Introdução

A biblioteca em C pode ser encontrada no Kit de desenvolvimento do iDBIO. Os headers com as definições das funções estão disponíveis na pasta "include". Esta biblioteca apresenta 3 versões: Linux x86, Linux x64 e Windows.

Para utilizar a biblioteca em seu projeto basta "linkar" a mesma ao seu executável.

Exemplo com gcc:

```
gcc example.c -lcidbio -o example
```

Códigos de Retorno

int32_t *CIDBIO_GetErrorMessage* (int32_t error, char msg)**

Retorna mensagem em inglês do erro passado.

Parâmetros de entrada:*

- error: *int32_t*
  Código de erro

Parâmetros de saída:

- msg: *char***
  Descrição em inglês do erro.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

Todas as funções desta biblioteca retornam um número inteiro. Este código de retorno pode indicar uma das 3 situações abaixo:

- Sucesso (== 0) - Operação foi realizada com sucesso
- Aviso (> 0) - Operação foi realizada mas com alguma ressalva
- Erro (< 0) - Operação falhou

Valor	Define	Descrição
0	CIDBIO_SUCCESS	Operação realizada com sucesso
1	CIDBIO_WARNING_ALREADY_INIT	Biblioteca já inicializada
2	CIDBIO_WARNING_NO_IDS_ON_DEVICE	Nenhum Template cadastrado
3	CIDBIO_WARNING_OVERWRITING_TEMPLATE	Template foi sobrescrito
-1	CIDBIO_ERROR_UNKNOWN	Erro desconhecido
-2	CIDBIO_ERROR_NO_DEVICE	Dispositivo não encontrado
-3	CIDBIO_ERROR_NULL_ARGUMENT	Argumento nulo
-4	CIDBIO_ERROR_INVALID_ARGUMENT	Argumento inválido
-5	CIDBIO_ERROR_CAPTURE	Erro durante a captura
-6	CIDBIO_ERROR_CAPTURE_TIMEOUT	Tempo de captura expirado
-7	CIDBIO_ERROR_COMM_USB	Erro de comunicação USB
-8	CIDBIO_ERROR_IO_ON_HOST	Erro de comunicação do Host
-9	CIDBIO_ERROR_TEMPLATE_ALREADY_ENROLLED	Template já cadastrado

Valor	Define	Descrição
-10	CIDBIO_ERROR_MERGING	Falha no Merge
-11	CIDBIO_ERROR_MATCHING	Falha no Match
-12	CIDBIO_ERROR_INVALID_FW_FILE	Arquivo de Firmware inválido
-13	CIDBIO_ERROR_NO_SPACE_LEFT_ON_DEVICE	Espaço no dispositivo esgotado
-14	CIDBIO_ERROR_NO_TEMPLATE_WITH_ID	Template não cadastrado
-15	CIDBIO_ERROR_INVALID_ERRNO	Código de erro inválido
-16	CIDBIO_ERROR_UNAVAILABLE_FEATURE	Funcionalidade não disponível
-17	CIDBIO_ERROR_PREVIOUS_FW_VERSION	Versão do firmware é anterior à atual
-18	CIDBIO_ERROR_NOT_IDENTIFIED	Template não identificado
-19	CIDBIO_ERROR_BUSY	Dispositivo esta ocupado
-20	CIDBIO_ERROR_CAPTURE_CANCELED	Captura foi cancelada
-21	CIDBIO_ERROR_NO_FINGER_DETECTED	Digital não foi detectada

Inicialização e Finalização

int32_t *CIDBIO_SetSerialCommPort* (char* comPort)

Configura em qual porta o dispositivo deve se conectar.
Disponível a partir da versão da SDK: *1.4.0*

Parâmetro de entrada:

- comPort: *char**
  Porta serial ao qual o dispositivo deve se conectar. Deve ser "COMx" em Windows e "/dev/ttyACMx" em Linux

int32_t *CIDBIO_Init* (void)

Checa e inicializa o dispositivo.

int32_t *CIDBIO_Terminate* (void)

Finaliza o dispositivo.

