

## Biblioteka lapoDlib.dll

Wymaga zainstalowanego sterownika do LAPO-RTU oraz biblioteki libusb0.dll

### Dostępne funkcje

- InitDevice
- getTextWidth
- sendPassing
- sendCurrentPos
- sendFlag
- sendPicture
- setRPMLimit
- sendMessage
- readData
- sendData
- closeDevice
- readDataWithCallback
- sendDataToRemoteControl
- sendMessage2
- sendMessage4
- GetLibVersion

### **InitDevice**

function InitDevice: pointer; stdcall;

Funkcja ta inicjalizuje połączenie z LAPO-RTU. Zwracany jest wskaźnik na uchwyt urządzenia niezbędny do wywołania pozostałych funkcji. Jeśli w wyniku funkcji zwrócona zostanie wartość nil oznacza to błąd inicjalizacji

### **getTextWidth**

function getTextWidth(ffont:integer;t:array of byte): integer; stdcall;

Funkcja zwraca szerokość tekstu w pikselach dla podanej czcionki. Może być używana do określenia miejsca gdzie należy wypisać kolejny tekst przy składaniu napisów z kilku tekstów (np. napis dłuższy niż 14 znaków nie może być przesłany jako pojedynczy napis i musi zostać złożony)

Parametry wejściowe:

- ffont - czcionka (patrz dostępne czcionki)
- t - tekst do wyświetlenia. Wartość 0 na 1 bajcie oznacza brak tekstu (Maksymalnie 14 znaków)

Wartość zwracana: szerokość tekstu w pikselach

### **sendPassing**

function

sendPassing(udev:pointer;faddr:integer;fpos:integer;ftime:integer;diff1st:integer;better:integer;mybesttime:integer): integer; stdcall;

Funkcja ta powinna być wywołana po przejechaniu pojazdu przez linię mety. Parametry wejściowe

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- faddr – adres laptimer – 2 młodsze bajty numeru seryjnego laptimer UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- fpos – pozycja pojazdu w wyścigu
- ftime – czas okrążenia , które właśnie zostało ukończone. Czas powinien być podany jako \*100 czyli czas 42.67 należy przesłać jako 4267

- diff1st – różnica czasu ftime w stosunku do najlepszego czasu w sesji. Wartość ujemna oznacza że aktualny czas jest nowym najlepszym czasem sesji
- better – flaga określająca czy ftime to nowy najlepszy czas danego kierowcy czy nie. Wartość 1 oznacza poprawienie własnego najlepszego czasu
- mybesttime – najlepszy czas danego kierowcy w sesji

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

### **sendCurrentPos**

function

sendCurrentPos(udev:pointer;faddr:integer;fpos:integer;ftime:integer;diff1st:integer;mybesttime:integer): integer; stdcall;

Funkcja odświeża dane o pozycji kierowcy w wyścigu. Powinna być wywołana po przejechaniu przez linię mety dla pozostałych pojazdów na torze w celu uaktualnienia ich pozycji. Parametry wejściowe

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- faddr – adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- fpos – pozycja pojazdu w wyścigu
- ftime – czas który ma być wyświetlony na ekranie głównym. Może to być najlepszy czas kierowcy w sesji albo czas ostatniego okrążenia
- diff1st – różnica czasu ftime w stosunku do najlepszego czasu w sesji. Wartość ujemna oznacza że aktualny czas jest nowym najlepszym czasem sesji
- mybesttime – najlepszy czas danego kierowcy w sesji

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

### **sendFlag**

function sendFlag(udev:pointer;faddr:integer;f:integer): integer; stdcall;

Funkcja wysyła flagę. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- faddr – adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- f – numer flagi. 1- szachownica; 2- żółta; 3- niebieska; 4- czerwona; 5 – zielona; 6- biała ; 7- czarna

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

### **sendPicture**

function sendPicture(udev:pointer;faddr:integer;no:integer;blink:integer): integer; stdcall;

Funkcja wysyłająca rozkaz wyświetlenia obrazka z biblioteki graficznej. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- faddr - adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- no – numer grafiki w bibliotece

- blink – jeśli 1 oznacza że ekran ma efekt migania

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

### setRPMLimit

function setRPMLimit(udev:pointer;faddr:integer;frpmlimit:word): integer; stdcall;

Funkcja wysyła ograniczenie obrotów. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- addr - adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- frpmlimit wartość ograniczenia w obrotach na minutę. Wartość 0 znosi ograniczenie.

