Library lapoDlib.dll
Requires the driver installed to LAPO-RTU and library libusb0.dll

available functions
– InitDevice
– getTextWidth
– sendPassing
– sendCurrentPos
– sendFlag
– sendPicture
– setRPMLimit
– sendMessage
– ReadData
– sendData
– closeDevice
– readDataWithCallback
– sendDataToRemoteControl
– sendMessage2
– sendMessage4
– GetLibVersion

InitDevice
InitDevice function: pointer; stdcall;
This function initializes the connection to the LAPO-RTU. Pointer is returned to the unit holder is necessary to call other functions. If the result of the function return value is nil means initialization error

getTextWidth
getTextWidth function (ffont: integer; t: array of byte): integer; stdcall;
The function returns the text width in pixels for the specified font. It can be used to identify the place where you should write another text when submitting the subtitle of several texts (eg. String longer than 14 characters can not be transmitted as a single string, and must be submitted)
Input parameters:
– ffont - font (see font available)
– t - the text to display. A value of 0 to 1 byte is no text (Maximum 14 characters)
Return value: the width of the text in pixels

sendPassing
functionsendPAssing(udev:pointer;faddr:integer;fpos:integer;ftime:integer;diff1st:integer;better:integer;mybesttime:integer): integer; stdcall;
This function should be called after driving the vehicle across the finish line. input Parameters
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes of serial younger laptimera NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– fpos - the position of the vehicle in the race
– ftime - lap time, which has just been completed. The time should be specified as * 100 or 42.67 times should be sent as 4267
– diff1st - ftime time difference relative to the best time in a session. A negative value indicates that the current time is the new best time of the session

– better - ftime flag specifying whether a new best time of the driver or not. A value of 1 means making your own best time
– mybesttime - the best time of the driver in the session
The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

sendCurrentPos
function
sendCurrentPos(udev:pointer;faddr:integer;fpos:integer;ftime:integer;diff1st:integer;mybesttime:integer): integer; stdcall;
The data refreshes the driver's position in the race. It should be called after crossing the finish line for the other vehicles on the track in order to update their position. input Parameters
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– fpos - the position of the vehicle in the race
– ftime - time to be displayed on the main screen. This may be the best time of the driver in the session or the time of the last lap
– diff1st - ftime time difference relative to the best time in a session. A negative value indicates that the current time is the new best time of the session
– mybesttime - the best time of the driver in the session
The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

sendFlag
sendFlag function (udev: pointer; faddr: integer; f: integer): integer; stdcall;
Send the flag. Input parameters:
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– f - number of flags. 1- chessboard; 2 yellow; 3 blue; 4 red; 5 - green; 6 white; 7 black
The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

sendPicture
sendPicture function (udev: pointer; faddr: integer; no: integer; blink: integer): integer; stdcall;
The display command sending a picture from the video library. Input parameters:
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– no - the number of graphics in the library
– blink - where 1 means that the screen is flickering
The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

setRPMLimit

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

This function sends a limitation of rotation. Input parameters:

– udev - pointer to the device handle returned by InitDevice

– addr - address laptimera - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)

– frpmlimit limit value in revolutions per minute. A value of 0 removes the limitation.

The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

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;

This function allows you to view the image and 3 texts. Input parameters:

– udev - pointer to the device handle returned by InitDevice

– addr - address laptimera - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)

– fnrback - a number of images from the graphics library (NOTE! 0 displays the main screen obrotościomierzem)

– ftimeBack- time in seconds the screen will be displayed. After this time, LAPO-D will return to the main screen. A value of 0 will display permanently to time to send another command

– FtextOnly - 0 is displayed for both the screen and the lyrics. For 1 screen remains the same as it was only credited texts. This flag can be used to add more text on the screen

– FT1 - the text to display. A value of 0 to 1 byte is no text (Maximum 14 characters)

– fcolor1- text color

– fbcolor1 - background string (only relevant if the flag of transparency will not be selected)

– fx1 - the x coordinate of the text

– FY1 - the y coordinate of the text

– ffont1 - font (see font available)

– ftransp1 - flag was thoroughly transparent background

– fblink1 - a value of 1 will flash text

– ft2 - the text to display. A value of 0 to 1 byte is no text (Maximum 14 characters)

