

# LAB 3

## HEART RATE MONITOR

### CLUSTERS, FILES, SUBVI

Przydatne skróty klawiaturowe:

<b>Ctrl+S</b>	zapis pliku
<b>Ctrl+Shift+S</b>	zapis wszystkich otwartych programów
<b>Ctrl+R</b>	uruchomienie programu
<b>Ctrl+E</b>	nawigacja pomiędzy oknami Block Diagram/Front Panel
<b>Ctrl+A</b>	zaznaczenie wszystkich elementów
<b>Ctrl+Shift+A</b>	wyrównanie elementów block diagramu lub front panelu
<b>Ctrl+D</b>	równomierna dystrybucja elementów
<b>Ctrl+H</b>	pomoc kontekstowa
<b>Ctrl+B</b>	usunięcie nieprawidłowych połączeń
<b>Ctrl+Shift+right click</b>	tools palette
<b>Ctrl+Space</b>	quick drop
<b>Ctrl+N</b>	nowy plik

Efektom dzisiejszych zajęć powinien być uporządkowany kod, dane zamknięte w struktury danych oraz nowa funkcjonalność pozwalająca na wczytanie danych pomiarowych z pliku.

Utwórz klaster 'HR Data Structure', który uporządkuje i zgrupuje dane przechowywane w programie (rys. obok).

Zapisz każdy z klastrów jako typedef (Right Click/Make Type Def, Right Click/Open Type Def, Save). Używanie kontrolki typu typedef we wszystkich podprogramach pozwoli na automatyczną aktualizację zmian we wszystkich instancjach 'HR Data Structure'.

Za pomocą funkcji 'bundle by name' i 'unbundle by name', będziesz miał dostęp do danych zawartych w klastrze.

HR Data Structure

Runner's Data

Name

Age

HR Max

Raw Data

Time

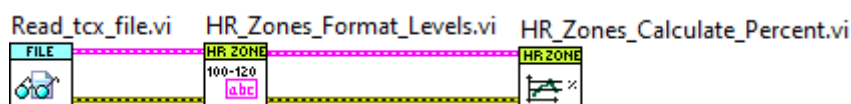
Bpm

% HR MAX

HR Zones

Heart Rate Zone No

Na ostatnich laboratoriach zaimplementowałeś kod pozwalający na formatowanie zakresów danych do tablicy string, oraz wyliczenie procentowe czasu spędzonego w każdej ze stref tętna. Teraz te fragmenty kodu zamknij do funkcji (SubVI), żeby uporządkować kod (zaznacz fragment kodu / Edit / Create SubVI). Pamiętaj o zdefiniowaniu wejść, wyjść, klastrów błędów, zdefiniowaniu terminali w Connector Pane oraz utworzeniu odpowiednich ikon, np:

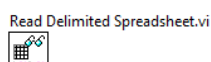



Standardowe podłączenie terminali:



W tekstowym pliku activity\_2.tcx znajdują się dane pomiarowe z jednego z treningów. Otwórz ten plik w edytorze tekstowym i sprawdź jego strukturę. Interesujące nas dane oznaczone są jako: 'Time' oraz 'HeartRateBpm'.

```
<Extensions>
<ns3:TPX>
  <ns3:Speed>1.781999945640564</ns3:Speed>
  <ns3:RunCadence>84</ns3:RunCadence>
</ns3:TPX>
</Extensions>
</Trackpoint>
<Trackpoint>
  <Time>2024-08-14T05:13:27.000Z</Time>
  <Position>
    <LatitudeDegrees>50.03753394819796</LatitudeDegrees>
    <LongitudeDegrees>19.9655238725245</LongitudeDegrees>
  </Position>
  <AltitudeMeters>212.1999969482422</AltitudeMeters>
  <DistanceMeters>19.100000381469727</DistanceMeters>
  <HeartRateBpm>
    <Value>92</Value>
  </HeartRateBpm>
</Trackpoint>
<Extensions>
  <ns3:TPX>
    <ns3:Speed>2.3329999446868896</ns3:Speed>
```



Odczytaj plik tekstowy, np. przy użyciu funkcji  i wczytaj wybrane dane do tablicy 'bpm' i 'time'. Zapisz kod jako kolejny podprogram (subVI).

Wszystkie subVI powinny mieć zdefiniowane:

- wejście: kontrolki HR Data Structure, Error In,

- wyjście: indykatory HR Data Structure, Error Out

(oraz ewentualnie w zależności od implementacji dodatkowe wejścia / wyjścia).

Front panel głównego programu będzie analogiczny do poprzedniego etapu projektu, dodajemy jedynie klaster 'Runner's Data'. Przykładowy block diagram głównego programu może wyglądać następująco.