Captura de dedo

int32_t *CIDBIO_CaptureImage* (byte imageBuf, uint32_t* width, uint32_t* height)**

Captura uma imagem de um dedo. Quando esta função é chamada o dispositivo entra no modo de captura de imagens. Esta função não retorna enquanto um dedo não for capturado, ou o tempo de captura for expirado (30 segundos).

Parâmetros de saída:

- imageBuf: *byte***
  Buffer com o conteúdo bitmap da imagem.
  Usar **CIDBIO_FreeByteArray()** para liberar memória deste parâmetro.
- width: *uint32_t**
  Largura da imagem
- height: *uint32_t**
  Altura da imagem

int32_t *CIDBIO_CheckFingerprint* (byte imageBuf, uint32_t* width, uint32_t* height)**

Checa se existe um dedo pressionando o leitor. Em caso positivo a imagem da digital é retornada, em caso negativo o erro **CIDBIO_ERROR_NO_FINGER_DETECTED** é retornado. Esta função sempre retorna imediatamente.

Disponível a partir da versão do firmware: *1.2.0*

Parâmetros de saída:

- imageBuf: *byte***
  Buffer com o conteúdo bitmap da imagem.
  Usar **CIDBIO_FreeByteArray()** para liberar memória deste parâmetro.
- width: *uint32_t**
  Largura da imagem
- height: *uint32_t**
  Altura da imagem

int32_t *CIDBIO_CaptureImageAndTemplate* (char temp, byte** imageBuf, uint32_t* width, uint32_t* height, int32_t* quality)**

Captura uma imagem de um dedo e extrai o seu template. Quando esta função é chamada o dispostivo entra no modo de captura de imagens. Esta função não retorna enquanto um dedo não for capturado, ou o tempo de captura for expirado (30 segundos).

Parâmetros de saída:

- temp: *char***
  Base64 do template extraído.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.
- imageBuf: *byte***
  Buffer com o conteúdo bitmap da imagem.
  Usar **CIDBIO_FreeByteArray()** para liberar memória deste parâmetro.
- width: *uint32_t**
  Largura da imagem
- height: *uint32_t**
  Altura da imagem.

- quality: *int32_t**
  Valor de 0 a 100 que define a qualidade da imagem do dedo capturado.

int32_t *CIDBIO_CaptureAndEnroll* (int64_t id)

Captura 3 imagens de um dedo e o cadastra no dispositivo. Quando esta função é chamada o dispositivo entra no modo de captura de imagens. Esta função não retorna enquanto um dedo não for capturado 3 vezes, ou o tempo de captura for expirado (30 segundos por iteração).

Parâmetro de entrada:

- id: *int64_t*
  Número identificador do template

int32_t *CIDBIO_CaptureAndIdentify* (int64_t* id, int32_t* score, int32_t* quality)

Captura uma imagem de um dedo e tenta encontrar uma biometria semelhante cadastrada no dispositivo. Quando esta função é chamada o dispositivo entra no modo de captura de imagens. Esta função não retorna enquanto um dedo não for capturado, ou o tempo de captura for expirado (30 segundos).

Parâmetros de saída:

- id: *int64_t**
  Número identificador do template encontrado.
- score: *int32_t**
  Valor entre 0 e 20000 que define a semelhança entre o dedo capturado e o cadastrado.
- quality: *int32_t**
  Valor de 0 a 100 que define a qualidade da imagem do dedo capturado.

int32_t *CIDBIO_CaptureAndMatch* (int64_t id, int32_t* score, int32_t* quality)

Captura uma imagem de um dedo e verifica a semelhança com a biometria cadastrada com identificador *id*. Quando esta função é chamada o dispositivo entra no modo de captura de imagens. Esta função não retorna enquanto um dedo não for capturado, ou o tempo de captura for expirado (30 segundos).