– fcolor2- text color

– fbcolor2 - background string (only relevant if the flag of transparency will not be selected)

– fx2 - the x coordinate of the text

– FY2 - the y coordinate of the text

– ffont2 - font (see font available)

– ftransp2 - flag was thoroughly transparent background

– fblink2 - a value of 1 will flash text

– ft3 - the text to display. A value of 0 to 1 byte is no text (Maximum 14 characters)

– fcolor3- text color

– fbcolor3 - background string (only relevant if the flag of transparency will not be selected)

– fx3 - the x coordinate of the text
– fy3 - the y coordinate of the text
– ffont3 - font (see font available)
– ftransp3 - flag was thoroughly transparent background
– fblink3 - a value of 1 will flash text
The return value - sent msg_id. To remember and check reading data from the laptimera. If, after 1.5 seconds of data received rcv_msg_id will not equal the latter can be consigned id that the package has not arrived, and send it again

NOTE! Colors should be given in numbers 4 byte where the 3 younger bytes are important 0x00BBGGRR (BB- blue component; gg- component of green; RR - red component)

available fonts
0 -F5x7 - Letters
1 - reserve
2 - provision
3 - provision
4 -F16x27 - Letters
5 - provision
6 - reserve
7 - reserve
8 -F23x31 - Only digits
9 -F37x52 - Only digits
10 -F47x66 - Only digits

ReadData
ReadData function (udev: pointer; var t: array of byte): integer; stdcall;
Capability LAPO the read data from the RTU. Input parameters:
– udev - pointer to the device handle returned by InitDevice
– t - return buffer structure
The return value of 0- lack of received data; 1 were read data
Returned data are consistent with the structure:
The first 28 bytes of transmission control bytes (not relevant for orpogramowania PC)
Further 100 bytes of the structure:
radio_telemetria_t = packed record
sn: longword; // LAPO serial number. This value also determines the number of the node, necessary to send data to the device. The node address is calculated as sn and 0xFFFF (2 bytes youngest sn)
rcv_msg_id: byte; // ID of the last received packet from a PC. Is used to verify that the package is sent to LAPO has recently received and allows you to control the accuracy of the data is received by the device
seq_nr: byte; // Increment the counter LAPO-D transmission
frame_type: byte; // always 0
race_id: byte; // Id race - not used
res: array [0..3] of longword; // - Reserve
race_active: byte; // unused
lapNo: byte; // unused
SectorNo: byte; // unused
lines_number: byte; // unused

```
  sector_number_last: byte; // unused
  actual_lap_no: byte; // unused
  actual_sector_no: byte; // unused
  reserved: byte; // unused
  lap_time: longword; // unused
  sector_time: longword; // unused
  RPM_max_lap: word; // unused
  RPM_min_lap: word; // unused
  RPM_max_sector: word; // unused
  RPM_min_sector: word; // unused
  actual_laptime: longword; // unused
  actual_sessiontime: longword; // unused
  actual_RPM: word; // Measure the actual speed
  VBAT: word; // The battery charge
  best_laptime: longword; // unused
  res32: array [0..8] of longword; // unused
end;
```

sendData

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

This feature allows you to send the data directly. Result of the function is always 1. Input Parameters:

