

УТВЕРЖДЕН

ДССЛ.00100-01 31 01 - ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«3i Gender ID SDK»

Описание применения

ДССЛ.00100-01 31 01

Листов 18

Литера О<sub>1</sub>

2016

## АННОТАЦИЯ

Настоящий документ предназначен для ознакомления с программным обеспечением «3i Gender ID SDK» и содержит описание интерфейса программирования (API) для программистов, обеспечивающих использование 3i Gender ID SDK в качестве модуля, встраиваемого в другое программное обеспечение.

В разделе 1 приводятся сведения о назначении и составе 3i Gender ID SDK.

В разделе 2 указаны требования к программно-техническим средствам, необходимым для работы 3i Gender ID SDK.

В разделе 3 указывается описание задач, решаемых 3i Gender ID SDK, даются сведения об используемых технологиях.

В разделе 4 даются сведения об установке 3i Gender ID SDK.

В разделе 5 приводится описание интерфейса программирования 3i Gender ID SDK.

В разделе 6 даются сведения о входных и выходных данных 3i Gender ID SDK.

В разделе 7 приводятся основные сообщения оператору при работе с 3i Gender ID SDK.

## СОДЕРЖАНИЕ

АННОТАЦИЯ.....	3
1. Назначение программы.....	5
2. Условия применения.....	6
3. Описание задачи.....	7
3.1 Технологии работы с речевыми данными.....	7
3.2 Состав 3i Gender ID SDK.....	7
4. Вызов и загрузка программы.....	8
4.1 Установка 3i Gender ID SDK.....	8
5. Выполнение программы.....	9
5.1 Комплект модуля 3i Gender ID SDK.....	9
5.2 Описание функционала API.....	9
6. Входные и выходные данные.....	13
7. Сообщения оператору.....	14
Перечень сокращений.....	16

## 1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Программное обеспечение «3i Gender ID SDK» предназначено для решения задач определения (идентификации) пола по диктору. 3i Gender ID SDK используется в качестве библиотеки модулей, встраиваемых в другое программное обеспечение, предоставляя разработчику соответствующий функционал API.

API (сокр. англ. Application Programming Interface, интерфейс программирования приложений, интерфейс прикладного программирования) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах.

К основным преимуществам приложения, предоставляющего shared object API в качестве инструмента доступа к функциональным возможностям, можно отнести:

- надёжность (за счет отсутствия необходимости взаимодействия с удаленным модулем);
- производительность (за счет использования кэша и подгрузки кода и данных в пространство адресов приложения);
- прозрачность системы взаимодействия;
- легкость внесения изменений;
- масштабируемость.

Продукт реализован в виде динамической библиотеки, написанной на языке C++.

## 2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Для функционирования 3i Gender ID SDK необходима вычислительная система с параметрами не хуже:

- CPU Intel Core i7 – 5820К 3.3 ГГц (6 физических вычислительных ядер);
- ОЗУ 16 ГБ;
- 100 Гб свободного места на жёстком диске.

2.2. Для функционирования 3i Gender ID SDK на вычислительной системе должно быть установлено следующее общее программное обеспечение:

для семейства операционных систем Microsoft Windows:

- операционная система – Microsoft Windows 7 (x64) SP1 или выше;
- распространяемый пакет Microsoft VC 2013 Redist (x64).

для семейства 64-разрядных операционных систем на базе ядра Linux:

- операционная система – представитель семейства 64-разрядных операционных систем на базе ядра Linux (CentOS, Ubuntu, Astra Linux и пр.);
- распространяемый пакет libtbb2 для семейства операционных систем на базе ядра Linux, обеспечивающий управление нагрузкой многоядерных процессоров при исполнении модулей C++ (параллельная обработка данных).

### 3. ОПИСАНИЕ ЗАДАЧИ

#### 3.1 Технологии работы с речевыми данными

Возможности 3i Gender ID SDK определения пола диктора базируются на технологиях традиционного подхода, основанного на Гауссовых смесях (GMM – Gaussian Mixture Model).

Встроенный специальный модуль, выполняющий очистку входного звукового потока от посторонних – неречевых – вставок и выделяющий участки с речевой активностью на основе анализа энергии сигнала, способствует увеличению качества идентификации. Указанные технологии определения пола диктора обеспечивают:

- использование в нескольких параллельных потоках, что увеличивает производительность модуля;
- высокое качество идентификации.

