

1.1. Odczyt kodu QR

W przypadku pierwszego uruchomienia jednej z metod, bądź chęci załadowania do pamięci nowego filtra operator musi zeskanować kod QR za pomocą wybranej kamery. Czynność ta odbywa się w programie głównie w trzech i paru dodatkowych stanach. Początkowo w stanie *Action: Clear Data* „czyszczony” jest zapisany kod QR wraz z innymi danymi wyświetlanymi na panelu czołowym. Następnie wyłączany jest przycisk *Graph Tab*, tak aby na czas akwizycji obrazu operator nie mógł przenieść się na zakładkę z wykresem, przez co musiałby powrócić do poprzedniej zakładki. W kolejnym stanie, czyli *Action: Camera Initialization* wywoływane jest utworzone subVI odpowiedzialne za włączenie kamery wraz z alokacją tymczasowej pamięci na rejestrowany obraz. Następnie w stanie *Action: Read QR* rejestrowany jest obraz wraz z próbą znalezienia kodu QR na nim i jego poprawnym odczytem. Pobierany obraz zamieniany jest na przestrzeń 8-bit (szary), a następnie przekazywany do gotowej funkcji z biblioteki *IMAQ*. Funkcja ta wymaga takiego formatu obrazu. W przypadku znalezienia kodu QR i poprawnego odczytania jego wartości, rysowana jest ramka wokół niego, dzięki czemu operator może sprawdzić, który z kodów został odczytany w przypadku ich większej ilości obejmowanej przez kamerę. Każdorazowo po tym etapie aplikacja przechodzi do stanów *Event Structure* (w celu zapewnienia responsywności) oraz *UI: Update* (w celu wyświetlenia rejestrowanego obrazu). W przypadku gdy w tym stanie nie uda się odczytać kodu, to dodawane jest jeszcze ponownie jego wywołanie. Zatem stan ten wywołuje się tak długo, aż użytkownik tego nie przerwie lub nie zostanie poprawnie odczytany kod. Każdorazowo czyszczony jest także kod błędu -1074395318 wskazujący na brak sukcesywnego odczytu kodu. Zależnie od powodzenia operacji wyświetlana jest za pośrednictwem Status Bar’a informacja użytkownikowi. W ostatnim z głównych stanów, a mianowicie *Action: Camera Stop* następuje wyłączenie kamery wraz ze zwolnieniem pamięci przeznaczonej na rejestrowany obraz. Na koniec załączany jest przycisk, dzięki któremu operator może wyświetlić zakładkę z wykresem (do której jest automatycznie przenoszony po pomyślnym odczycie kodu QR) na tzw. *Control Tab* za pośrednictwem stanu *UI: Disable*.

1.2. Odczyt konfiguracji z bazy danych

Każdorazowo niezależnie od wybranej metody testowania program odczytuje konfigurację z bazy danych. Czynność ta odbywa się w stanie *Action: Read from Database*, po zakończonym z sukcesem odczycie kodu QR. Do zrealizowania tego zadania wykorzystano bibliotekę *SQLite* wraz z programem tej samej firmy o nazwie *SQLite Expert Personal 5 – 64bit* pozwalającym na tworzenie bazy danych i zarządzaniu nią za pośrednictwem interfejsu graficznego. Początkowo otwierane jest połączenie z bazą danych o nazwie *tfa_db.db* znajdującego się w lokalizacji projektu. W przypadku gdy plik taki nie istnieje wyświetlane jest okno dialogowe, w celu wybrania odpowiedniego pliku przez użytkownika. Następnie wysyłane jest zapytanie SQL, gdzie wyszukiwany jest odpowiedni wiersz na podstawie kodu QR, który jest równoważny z ID filtra. Odczytywane dane przypisywane są do odpowiednich zmiennych w klastrach. W bazie danych wyszukiwane są:

- a) Informacje opisowe o filtrze, takie jak typ filtra i częstotliwości odcięcia, które są składowymi definicji typu o nazwie *Filter description.ctf*.
- b) Część konfiguracji karty DAQ, a mianowicie: amplituda generowanego sygnału, ilość generowanych próbek dla sinusoidy oraz sygnału szerokopasmowego, ilość cykli sinusoidy, częstotliwość próbkowania sygnału szerokopasmowego oraz jego opóźnienie.
- c) W postaci tekstu sekwencje dla sygnału sinusoidalnego: częstotliwość, częstotliwość próbkowania, ilość próbek do akwizycji.
- d) Ścieżka do pliku zawierającego limity dla danego filtra.