Parâmetros de entrada:

- id: *int64_t*
  Número identificador do template a ser comparado.

Parâmetros de saída:

- score: *int32_t**
  Valor entre 0 e 20000 que define a semelhança entre o dedo capturado e o cadastrado.
- quality: *int32_t**
  Valor de 0 a 100 que define a qualidade da imagem do dedo capturado.

int32_t *CIDBIO_CancelCapture* ()

Cancela a operação de captura atual.
Disponível a partir da versão do firmware: *1.2.0*

Gerenciar Templates

int32_t *CIDBIO_ExtractTemplateFromImage* (uint32_t width, uint32_t height, byte* imageBuf, char temp, int32_t* quality)**

Extrai o template de uma imagem.

Parâmetros de entrada:

- width: *uint32_t*
  Largura da imagem
- height: *uint32_t*
  Altura da imagem.
- imageBuf: *byte**
  Buffer com o conteúdo bitmap da imagem.

Parâmetros de saída:

- temp: *char***
  Base64 do template extraído.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.
- quality: *int32_t**
  Valor de 0 a 100 que define a qualidade da imagem do dedo capturado.

int32_t *CIDBIO_MergeTemplates* (char* temp1, char* temp2, char* temp3, char tempFinal)**

Mescla 3 templates em um único template. Templates devem pertencer ao mesmo dedo. (Formatos "ISO" e "ANSI" não são suportados para importação de templates)

Parâmetros de entrada:

- temp1: *char**
  Base64 do primeiro template.
- temp2: *char**
  Base64 do segundo template.
- temp3: *char**
  Base64 do terceiro template.

Parâmetros de saída:

- tempFinal: *char***
  Base64 do template mesclado.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

int32_t *CIDBIO_MatchTemplates* (char* temp1, char* temp2, int32_t* score)

Verifica se 2 templates são iguais. (Formatos "ISO" e "ANSI" não são suportados para importação de templates)

Parâmetros de entrada:

- temp1: *char**
  Base64 do primeiro template.
- temp2: *char**
  Base64 do segundo template.

Parâmetros de saída:

- score: *int32_t**
  Valor entre 0 e 20000 que define a semelhança entre os 2 templates.

int32_t *CIDBIO_MatchTemplateByID* (int64_t id, char* temp, int32_t* score)

Verifica se o template *temp* é igual ao template cadastrado com identificador *id*. (Formatos "ISO" e "ANSI" não são suportados para importação de templates)

Parâmetros de entrada:

- id: *int64_t*
  identificador do template a ser verificado.

- temp: *char**
  Base64 do template.

Parâmetros de saída:

- score: *int32_t**
  Valor entre 0 e 20000 que define a semelhança entre os 2 templates.

int32_t *CIDBIO_GetTemplateIDs* (int64_t ids, uint32_t* len)**

Carrega os identificadores de todos templates cadastrados.

Parâmetros de saída:

- ids: *int64_t***
  Identificadores de todos os templates cadastros no dispositivo.
  Usar **CIDBIO_FreeIDArray()** para liberar memória deste parâmetro.
- len: *uint32_t**
  Número de templates cadastros no dispositivo.

int32_t *CIDBIO_GetTemplate* (int64_t id, char temp)**

Carrega o template cadastrado com identificador *id*.

Parâmetros de entrada:

- id: *int64_t*
  Identificador do templates cadastro no dispositivo.

Parâmetros de saída:

- temp: *char***
  Base64 do template.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

int32_t *CIDBIO_SaveTemplate* (int64_t id, char* temp)

Salva o template *temp* com o identificador *id*. (Formatos "ISO" e "ANSI" não são suportados para importação de templates)

Parâmetros de entrada:

- id: *int64_t*
  Identificador do templates cadastro no dispositivo.
- temp: *char**
  Base64 do template.

int32_t *CIDBIO_DeleteTemplate* (int64_t id)

Deleta o template cadastrado com identificador *id*.

