

Phoenix mode on CardMaster USB

Important !!!!

All calls to the DLL is performed with *cdecl* calling convention.

Also keep in mind, that data from the device is being readen by a timer, this means your main thread must give time to DLL at regular basic eg. by calling the windows wait(time) function.

Functions

```
function Connect: Cardinal; cdecl;  
procedure Disconnect; cdecl;  
procedure SetProtocol(p: Integer);  
procedure greenOn; cdecl;  
procedure greenOff; cdecl;  
procedure redOn; cdecl;  
procedure redOff; cdecl;  
procedure setBaudRate(br: Integer); cdecl;  
procedure transmit(data: pchar; len: integer); cdecl;  
function receive(buff: pchar; len: integer): Integer; cdecl;  
procedure reset; cdecl;
```

Startup

Before the communication can begin a connection need to made and the baudrate and protocol needs to be set, this is done by calling connect followed by setBaudRate and setProtocol.

Connecting

```
function Connect: Cardinal; cdecl;
```

Parameters:
 none

Returns:
 1 on success.
 0 on failure.
 - Note this value should be ignored for now, due to a bug.

Description:
 Establish a connection to CardMasterUSB device.

Disconnecting

```
procedure Disconnect; cdecl;
```

Parameters:
 none

Returns:
 nothing

Description:
 Disconnects from CardMasterUSB device.

Setting baudrate

```
procedure setBaudRate(br: Integer); cdecl;
```

Parameters:

br: A constant indicating which baudrate to use, the possibilities are :

- 9600bps => br=0;
- 9895bps => br=1;

Returns;

nothing

Description:

This function sets the current baudrate for communication.

Setting Protocol

```
procedure SetProtocol(p: Integer);
```

Parameters:

p: A constant indicating which protocol type to use, the possibilities are :

- T=0 => p=0 (Direct conversion)
- T=1 => p=1 (Indirect, note bug in HW means parity bit is wrong when sent to card)

Returns:

nothing

Description:

This function sets the current protocol for communication.

Resetting

```
procedure reset; cdecl;
```

Parameters:

none

Returns:

nothing

Description:

Triggers a reset of the smartcard currently in reader, this should be done after settings protocol and baudrate, aswell as when a card is inserted.

Controlling LED's

```
procedure greenOn; cdecl;  
procedure greenOff; cdecl;  
procedure redOn; cdecl;  
procedure redOff; cdecl;
```

Parameters:
 none

Returns:
 nothing

Description:
 Either turns on or off a LED.
 If both red and green are on the light will be yellow/orange.

Transmitting to card

```
procedure transmit(data: pchar; len: integer); cdecl;
```

Parameters:

- data: pointer to the data that should be sent, this data
 should be in hex format eg. "12AB4312".
- Len: Length of the data, should be number of hex bytes,
 eg. if you send "1ACC" to card the length is only
 2 since it is only two bytes.

Returns:
 nothing

Description:
 Transmit data to card.

Receiving from card

```
function receive(buff: pchar; len: integer): Integer; cdecl;
```

Parameters:

- data: pointer to a buffer which should be big enough to
 hold the received data.
- Len: Max length of the data we want to receive, if the
 len is bigger than the available number of bytes,
 the available bytes will be received and the length
 will be returned.

Returns:
 Number of bytes actually read.

Description:
 Receives data from card.

Functions coming in next version

```
procedure SetXtal(x: Integer);  
procedure SetParity(p: Integer);
```