Pracownia Specjalistyczna – ROOT
Zadanie 1

W katalogu Pliki/Class Materials/zadanie_1 znajdziecie plik wave0.dat. Jest to plik binarny zawierający dane z detektora scyntylacyjnego. Dane te mają postać całych sygnałów zapisanych z czasem próbkowania 1 ns. Okno czasowe jednego sygnału to 1024 ns. Liczby zapisane w pliku binarnym to liczby zmiennoprzecinkowe typu float. Każda z liczb to wartość amplitudy sygnału w danej chwili, przy czym jest ona zapisana w jednostkach arbitralnych – kanałach ADC. Aby przekonwerotwać tę wartość na jednostki ficzyczne trzeba wiedzieć, że w układzie akwizycji danych, który zarejestrował te dane 1 mV odpowiada 4.096 ADC.

Napiszemy makro ROOT, które pozwoli na przeglądanie histogramów jeden po drugim:

1. Niech makro bierze jako argument nazwę pliku, który należy otworzyć. (1 pkt)

2. Otwieraj podany plik do odczytu, z założeniem że będzie to plik binarny. Pamiętaj o sprawdzeniu, czy plik został poprawnie otwarty. (2 pkt)

3. Stwórz histogram, w którym rysowane będą kolejne sygnały, oraz canvas, na którym rysowany będzie histogram. Zadbaj o nadanie odpowiednich tytułów osiom oraz histogramowi. (2 pkt)

4. W pętli while() odczytuj dane z pliku i wypełniaj nimi histogram. Po odczytaniu każdych kolejnych 1024 liczb otrzymujemy pełen obraz zarejestrowanego sygnału. Przy wypełnianiu pamiętaj o przeliczeniu odczytanych wartości (w ADC) do mV. (2 pkt)

5. Po odczytaniu 1024 liczb z pliku chcielibyśmy obejrzeć histogram, aby zobaczyć nasz sygnał, a następnie przejść do następnego sygnału, itd. W klasie TPad zaimplementowano funkcję WaitPrimitive(), która „zatrzymuje” działanie programu do momentu aż pojawi się obiekt o zadanej nazwie. Przejście do kolejnego przebiegu plętli wywołujemy przez podwójne kliknięcie na canvas. Użyjcie tej funkcji w swoich programach. Dzięki temu Wasz program umożliwi przeglądanie sygnałów, jeden po dugim, po prostu przez klikanie na canvas. W dokumentacji funkcji WaitPrimitive() znajdziecie również przykład jej użycia w kodzie. (1 pkt)

6. Po przejrzeniu kilku sygnałów zauważycie, że linia bazowa nie wypada w zerze, ale na jakiejś niezerowej wartości. Dopisz do programu opcję (np. przez dodadnie dodatkowego argumentu funkcji), określającego czy odejmować linię bazową czy nie. Odejmowanie linii bazowej polega na wyznaczeniu poziomu linii bazowej i odjęciu tej wartości od wszystkich próbek w sygnale. Wartość linii bazowej możemy wyznaczyć np. jako średnią z pierwszysch 50 próbek w oknie czasowym sygnału. Za każdym razem, gdy odejmujesz linię bazową wypisuj (na ekranie lub na canvasie) jaka wartość została odjęta. (2 pkt)