (1 .. 4)
SelectClockSource(T2, clock);     // wybór źródła taktowania licznika czasowego karty
SetTmrMode(T2, 0);                // ustawienie trybu pracy licznika
WriteCount(T2, 40);               // określenie liczby zliczeń licznika, który zlicza w dół (do 0)
Repeat Until OutLevel(T2) = 1;    // oczekiwanie na koniec zliczania przez licznik impulsów
ADCval := Port[$308];             // odczyt wartości napięcia
End;
```

Jak widać pomiędzy przekazaniem do karty słowa sterującego, decydującego o wyborze kanału pomiarowego, a odczytem wartości napięcia wywoływane są cztery funkcje: SelectClockSource, SetTmrMode, WriteCount i OutLevel. Funkcje te w postaci odpowiednich *VI* można znaleźć w zakładce *User Libraries* w palecie funkcji LabView. Ich kod LabView odpowiada dokładnie kodowi tych funkcji w języku *pascal*, który jest przedstawiony w podręczniku do programu COACH. Nie różni się on znacząco od kodu przedstawionej funkcji ADCval tzn. opiera się również przede wszystkim na zapisie i odczycie różnych wartości na określonych portach. Oczekiwanie na odczyt napięcia realizowane jest w tym przypadku poprzez ustawienie wewnętrznego czasowego licznika karty, który zlicza impulsy aż do chwili gdy odczyt napięcia może już nastąpić. T2 i clock to wartości zmiennych typu wyliczeniowego, a odpowiadają im liczby 2 i 0. W przypadku korzystania z licznika, a zatem używania funkcji SelectClockSource, SetTmrMode, WriteCount i OutLevel konieczne jest zainicjowanie karty funkcją *InitUIAB* dostępną z tej samej palety funkcji co pozostałe.

Wartość zwracana przez funkcję ADCval zakodowana jest w 8 bitach co oznacza, że minimalnej wartości napięcia odpowiada 0, a maksymalnej dostępnej 255. Należy zatem odpowiednio przeskalować zwróconą przez ADCval wartość i uwzględnić fakt, że rzeczywisty sygnał oscyluje wokół zera. Utworzenie sterownika umożliwiającego odczyt napięcia z karty UIA polega zatem wyłącznie na dodaniu do istniejącej biblioteki gotowych funkcji, dodatkowej funkcji

zwracającej wartość napięcia.

W przypadku kart UIB odczyt wartości napięcia może odbyć się z użyciem tej samej funkcji jednak należy pamiętać o tym, że sygnał w tej karcie jest kodowany 12 bitami. Przedstawiona wyżej funkcja ADCval odczytuje wyłącznie 8 młodszych bitów. W efekcie jeżeli sygnał będzie na tyle duży, że przekroczy wartość możliwości zapisu w 8 bitach, funkcja ta zacznie przekazywać błędne wartości. Prawidłowe wartości napięcia można uzyskać jeżeli pobierze się również cztery starsze bity, które są przekazywane poprzez port \$309. Wartość z tego portu należy odczytać przed odczytem wartości z portu \$308, ale po przeprowadzeniu przez kartę konwersji analogowo-cyfrowej (czyli po ostatnim wywołaniu funkcji OutLevel). Następnie oba uzyskane słowa należy połączyć w jedną liczbę za pomocą funkcji *Join Numbers* z palety funkcji LabView. Uzyskana w ten sposób liczba będzie odpowiadała napięciu zakodowanemu 12 bitami. I w tym przypadku konieczne jest odpowiednie przeskalowanie uzyskanej wartości napięcia i jej przesunięcie względem zera.

Aplikacja

W pierwszej kolejności należy stworzyć funkcję ADCval. Mówiąc o funkcji w LabView należy mieć na myśli *subVI*, mogący przyjmować i zwracać jakieś parametry. W przypadku ADCval *subVI* powinien przyjmować jeden parametr, którym jest numer używanego kanału karty i zwracać jedną wartość – napięcie. Bez względu na używaną kartę podłączamy generator do przedwzmacniacza połączonego z kartą. Jeżeli nie wiemy jak powinien wyglądać generowany sygnał podłączamy wcześniej generator do oscyloskopu, aby zapoznać się z kształtem sygnału (a również z oscyloskopem). Tworzymy prosty program w LabView, który korzystając z przygotowanej funkcji ADCval, wyświetla najpierw odczytywaną z karty wartość na wskaźniku (by się przekonać, że coś rejestruje), następnie na wykresie. W dalszym kroku należy dopracować funkcję ADCval tak, aby zwracała rzeczywistą wartość napięcia, uwzględniając wszystkie czynniki wymienione w poprzednim rozdziale.