Całość powyższych operacji odbywa się w jednym subVI, w którym zdefiniowano również generowanie własnego błędu o numerze 5001 w przypadku, gdy baza danych nie posiada filtra o podanym ID. Następnie w tym samym stanie znajduje się kolejne subVI, w którym przekształcana jest sekwencja metody sinusoidalnej w postaci tekstu na tablicę o typie *double*. Jak pokazano na Rys. 1 kolejne wartości częstotliwości sinusoidy, częstotliwości próbkowania oraz ilości próbek do akwizycji (kolumny *Sin_Frequencies*, *Sin_Sample_Rates*, *Sin_AI_Samples*) podawane są w bazie danych kolejno po sobie oddzielone separatorem ‘;’. W bloku funkcyjnym rozdzielane są kolejne wartości, przekształcane na typ *double* i dodawane do tablicy. Tablica ta jest zapisana, tak że kolumny oznaczają kolejne wartości, a trzy wiersze oznaczają wcześniej wspomniane parametry.

row_ID	Type	Freq_Cutoff	Amplitude	Sin_Samples	Sin_Cycles	Sin_Frequencies	Sin_Sample_Rates	Sin_All_Samples	Broadband_Samples	Broadband_Sample_Rate	Broadband_Delay	Limits_path
1 000001	Band-stop 1st. orde	50-235 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	51200	51200	200	limits\filtr_0
2 000002	Band-stop 2nd. ord	50-235 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	51200	51200	200	limits\filtr_0
3 000003	Band-pass 1st. orde	50-235 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	51200	51200	200	limits\filtr_0
4 000004	Band-pass 2nd. ord	50-235 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	51200	51200	200	limits\filtr_0
5 000005	High-pass 1st. orde	50 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	16384	51200	200	limits\filtr_0
6 000006	High-pass 2nd. ord	50 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	16384	51200	200	limits\filtr_0
7 000007	Low-pass 1st. orde	50 [Hz]	1	100	1	5;10;15;20;25;30	400;800;1200;1600	1000;1000;1000;	16384	51200	200	limits\filtr_0
8 000008	Low-pass 2nd. o	50 [Hz]	1	100	1	5;10;15;20;25	400;800;1200;1	1000;1000;1000;	16384	51200	200	limits\filtr_0

Rys. 1 Utworzona baza danych w programie SQLite Expert Personal 5.3 64-bit

Podobnie jak w czynności odczytywania kodu QR opisanej w rozdziale 1.1, tutaj też użytkownik informowany jest o wykonywanej czynności za pośrednictwem Status Bar'a.

1.3. Wczytywanie wartości granicznych dla filtrów

W przypadku ładowania nowego filtra do pamięci lub pierwszego uruchomienia jednej z metod po odczytaniu wszelkich danych z bazy wywoływany jest stan *Action: Read Limits*. W stanie tym na podstawie ścieżki do pliku tekstowego, która przechowywana jest w bazie danych, wczytywane są do tablic wartości graniczne dla danego filtra. Zawartość przykładowego pliku z limitami pokazano na Rys. 2.

Plik	Edycja	Format	Widok	Pomoc
0	5	-5		
30	5	-5		
40	4	-6		
50	1	-9		
100	-2	-19		
1000	-12	-49		
10000	-22	-79		
25000	-26	-90		

Rys. 2 Przykładowa zawartość pliku z ograniczeniami filtra

W plikach z limitami zawarte są trzy kolumny. Pierwsza z nich to wartości częstotliwości w hercach, czyli współrzędne X punktów. Następne dwie kolumny to wartości tłumień odpowiednio dla górnego i dolnego limitu podane w decybelach. Taka struktura pliku pozwala użytkownikowi na szybkie i przejrzyste konstruowanie ograniczeń poprzez zdefiniowanie pola za pomocą punktów, w którym powinna mieścić się odpowiedź filtra, by uznać ją za prawidłową. Aby odczytać zawartość pliku wykorzystano gotową funkcję *Read Delimited Spreadsheet.vi*. Następnie dane wpisywane są do klastra zapisanego jako definicja typu *Limits.ctf*. Ponownie, jak w przypadku odczytywania danych z bazy, w przypadku

gdy nie istnieje plik o podanej nazwie to użytkownikowi wyświetli się okno dialogowe, gdzie będzie mógł go „ręcznie” wybrać.