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

### sendMessage

function

sendMessage(udev:pointer;faddr:integer;fnrback:integer;ftimeBack:integer;FtextOnly:integer; ft1:array of byte;fcolor1:integer;fbcolor1:integer;fx1,fy1:word;ffont1,ftransp1,fblink1:byte; ft2:array of byte;fcolor2:integer;fbcolor2:integer;fx2,fy2:word;ffont2,ftransp2,fblink2:byte; ft3:array of byte;fcolor3:integer;fbcolor3:integer;fx3,fy3:word;ffont3,ftransp3,fblink3:byte ): integer; stdcall;

Funkcja pozwala na wyświetlenie obrazka i 3 teksty. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- addr - adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- fnrback – numer obrazka z biblioteki graficznej (UWAGA! 0 wyświetla ekran główny z obrotościomierzem)
- ftimeBack- czas w sekundach przez jaki ma być ekran wyświetlany. Po tym czasie LAPO-D powróci do ekranu głównego. Wartość 0 spowoduje wyświetlenie ekranu na stałe do czasu wysłania innego rozkazu
- FtextOnly – dla 0 wyświetlony zostaje zarówno ekran jak i teksty. Dla 1 ekran pozostaje taki jak był a jedynie dopisywane są teksty. Flaga ta może być używana do dodawania kolejnych tekstów na ekran
- ft1 – tekst do wyświetlenia. Wartość 0 na 1 bajcie oznacza brak tekstu (Maksymalnie 14 znaków)
- fcolor1- kolor tekstu
- fbcolor1 – tło napisu (istotne tylko jeśli flaga przezroczystości nie będzie zaznaczona)
- fx1 – współrzędna x tekstu
- fy1 – współrzędna y tekstu
- ffont1 – czcionka (patrz dostępne czcionki)
- ftransp1 – flaga przezroczystości tła tekstu
- fblink1 – wartość 1 spowoduje miganie tekstu
- ft2 – tekst do wyświetlenia. Wartość 0 na 1 bajcie oznacza brak tekstu (Maksymalnie 14 znaków)
- fcolor2- kolor tekstu
- fbcolor2 – tło napisu (istotne tylko jeśli flaga przezroczystości nie będzie zaznaczona)
- fx2 – współrzędna x tekstu
- fy2 – współrzędna y tekstu
- ffont2 – czcionka (patrz dostępne czcionki)

- ftransp2 – flaga przezroczystości tła tekstu
- fblink2 – wartość 1 spowoduje miganie tekstu
- ft3 – tekst do wyświetlenia. Wartość 0 na 1 bajcie oznacza brak tekstu (Maksymalnie 14 znaków)
- fcolor3- kolor tekstu
- fbcolor3 – tło napisu (istotne tylko jeśli flaga przezroczystości nie będzie zaznaczona)
- fx3 – współrzędna x tekstu
- fy3 – współrzędna y tekstu
- ffont3 – czcionka (patrz dostępne czcionki)
- ftransp3 – flaga przezroczystości tła tekstu
- fblink3 – wartość 1 spowoduje miganie tekstu

Wartość zwracana – wysłane msg\_id. Należy zapamiętać i sprawdzać odczyt danych z tego laptimera. Jeśli po 1.5 sekundy w danych odebranych rcv\_msg\_id nie będzie równe ostatniemu wysłanemu id można uznać że pakiet nie dotarł i należy go ponownie wysłać

UWAGA! Kolory należy podawać w formacie liczb 4 bajtowych gdzie 3 młodsze bajty mają znaczenie 0x00BBGRR (BB- składowa niebieska; GG- składowa zielona; RR – składowa czerwona)