Parâmetros de entrada:

- id: *int64_t*
  Identificador do templates cadastro no dispositivo.

int32_t *CIDBIO_DeleteAllTemplates* (void)

Deleta todos os template cadastrados no dispositivo.

Configuração do dispositivo

É possível alterar o comportamento do dispositivo atráves das configurações abaixo:

Valor	Define	Descrição
1	CIDBIO_PARAM_MIN_VAR	Variância mínima para captura de um dedo. (padrão: "1000")
2	CIDBIO_PARAM_SIMILIARITY_THRESHOLD	Valor de 1 a 20000 que determina o valor mínimo para 2 templates serem considerados iguais. Valor 0 para automático (padrão)
4	CIDBIO_PARAM_BUZZER_ON	"1" (padrão) ou "0" para ligar ou desligar o buzzer
5	CIDBIO_PARAM_TEMPLATE_FORMAT	Configura o formato do template. Valores possíveis: "**ICS**": Formato proprietário (padrão) "**ANSI**": Formato ANSI (Somente para exportação de templates) "**ISO**": Formato ISO (Somente para exportação de templates) "**ANSI_PLUS**": Formato ANSI com informações proprietárias "**ISO_PLUS**": Formato ISO com informações proprietárias
6	CIDBIO_PARAM_ROTATION	Rotação máxima aceita para que dois dedos sejam aceitos como iguais. Esta rotação é aplicada no sentido horário e anti-horário. Este parâmetro impacta na velocidade de identificação, quanto maior este valor, mais tempo a identificação demorará. Valor de 0 a 180 (padrão: "30")
7	CIDBIO_PARAM_DETECT_TIMEOUT	Tempo, em milissegundos, de espera para detecção de um dedo. (padrão: "30000")

int32_t *CIDBIO_SetParameter* (int32_t config, char* value)

Configura o parâmetro *config* com o valor *value*.

Parâmetros de entrada:

- config: *int32_t*
  Parâmetro a ser configurado
- value: *char**
  Valor a ser configurado.

int32_t *CIDBIO_GetParameter* (int32_t config, char value)**

Lê o valor do parâmetro *config*.

Parâmetros de entrada:

- config: *int32_t*
  Parâmetro a ser configurado

Parâmetros de saída:

- value: *char***
  Valor atual do parâmetro.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

int32_t *CIDBIO_GetDeviceInfo* (char version, char** serialNumber, char** model)**

Lê as informações do dispositivo.

Parâmetros de saída:

- version: _char**
  Versão do firmware do dispositivo.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

- serialNumber: *char***
  Número de série do dispositivo.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.
- model: *char***
  Modelo do dispositivo.
  Usar **CIDBIO_FreeString()** para liberar memória deste parâmetro.

int32_t *CIDBIO_UpdateFirmware* (char* filePath)

Atualiza o firmware do dispositivo com o arquivo em *filePath*. Este procedimento pode demorar alguns minutos. Por favor, não desconecte o dispositivo durante este período, pois isto pode danificar o dispositivo de forma permanente.
A última versão do fimware pode ser encontrada aqui.

Parâmetros de entrada:

- filePath: *char**
  Caminho completo para o arquivo contendo a atualização de firmware.

Desalocação de memória

int32_t *CIDBIO_FreeByteArray* (byte* array)

Desaloca memória de um array de bytes.

Parâmetros de entrada:

- array: *byte**
  Array de bytes a ser desalocado da memória.

int32_t *CIDBIO_FreeString* (char* str)

Desaloca memória de uma string de c.

Parâmetros de entrada:

- str: *char**
  String a ser desalocado da memória.

int32_t *CIDBIO_FreeIDArray* (int64_t* ids)

Desaloca memória de um array de inteiros de 64 bits.

Parâmetros de entrada:

- ids: *int64_t**
  Array de inteiros de 64 bits a ser desalocado da memória.