Modyfikujemy aplikację tak, aby zbierała dane z karty do tablicy (napięć) o arbitralnie wybranej wielkości, która po wypełnieniu powinna być wyświetlana na wykresie. Każdemu elementowi tablicy napięć powinien zostać przypisany odpowiadający mu czas odczytu. Czas można w LabView wyznaczyć za pomocą funkcji *Tick Count (ms)*. Funkcja ta określa go jednak z dokładnością milisekund. Rozdzielczość taka jest zdecydowanie zbyt mała w wielu zastosowaniach. Dlatego należy wykorzystać bardziej rozbudowaną wersję tej funkcji, dostępną na stronie firmy National Instruments, o mikrosekundowej zdolności rozdzielczej. Funkcja ta również nazywa się *Tick Count* i jest dostępna z palety *User Libraries* lub z pulpitu systemowego. Określenie chwili czasu jakiej odpowiada dany element tablicy wartości napięć można zrealizować na dwa sposoby.

Pierwszy, prostszy, polega na odczycie czasu w jakim rozpoczęto i zakończono akumulację danych. Następnie wartości tych czasów są odpowiednio wykorzystane przy obliczeniu czasów odpowiadających kolejnym elementom tablicy (której rozmiar jest ustalony). Druga metoda polega na określaniu czasu podczas każdego aktu odczytu wartości napięcia z karty. Wymaga to zatem wypełniania wartościami dodatkowej tablicy czasów o takim samym rozmiarze jak tablica napięć. Zamiast dwóch tablic można również skorzystać z tablicy klastrów (*cluster*, rekordy, struktury). Zadaniem wykonującego ćwiczenie jest wybór lepszej metody i jego uzasadnienie.

Korzystając z tablicy napięć i czasów odpowiadających kolejnym wartościom napięcia należy doprowadzić aplikację do stanu, w którym będzie wyświetlać na wykresie zależność odczytywanego na karcie napięcia od czasu, przy czym przebieg sygnału powinien być zgodny z oczekiwaniami. Zatem np. w przypadku sygnału sinusoidalnego o częstotliwości 10 Hz na wykresie w przedziale 1s powinno być widocznych 10 pełnych przebiegów fali.

Zauważmy, że ilość wyświetlanych na wykresie danych zależy od liczby elementów tablicy, w której są one akumulowane. Przedział czasu jakiemu odpowiadają wyświetlane wartości zależy w takiej sytuacji od czasu dzielącego dwa kolejne pomiary. Najczęściej próbując zebrać jakiś sygnał na oscyloskopie dysponujemy szacunkową wiedzą o czasie w jakim interesujące nas zjawisko zachodzi. Dlatego w oscyloskopach jednym z podstawowych parametrów jest tzw. podstawa czasu określająca przedział czasu jakiemu ma odpowiadać wyświetlany na wykresie sygnał.¹ Tworzona aplikacja powinna mieć podobną cechę co oznacza, że o ilości wyświetlanych na jej wykresie danych nie decyduje ustalona z góry liczba elementów tablicy napięć (i czasów²), ale całkowity czas pomiaru. Powinien być on określany za pomocą pokrętła, jednak możliwe powinno być określenie dowolnej jego wartości (z dokładnością nie większą niż możliwości stosowanej karty i komputera, które należy określić). Oznacza to, że przy stałej wartości czasu dzielącej kolejne dwa pomiary napięcia, liczba elementów tablicy napięć będzie zmienna, zależna od użytej podstawy czasu. Pomiar powinien zatem przebiegać w sposób następujący: po ustaleniu podstawy czasu i uruchomieniu pomiaru, powinien zostać odczytany czas początkowy po czym dane mają być zbierane do tablicy napięć. Czas musi w takiej sytuacji być odczytywany przy każdym odczycie wartości napięcia, gdyż cała akumulacja powinna zostać przerwana w chwili gdy jej całkowity czas trwania osiągnie podstawę czasu. Gdy tablica zostanie wypełniona, jej zawartość zostanie wyświetlona na ekranie, a następnie cały pomiar będzie kontynuowany.

Kontynuacja pomiaru oznacza stałe odświeżanie zawartości wykresu z częstotliwością równą w przybliżeniu odwrotności podstawy czasu. W przypadku powtarzalnego sygnału, takiego jak sinusoidalny, odświeżanie to jest wskazane, gdyż pozwala stwierdzić czy sygnał nie ulega

¹ Podstawa czasu określa w oscyloskopach najczęściej czas przypadający na jedną główną podziałkę skali czasu. Dla uproszczenia będziemy jednak przez podstawę czasu rozumieć cały okres czasu w jakim następuje pomiar.