### Dostępne czcionki

- 0 -F5x7 – Litery
- 1 – rezerwa
- 2 – rezerwa
- 3 - rezerwa
- 4 -F16x27 - Litery
- 5 – rezerwa
- 6 – rezerwa
- 7 - rezerwa
- 8 -F23x31 – Tylko cyfry
- 9 -F37x52 - Tylko cyfry
- 10 -F47x66 - Tylko cyfry

### readData

function readData(udev:pointer;var t:array of byte): integer; stdcall;

Funkcja odczytująca dane z LAPO-RTU. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- t – bufor na zwracaną strukturę

Zwracana wartość 0- brak odebranych danych ; 1- odczytane zostały dane

Zwracane są dane zgodne ze strukturą:

Pierwsze 28 bajtów bajty sterujące transmisją (nieistotne dla oprogramowania na PC)

Dalsze 100 bajtów to struktura:

radio\_telemetria\_t=packed record

sn:longword; // numer seryjny LAPO. Wartość ta określa również numer węzła niezbędny do wysłania danych do urządzenia. **Adres węzła** obliczamy jako sn and 0xFFFF (2 najmłodsze bajty sn)

rcv\_msg\_id:byte; // ID ostatnio odebranego pakietu z PC. Służy do weryfikacji który pakiet

wysłany do LAPO został ostatnio odebrany i umożliwia kontrolę poprawności odebrania danych przez urządzenie

```
seq_nr:byte; // inkrementowany przez LAPO-D licznik transmisji
frame_type:byte; //zawsze 0
race_id:byte; // id wycigu - niewykorzystywane
res: array [0..3] of longword; //- rezerwa
race_active:byte; // niewykorzystywane
lapNo:byte; // niewykorzystywane
sectorNo:byte; // niewykorzystywane
lines_number :byte; // niewykorzystywane
sector_number_last:byte; // niewykorzystywane
actual_lap_no:byte; // niewykorzystywane
actual_sector_no:byte; // niewykorzystywane
reserved:byte; // niewykorzystywane
lap_time:longword; // niewykorzystywane
sector_time:longword; // niewykorzystywane
RPM_max_lap:word; // niewykorzystywane
RPM_min_lap:word; // niewykorzystywane
RPM_max_sector:word; // niewykorzystywane
RPM_min_sector:word; // niewykorzystywane
actual_laptime:longword; // niewykorzystywane
actual_sessiontime:longword; // niewykorzystywane
actual_RPM:word; // pomiar aktualnej prędkości obrotowej
vbat:word; // poziom naładowania akumulatora
best_laptime:longword; // niewykorzystywane
res32:array [0..8] of longword; // niewykorzystywane
end;
```

### **sendData**

function sendData(udev:pointer;t:array of byte): integer; stdcall;

Funkcja ta umożliwia bezpośrednio wysłanie danych. Wynik funkcji zawsze 1. Parametry wejściowe:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- t – tablica z danymi do wysłania. Dane powinny być zgodn ze strukturą:

wiadomosc\_t=packed record

```
dest_adr :word;
status:byte;
msg_id:byte;
background_no :word;
back_flags :word;
```

```

back_time :word;
session_time :word;
diff_time :word;
position :byte;
rez1:byte;
my_best_lap_time:longword;
res1,res2:longword;
text_1:text_conf_t ;
text_2:text_conf_t ;
text_3:text_conf_t ;
end;
text_conf_t= packed record
  x:word;
  y:word;
  text_flags :word;
  s:array [0..13] of char;
  text_color :word;
  back_color :word;
end;

```

### **closeDevice**

```
function closeDevice(udev:pointer): integer; stdcall;
```

Zamknięcie połączenia z LAPO-RTU

### **readDataWithCallback**

```
function readDataWithCallback(udev:pointer; t:PByteArray
    ;callback_func:TcallFunc;callback_funcLAPO:TcallFuncLAPO): integer; stdcall;
gdzie
TcallFunc = function(g:integer;dest:integer;seq:integer) : integer;stdcall;
TcallFuncLAPO = function(g:integer;received_id:integer) : integer;stdcall;
```

Funkcja ta umożliwia odczyt danych (to samo co readData).