#### 3.2 Состав 3i Gender ID SDK

3i Gender ID SDK включает в себя следующие модули (библиотеки):

- модуль определения пола диктора.

## 4. ВЫЗОВ И ЗАГРУЗКА ПРОГРАММЫ

### 4.1 Установка 3i Gender ID SDK

Установка модулей (библиотек) 3i Gender ID SDK осуществляется переносом (копированием) архива с модулями с загрузочного диска в требуемую директорию на жестком диске вычислительной системы. После переноса (копирования) архива модулей, динамические библиотеки модулей разархивируются из него в требуемую директорию на жестком диске вычислительной системы. Модули (библиотеки) 3i Gender ID SDK готовы к эксплуатации.

## 5. ВЫПОЛНЕНИЕ ПРОГРАММЫ

### 5.1 Комплект модуля 3i Gender ID SDK

Модуль 3i Gender ID SDK имеет классический интерфейс, характерный для динамически подгружаемых библиотек. В комплекте с модулем поставляется:

- бинарный файл динамически связываемой библиотеки (расширение so, dll);
- бинарный файл прокси-библиотеки для статического связывания приложения пользователя с динамически связываемой библиотекой (расширение a);
- файл-хидер, содержащий исходный код интерфейса модуля на C++ для сборки связывания динамической библиотеки модуля и приложения пользователя (расширение h);
- модели обработки данных для каждого модуля.

Описание функционала API 3i Gender ID SDK представлено ниже.

### 5.2 Описание функционала API

Список функций API 3i Gender ID SDK представлен в таблице 5.2.1.

Таблица 5.2.1 Список функций API 3i Gender ID SDK

Наименование функции	Описание функции
<i>init</i>	Инициализация библиотеки
<i>deInit</i>	Деинициализация библиотеки
<i>process_FromFile</i>	Определение пола диктора по голосу (источник: WAV-файл)
<i>process_FromBuf</i>	Определение пола диктора по голосу (источник: буфер памяти)
<i>libIsInit</i>	Получение информации о статусе



Наименование функции	Описание функции
	инициализации библиотеки
<i>getErrMsg</i>	Получение строки с текстом ошибки

Описание функций API 3i Gender ID SDK представлено ниже.

## 1. Инициализация библиотеки

Прежде, чем приступить к работе с библиотекой, необходимо её инициализировать – загрузить модели мужского и женского голосов, а также задать режим работы встроенного детектора стационарных сигналов.

API функции:

```
int init (const char* pathToModelM, const char* pathToModelF, bool  
useInternalSD);
```

*pathToModelM* – путь к модели мужского голоса;

*pathToModelF* – путь к модели женского голоса;

*useInternalSD* – значение «TRUE» для использования детектора стационарных сигналов и фоновых шумов, «FALSE» – не использовать.

Установка флага *useInternalSD=FALSE* подразумевает, что входной аудио поток предварительно очищен от неречевых сигналов, таких как тишина, фоновый шум, гудки и DTMF-сигналы. Процесс вычисления модели голоса диктора может незначительно ускориться относительно времени, затрачиваемого при *useInternalSD=TRUE*. Функция возвращает код согласно таблице возвращаемых кодов.

## 2. Деинициализация библиотеки

По окончании работы с библиотекой необходимо выгрузить, которые были загружены при инициализации библиотеки.

API функции:

```
int deInit ();
```

После вызова данной функции будет заблокирована возможность работы с некоторыми функциями библиотеки, а именно *process\_FromFile()* и

*process\_FromBuf()*. Функция *deInit()* будет «ждать» окончания работы всех запущенных ранее функций из числа перечисленных, а по завершении их работы выполнит выгрузку данных. Любая из этих функций, вызванная в момент «ожидания», завершится кодом IS\_BLOCKED=8. Функция возвращает код согласно таблице возвращаемых кодов.

### 3. Определение пола диктора по голосу (источник: WAV-файл)

API функции:

```
int process_FromFile (float& score_m, float& score_f, const char*  
srcFilePath);
```

*score\_m* – адрес для записи значения оценки близости к «мужской» модели;

*score\_f* – адрес для записи значения оценки близости к «женской» модели;

*srcFilePath* – путь к WAV-файлу.