² W dalszej części opisu przez tablicę napięć rozumiane będą obie tablice, napięć i czasów, lub jedna tablica klastrów tych dwóch wielkości.

zmianie. Jednak jeżeli obserwowany przez nas sygnał nie jest tworzony przez generator, lecz jest jednorazowy i zanika z czasem, wówczas choć w pewnej chwili zostanie on wyświetlony na wykresie, w kolejnym przebiegu akumulacji zniknie. Stanie się tak, gdyż tablica napięć zostanie wypełniona wartościami oscylującymi wokół zera (szumem). Rozwiązaniem tego problemu jest wprowadzenie do procesu akumulacji danych tzw. progu wyzwolenia. Wielkość ta określa minimalną wartość napięcia jaka musi zostać zarejestrowana na karcie, aby proces wypełniania tablicy napięć został rozpoczęty. Jeżeli wartość odczytywanego napięcia przekroczy próg wyzwolenia odczytywany jest czas, po czym następuje opisywana wyżej akumulacja przez okres wyznaczony podstawą czasu. W oscyloskopach próg wyzwolenia ustala się zazwyczaj pokrętłem i najczęściej reprezentuje tę wielkość na wykresie krótka, pozioma kreska poruszająca się po lewej lub prawej osi wykresu. Jego wartość często jest wyświetlana na ekranie oscyloskopu. W tworzonej aplikacji może być ona wyświetlana na niezależnym wskaźniku (który można położyć na wykresie).

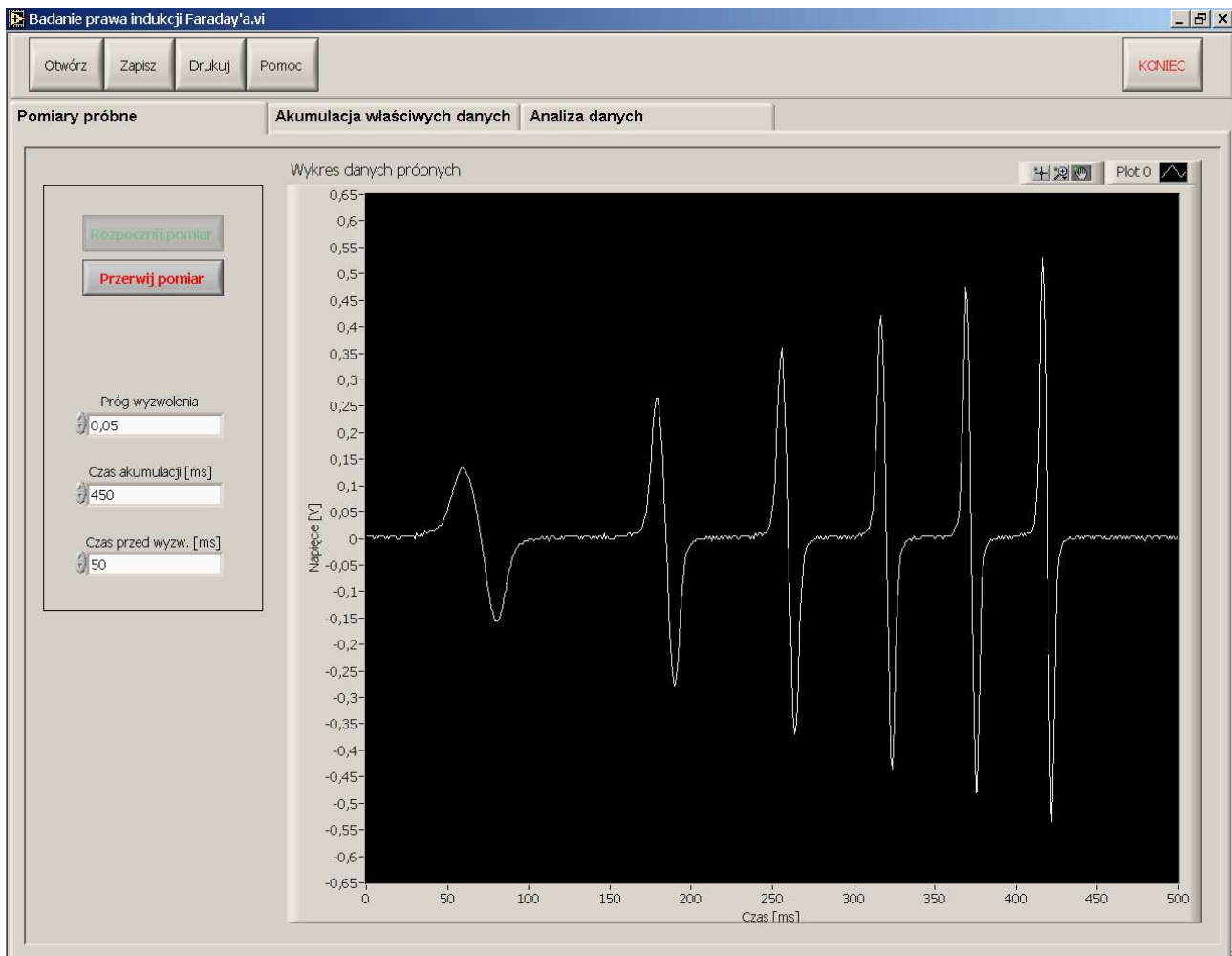
Ostatnią ważną cechą związaną z akumulacją danych, jaką musi spełniać tworzona aplikacja jest możliwość określania czasu akumulacji przed wyzwoleniem. Jeżeli badany sygnał pojawia się znacznie rzadziej niż wykonywana jest akumulacja danych w tablicy napięć, wówczas konieczne jest ustalenie wartości progu wyzwolenia. Gdy zbierzemy już interesujący nas sygnał, wówczas jego przebieg przedstawiony na wykresie zawsze zaczyna się od wartości nie mniejszych od progu. Nie jest zatem możliwe przedstawienie przebiegu sygnału z okresu czasu przed chwilą wyzwolenia. Tymczasem bardzo często ten przebieg jest interesujący. *Głównym zadaniem jakie obok stworzenia sterownika urządzenia, stoi przed wykonującym ćwiczenie jest opracowanie prostego algorytmu, który pozwoli na pomiar dowolnego sygnału z zastosowaniem progu wyzwolenia, ale również z możliwością obejrzenia na wykresie dowolnie długiego przebiegu sygnału przed wyzwoleniem pomiaru.*