Parametry:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- t – bufor na zwracaną strukturę radio\_telemetria\_t
- callback\_func – funkcja ta zostanie wywołana jeśli odebrany zostanie pakiet z pilota LAPO. Jako pierwszy parametr wywołanej funkcji zostanie przekazany numer rozkazu z pilota a jako drugi numer pojazdu którego rozkaz dotyczy (0 oznacza że dotyczy wszystkich np. dla funkcji ograniczenia prędkości) trzeci parametr to seqnr. UWAGA! Koniecznie należy w obsłudze callbacka wywołać funkcję sendDataToRemoteControl i podać w niej odebrany seqnr. Jeśli chcemy aby operator na pilocie dostał na ekranie odpowiedź na wysłaną komendę wystarczy odpowiednio wypełnić parametry funkcji sendDataToRemoteControl
- callback\_funcLAPO – funkcja ta zostanie wywołana jeśli odebrany zostanie pakiet z dowolnego LAPO. Jako parametry zostanie przekazane – pierwszy parametr numer seryjny lapo a drugi parametr rcv\_msg\_id. Dzięki temu można wykorzystać to do weryfikacji czy poprzednio wysłany pakiet został poprawnie odebrany (rcv\_msg\_id musi być równy zwróconemu id\_msg przez dowolną funkcję biblioteczną wysyłającą dane)

### **sendDataToRemoteControl**

```
function
```

```
sendDataToRemoteControl(udev:pointer;t1:PByteArray;x1,y1:integer;t2:PByteArray;x2,y2:integer;
t3:PByteArray;x3,y3:integer;t4:PByteArray;x4,y4:integer;seqnr:byte):integer;stdcall;
```

Funkcja umożliwia wysłanie komunikatów tekstowych do pilota w odpowiedzi na zapytanie 55 (patrz `readDataWithCallback`).

Parametry:

- `udev` – wskaźnik na uchwyt urządzenia zwracany przez `InitDevice`
- `t1` – pierwsza linijka tekstu maksymalnie 16 znaków
- `x1,y1` – pozycja ekranowa tekstu
- `t2` – druga linijka tekstu maksymalnie 16 znaków
- `x2,y2` – pozycja ekranowa tekstu
- `t3` – trzecia linijka tekstu maksymalnie 16 znaków
- `x3,y3` – pozycja ekranowa tekstu
- `t4` – czwarta linijka tekstu maksymalnie 16 znaków
- `x4,y4` – pozycja ekranowa tekstu
- `seqnr` – numer sekwencji odebrany za pomocą `radDataWithCallback`

Poprzednie funkcje korzystały z predefiniowanych czcionek oraz ekranów zaprojektowanych jako domyślnie. W przypadku kiedy zachodzi konieczność użycia LAPO-D w inny sposób (własny design) jest możliwość korzystania z czcionek ttf (wgrzywanie czcionek umożliwia program `LAPOLibraryBuilder`). Umożliwia to używanie np. czcionek azjatyckich, arabskich, cyrylicy itd. Poniżej funkcje które pozwalają korzystać z wgranych czcionek:

## **sendMessage2**

function

```
sendMessage2(udev:pointer;faddr:integer;fnrback:integer;ftimeBack:integer;FtextOnly:integer;
ft1:PByteArray ;fcolor1:integer;fx1,fy1:word;ffont1,fsize1,fblink1:byte;
ft2:PByteArray ;fcolor2:integer;fx2,fy2:word;ffont2,fsize2,fblink2:byte
): integer; stdcall;
```

Funkcja ta umożliwia wyświetlenie 2 tekstów każdy maksymalnie po 16 znaków. Istnieje możliwość połączenia tekstów – w tym celu należy podać dla drugiego tekstu takie same współrzędne ekranowe wtedy oba napisy zostaną sklejone i wyświetlone jako jeden tekst.

