class PyFingerprint

methods

setMaxPacketSize

Sets the maximum packet size of sensor.

```
Arguments:

packetSize (int): 32, 64, 128 and 256 are supported.

Raises:

ValueError: if passed packet size is invalid

Exception: if any error occurs
```

getMaxPacketSize

Gets the maximum allowed size of a single packet.

```
Returns:

Return the max size (int).

Raises:

ValueError: if packet size is invalid

Exception: if any error occurs
```

getTemplateIndex

Gets a list of the template positions with usage indicator.

```
Arguments:

page (int): The page (value between 0 and 3).

Returns:

The list.

Raises:

ValueError: if passed page is invalid

Exception: if any error occurs
```

getTemplateCount

Gets the number of stored templates.

```
Returns:

The template count (int).

Raises:

Exception: if any error occurs
```

readImage

Reads the image of a finger and stores it in image buffer.

```
Returns:

True if image was read successfully or False otherwise.

Raises:

Exception: if any error occurs
```

downloadImage

Downloads the image from image buffer.

```
Arguments:

imageDestination (str): Path to image

Raises:

ValueError: if directory is not writable

Exception: if any error occurs
```

convertImage

Converts the image in image buffer to characteristics and stores it in specified char buffer.

```
Arguments:

charBufferNumber (int): The char buffer. Use`FINGERPRINT_CHARBUFFER1` or
`FINGERPRINT_CHARBUFFER2`.

Returns:

True if successful or False otherwise.

Raises:

ValueError: if passed char buffer is invalid
```

```
Exception: if any error occurs
```

storeTemplate

Stores a template from the specified char buffer at the given position.

```
Arguments:

positionNumber (int): The position

charBufferNumber (int): The char buffer. Use`FINGERPRINT_CHARBUFFER1` or
`FINGERPRINT_CHARBUFFER2`.

Returns:

The position number (int) of the stored template.

Raises:

ValueError: if passed position or char buffer is invalid

Exception: if any error occurs
```

searchTemplate

Searches inside the database for the characteristics in char buffer.

```
Arguments:

charBufferNumber (int): The char buffer. Use`FINGERPRINT_CHARBUFFER1` or
`FINGERPRINT_CHARBUFFER2`.

positionStart (int): The position to start the search

count (int): The number of templates

Returns:

A tuple that contain the following information:

0: integer(2 bytes) The position number of found template.

1: integer(2 bytes) The accuracy score of found template.

Raises:
```

```
Exception: if any error occurs
```

loadTemplate

Loads an existing template specified by position number to specified char buffer.

```
Arguments:

positionNumber (int): The position

charBufferNumber (int): The char buffer. Use`FINGERPRINT_CHARBUFFER1` or
`FINGERPRINT_CHARBUFFER2`.

Returns:

True if successful or False otherwise.

Raises:

ValueError: if passed position or char buffer is invalid

Exception: if any error occurs
```

deleteTemplate

Deletes templates from fingerprint database. Per default one.

```
Arguments:

positionNumber (int): The position

count (int): The number of templates to be deleted.

Returns:

True if successful or False otherwise.

Raises:

ValueError: if passed position or count is invalid

Exception: if any error occurs
```

compareCharacteristics

Compare the finger characteristics of char buffer 1 with char buffer 2 and returns the accuracy score.

```
Returns:

The accuracy score (int). 0 means fingers are not the same.

Raises:

Exception: if any error occurs
```

downloadCharacteristics

Downloads the finger characteristics from the specified char buffer.

```
Arguments:

charBufferNumber (int): The char buffer. Use`FINGERPRINT_CHARBUFFER1` or
`FINGERPRINT_CHARBUFFER2`.

characteristicsData (list): The characteristics

Returns:

The characteristics (list).

Raises:

ValueError: if passed char buffer is invalid

Exception: if any error occurs
```