1.4. Generowanie sygnału sinusoidalnego o dyskretnych wartościach częstotliwości

W przypadku wybrania przez użytkownika metody testowania w postaci generowania kolejnych sygnałów sinusoidalnych o dyskretnych wartościach częstotliwości całość akcji odbywa się w 9 różnych stanach. Czynność tę podzielono na tyle „części” ze względu na zapewnienie skalowalności aplikacji i zwiększenia przejrzystości kodu programu.

Na początku w stanie *Action: Sin Extract Freq* odczytywane są częstotliwości, którymi będzie testowany filtr. Brany jest pierwszy rząd z tablicy zawierającej kolejne sekwencje i wpisywany jest on do klastra przeznaczonego dla obliczonych tłumień (jako współrzędne X). Następnie w stanie *DAQ Sinus: Reconfigure Sequence* odczytywana jest pierwsza sekwencja testująca, czyli częstotliwość sinusoidy, częstotliwość próbkowania i ilość próbek do akwizycji. Odbywa się to na zasadzie odczytania pierwszej kolumny z tablicy zawierającej całą sekwencję i przypisaniu wartości do zmiennych. W stanie *AI: Configure* konfigurowany jest czas poboru próbek, w tym przypadku jest to maksimum 0,5 sekundy. Następnie w *DAQ Sinus: AO Configure and Start* ustawiane są parametry karty DAQ pod względem generowania sygnału. Z wykorzystaniem funkcji *Basic Function Generation.vi* tworzony jest wzór sinusoidy o zadanej amplitudzie i częstotliwości. Taki sygnał wprowadzany jest do funkcji *DAQmx Write.vi* w celu wygenerowania przez kartę DAQ zadanego przebiegu. W tym stanie zaaplikowane jest również oczekiwanie 0,5 sekundy w celu ustabilizowania się generowanego sygnału. Następnie w stanie *DAQ Sinus: AI Acquire* wykonywana jest akwizycja jednokrotna, czyli zarejestrowanie zadanej ilości próbek, z zadaną częstotliwością na danym wejściu karty DAQ. Akwizycja ustawiona jest na tryb *Finite Samples* zatem pobierana będzie określona ilość próbek, ale nie dłużej niż 0,5 sekundy, jak to zostało skonfigurowane w stanie *AI: Configure*. Wyznaczona odpowiedź filtra zapisywana jest w odpowiednim klastrze jako typ *Waveform*. W stanie *DAQ Sinus: AO Stop and Clear* wyłączane i czyszczone jest zadanie generowania sygnału przez kartę DAQ. Następnie w stanie *DAQ Sinus: Calculate* obliczana jest amplituda odpowiedzi filtra, ale przed tym usuwane jest pierwsze

100

próbek,

w celu kolejnego zabezpieczenia się przed nieustabilizowaną odpowiedzią. Po wyznaczeniu amplitudy dopisywana jest ona do tablicy. Po tej operacji w następnym stanie (*DAQ Sinus Rem 1st Column from Seq.*) usuwana jest pierwsza kolumna z tablicy zawierającej całą badaną sekwencję. Dalsze działanie programu zależne jest od rozmiaru tej tablicy. W przypadku gdy jest ona różna od zera to, wszystkie czynności zaczynając od stanu *DAQ Sinus: Reconfigure Sequence* wykonywane są ponownie. Przed nimi dodawane są dwie instancje, a mianowicie *Event Structure* w celu zapewnienia responsywności aplikacji oraz *UI: Update*, by wyświetlić użytkownikowi nową wykonywaną sekwencję. W przypadku gdy rozmiar tablicy jest równy zero, to oznacza, że nie pozostały żadne sygnały do wygenerowania, zatem program przechodzi do stanu *Action: Sin Calculate Dampings*, w którym obliczane są tłumienia dla konkretnych częstotliwości na podstawie wyznaczonych amplitud. Zatem po tym stanie otrzymywany jest klaster zawierający dwie tablice, w której jedna z nich to testowane częstotliwości, a druga to wyznaczone tłumienia. Mając takie dane możliwe jest wyświetlenie charakterystyki częstotliwościowej (wykresu Bode'go) na wykresie.