Aplikacja, która spełnia powyższe założenia powinna posiadać jeszcze kilka dodatkowych możliwości zwiększających jej funkcjonalność. Większość oscyloskopów zaopatrzona jest w proste funkcje matematyczne pozwalające na analizę obserwowanego sygnału. Podobne funkcje należy dodać do tworzonej aplikacji. Przede wszystkim jej wykres powinien być zaopatrzony w dwa kursory, pozwalające na określanie zakresu w jakim mają zostać przeprowadzone obliczenia na zebranych danych. Użytkownik powinien mieć możliwość wyboru rodzaju kursorów: wertykalnych lub horyzontalnych. Standardowo na wskaźnikach powinny być wyświetlane aktualne położenia kursorów (wartości czasów lub napięć) oraz różnica ich położenia. W zależności od wybranej funkcji matematycznej, w dodatkowym wskaźniku powinien być wyświetlany wynik obliczeń wykonanych na danych z zakresu wskazanego przez kursory. Podstawową funkcją powinno być określenie pola powierzchni pod(nad) krzywą sygnału. Podobnie jak w oscyloskopie aplikacja powinna mieć również możliwość przesuwania obserwowanego sygnału w pionie o wartość tzw. *offset'u* określaną

za pomocą pokrętła. Należy również dodać do aplikacji możliwość zapisu zebranych danych do pliku, w formacie przez siebie wybranym.

Eksperyment

Po zapoznaniu się z opisem doświadczenia „Badanie prawa indukcji Faraday'a” należy przeprowadzić próbę rejestracji sygnału indukowanej siły elektromotorycznej za pomocą stworzonego przez siebie wirtualnego oscyloskopu. W tym celu należy podłączyć wyjście cewek nawiniętych na plastikowej rurce do wejścia przedwzmacniacza, dobrać czas akumulacji danych na 450 ms, a czas przed wyzwoleniem na 50 ms, prowadzić rejestrację sygnału indukowanego przez spadający w rurce magnes zmieniając próg wyzwolenia tak, aby uzyskać przebieg podobny do widocznego na rysunku 2.



Rysunek 2. Przykładowy sygnał indukowany przez magnes spadający w rurce, na której nawinięto sześć cewek, połączonych szeregowo do karty pomiarowej.

Podsumowanie – główne zadania

1. Utworzenie funkcji sterownika do karty pomiarowej, odczytującej napięcie na wejściu karty.
2. Utworzenie aplikacji przedstawiającej na wykresie w zadanym przedziale czasu zależność rejestrowanego na karcie napięcia od czasu. Uzasadniony wybór metody określania czasu odpowiadającego kolejnym wartościom napięcia.
3. Określenie zdolności rozdzielczej używanej karty pomiarowej na danym komputerze.
4. Dodanie do aplikacji możliwości określania progu wyzwania, czasu przed akumulacją i *offset'u*.
5. Dodanie do aplikacji możliwości zapisu zebranych danych w pliku.