– udev - pointer to the device handle returned by InitDevice

– t - an array of data to be sent. The data should be consistent with the structure:

```
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
closeDevice function (udev: pointer): integer; stdcall;
Closing connection RTU LAPO

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

This feature allows you to read data (same as ReadData).
parameters:
– udev - pointer to the device handle returned by InitDevice
– t - the buffer to return the structure radio_telemetria_t
– callback_func - this function will be called if a packet from the remote LAPO. As the first parameter will be passed to the called function remote control order number and the second number of the vehicle which the order relates (0 means that all such concerns. Speed limit function) thitd parameter is seqnr. You MUST call sendDataToRemoteControl to confirm that message was received and give seqnr to this function. You can send messages to remote operator (eg. Lights are on etc.)

– callback_funcLAPO - functions The will be called if a packet from any LAPO. Will be passed as parameters - the serial number of the first parameter and the second parameter lapo rcv_msg_id. This allows you to use this to verify that the previously transmitted packet was successfully received (rcv_msg_id must be equal to the value returned by any function id_msg sending the data library)

sendDataToRemoteControl
functionsendDataToRemoteControl(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;

This function allows you to send text messages to the remote control in response to a query 55 (see readDataWithCallback).
parameters:
– udev - pointer to the device handle returned by InitDevice
– t1 - the first line of text up to 16 characters
– x1, y1 - the position of the text on-screen
– t2 - drugaa line of text up to 16 characters
– x2, y2 - Onscreen text entry
– t3 - the third line of text up to 16 characters
– x3, y3 - Onscreen text entry
– t4 - the fourth line of text up to 16 characters
– x4, y4 - Onscreen text entry
– seqnr - sequence number received via radDataWithCallback

Previous functions benefit from pre-defined fonts and screens designed as default. In the event that it is necessary to use LAPO-D in a different way (own design) is the ability to use fonts ttf (upload fonts allows LAPOLibraryBuilder program). This allows the use of such. Asian fonts, Arabic, Cyrillic etc.

The following functions allow you to use the uploaded fonts:

sendMessage2

```
sendMessage2 function (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;
```

This feature allows you to view two texts, each a maximum of 16 characters. You can combine texts - for this purpose should be given for the second text the same screen coordinates when both subtitles are glued together and displayed as a single text.
parameters:
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– fnrback - a number of images from the graphics library (NOTE! 0 displays the main screen obrotościomierzem)
– ftimeBack- time in seconds the screen will be displayed. After this time, LAPO-D will return to the main screen. A value of 0 will display permanently to time to send another command
– FtextOnly - 0 is displayed for both the screen and the lyrics. For 1 screen remains the same as it was only credited texts. This flag can be used to add more text on the screen
– FT1 - the first text up to 16 characters (NOTE UNICODE format which means that each character are 2 bytes)
– fcolor1- text color
– fx1, FY1 - screen coordinate of the start of the text
– ffont1 - font type (font number from the library program LAPOLibraryBuilder ttf)
– fsize1 - font size in pixels
– Ft2 - the second text up to 16 characters (NOTE UNICODE format which means that each character are 2 bytes)
– Fcolor2- text color
– Fx2, FY2 - screen coordinate of the start of the text
– Ffont2 - font type (font number from the library program LAPOLibraryBuilder ttf)
– Fsize2 - font size in pixels

sendMessage4

```
sendMessage4 function (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;
```

This feature allows up to 4 texts, each a maximum of 8 characters in the text of the second is always the same font, font color and size as the first and the fourth as the third text. Here, too, you can combine texts by specifying the same screen coordinates

parameters:
– udev - pointer to the device handle returned by InitDevice
– faddr - laptimera address - 2 bytes younger laptimera serial number. NOTE. Sending the address 0xFFFF will receive the package by matches any laptimery (broadcast)
– fnrback - a number of images from the graphics library (NOTE! 0 displays the main screen obrotościomierzem)
– ftimeBack- time in seconds the screen will be displayed. After this time, LAPO-D will return to the main screen. A value of 0 will display permanently to time to send another command
– FtextOnly - 0 is displayed for both the screen and the lyrics. For 1 screen remains the same as it was only credited texts. This flag can be used to add more text on the screen
– FT1 - the first text up to 8 characters (NOTE UNICODE format which means that each character are 2 bytes)
– fcolor1- text color
– fx1, FY1 - screen coordinate of the start of the text
– ffont1 - font type (font number from the library program LAPOLibraryBuilder ttf)
– fsize1 - font size in pixels
– Ft2 - the second text up to 8 characters (NOTE UNICODE format which means that each character are 2 bytes)
– Fx2, FY2 - screen coordinate of the start of the text
– Ft3 - the third text up to 8 characters (NOTE UNICODE format which means that each character are 2 bytes)
– Fcolor3- text color
– Fx3, fy3 - screen coordinate of the start of the text
– Ffont3 - font type (font number from the library program LAPOLibraryBuilder ttf)
– Fsize3 - font size in pixels
– Ft4 - the fourth text up to 8 characters (NOTE UNICODE format which means that each character are 2 bytes)
– FX4, FY4 - screen coordinate of the start of the text

NOTE! SendMessage2 function and sendMessage4 administration as the x value of 0xFFFF means centering the text on the display