Parametry:

- `udev` – wskaźnik na uchwyt urządzenia zwracany przez `InitDevice`
- `faddr` – adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres `0xFFFF` powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- `fnrback` – numer obrazka z biblioteki graficznej (UWAGA! 0 wyświetla ekran główny z obrotociomierzem)
- `ftimeBack` – czas w sekundach przez jaki ma być ekran wyświetlany. Po tym czasie LAPO-D powróci do ekranu głównego. Wartość 0 spowoduje wyświetlenie ekranu na stałe do czasu wysłania innego rozkazu
- `FtextOnly` – dla 0 wyświetlony zostaje zarówno ekran jak i teksty. Dla 1 ekran pozostaje taki jak był a jedynie dopisywane są teksty. Flaga ta może być używana do dodawania kolejnych tekstów na ekran
- `Ft1` – pierwszy tekst maksymalnie 16 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- `fcolor1` – kolor tekstu

- fx1,fy1 – współrzędne ekranowe początku tekstu
- ffont1 – typ czcionki (numer czcionki z biblioteki ttf z programu LAPOLibraryBuilder)
- fsize1 – rozmiar czcionki w pikselach
- Ft2 – drugi tekst maksymalnie 16 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- Fcolor2- kolor tekstu
- Fx2,fy2 – współrzędne ekranowe początku tekstu
- Ffont2 – typ czcionki (numer czcionki z biblioteki ttf z programu LAPOLibraryBuilder)
- Fsize2 – rozmiar czcionki w pikselach

### **sendMessage4**

function

```
sendMessage4(udev:pointer;faddr:integer;fnrback:integer;ftimeBack:integer;FtextOnly:integer;
ft1:PByteArray ;fcolor1:integer;fx1,fy1:word;ffont1,fsize1:byte;
ft2:PByteArray ;fx2,fy2:word;
ft3:PByteArray ;fcolor3:integer;fx3,fy3:word;ffont3,fsize3:byte;
ft4:PByteArray ;fx4,fy4:word
): integer; stdcall;
```

Funkcja ta umożliwia wyświetlenie do 4 tekstów każdy maksymalnie po 8 znaków przy czym tekst drugi ma zawsze ten sam krój, kolor i rozmiar czcionki jak pierwszy a tekst czwarty jak trzeci. Tutaj także można łączyć teksty poprzez podanie tych samych współrzędnych ekranowych

Parametry:

- udev – wskaźnik na uchwyt urządzenia zwracany przez InitDevice
- faddr – adres laptimera – 2 młodsze bajty numeru seryjnego laptimera. UWAGA. Wysłanie pod adres 0xFFFF powoduje odebranie pakietu przez wszystkie laptimery (rozgłoszenie)
- fnrback – numer obrazka z biblioteki graficznej (UWAGA! 0 wyświetla ekran główny z obrotociomierzem)
- ftimeBack- czas w sekundach przez jaki ma być ekran wyświetlany. Po tym czasie LAPO-D powróci do ekranu głównego. Wartość 0 spowoduje wyświetlenie ekranu na stałe do czasu wysłania innego rozkazu
- FtextOnly – dla 0 wyświetlony zostaje zarówno ekran jak i teksty. Dla 1 ekran pozostaje taki jak był a jedynie dopisywane są teksty. Flaga ta może być używana do dodawania kolejnych tekstów na ekran
- Ft1 – pierwszy tekst maksymalnie 8 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- fcolor1- kolor tekstu
- fx1,fy1 – współrzędne ekranowe początku tekstu
- ffont1 – typ czcionki (numer czcionki z biblioteki ttf z programu LAPOLibraryBuilder)
- fsize1 – rozmiar czcionki w pikselach
- Ft2 – drugi tekst maksymalnie 8 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- Fx2,fy2 – współrzędne ekranowe początku tekstu
- Ft3 – trzeci tekst maksymalnie 8 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- Fcolor3- kolor tekstu
- Fx3,fy3 – współrzędne ekranowe początku tekstu

- Ffont3 – typ czcionki (numer czcionki z biblioteki ttf z programu LAPOLibraryBuilder)
- Fsize3 – rozmiar czcionki w pikselach
- Ft4 – czwarty tekst maksymalnie 8 znaków (UWAGA w formacie UNICODE co oznacza że na każdy znak przypadają 2 bajty)
- Fx4,fy4 – współrzędne ekranowe początku tekstu

UWAGA ! Dla funkcji sendMessage2 i sendMessage4 podanie jako współrzędnej x wartości 0xFFFF oznacza wycentrowanie tekstu na wyświetlaczu