В случае успеха функция запишет в переменные *score\_m* и *score\_f* значения оценок близости голоса к модели мужского и женского голосов и возвращает код NO\_ERR=0, иначе – код ошибки (см. таблицу возвращаемых кодов).

Если  $score_m \geq score_f$ , то следует полагать, что голос, представленный в файле *srcFilePath*, является мужским, иначе – женским.

Примечание: при обработке звуковых данных большого объёма велика вероятность того, что в потоке присутствует речь более, чем одного диктора. В таком случае, результат идентификации будет менее корректным. Поэтому рекомендуется разбивать входной поток на части длительностью не более 5 минут каждый. Такой подход к использованию функции также обеспечит оптимальный расход памяти, необходимой для построения модели голоса диктора.

### 4. Определение пола диктора по голосу (источник: буфер памяти)

API функции:

```
int process_FromBuf ( float& score_m, float& score_f, const short*  
soundData, int soundDataSize );
```

*score\_m* – адрес для записи значения оценки близости к «мужской» модели;

*score\_f* – адрес для записи значения оценки близости к «женской» модели;

*soundData* – указатель на массив отсчетов;

*soundBufSize* – количество отсчетов в аудио буфере.

В случае успеха функция запишет в переменные *score\_m* и *score\_f* значения оценок близости голоса к модели мужского и женского голосов и возвращает код NO\_ERR=0, иначе – код ошибки (см. таблицу возвращаемых значений).

Если  $score_m \geq score_f$ , то следует полагать, что голос, представленный в файле *srcFilePath*, является мужским, иначе – женским.

Примечание: при обработке звуковых данных большого объема велика вероятность того, что в потоке присутствует речь более, чем одного диктора. В таком случае результат идентификации будет менее корректным. Поэтому рекомендуется разбивать входной поток на части длительностью не более 5 минут каждый. Такой подход к использованию функции также обеспечит оптимальный расход памяти, необходимой для построения модели голоса диктора.

## 5. Получение информации о статусе инициализации библиотеки

API функции:

```
bool libIsInit();
```

Функция возвращает значение «TRUE», если библиотека проинициализирована, иначе – «FALSE».

## 6. Получение строки с текстом ошибки

API функции:

```
const char* getErrMsg(int code);
```

*code* – код согласно таблице возвращаемых кодов.

Функция возвращает указатель на строку с текстовой интерпретацией кода ошибки.

## 6. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Требования к входным аудио данным:

- возможные источники: WAV-файлы, буфер отсчётов;
- частота дискретизации сигнала: 8 кГц;
- разрядность квантования: 8 бит, 16-бит;
- тип кодирования, если источником является WAV-файл: А-закон, Му-закон или РСМ (без дополнительных блоков типа list или fact);
- тип кодирования, если источником является буфер памяти: РСМ.

Требования к качеству сигнала:

- значение ОСШ должно составлять не менее 10дБ;
- допускается присутствие посторонних звуков в виде однотональных гудков и сигналов тонального набора, DTMF.

Выходные данные: переменные `score_m` и `score_f` значения оценок близости голоса к модели мужского и женского голосов и возвращает код `NO_ERR=0`. Если  $score_m \geq score_f$ , то следует полагать, что голос, представленный в входном файле (буфере), является мужским, иначе – женским.

## 7. СООБЩЕНИЯ ОПЕРАТОРУ

Все сообщения оператору реализуются через коды возвращаемых значений функциями модулей, далее дается расшифровка.

Таблица 7.1 Коды возвращаемых значений функциями модуля 3i Gender ID SDK

<b>Код</b>	<b>Значение</b>
0	NO_ERR
1	UNKNOWN_ERROR
2	UNKNOWN_EXCEPTION
3	MEMORY_ALLOCATION_ERROR
4	INCORRECT_POINTER
5	INCORRECT_INPUT_PARAM
6	INCORRECT_INPUT_CONTAINER
7	INCORRECT_INIT_PARAMS
8	IS_BLOCKED
9	IS_NOT_INIT
10	ALREADY_INIT
11	INITIALIZATION_ERROR
12	DEINITIALIZATION_ERROR
29	UNABLE_TO_SAVE_FILE
31	UNABLE_TO_LOAD_FILE
32	INCORRECT_FILE_FORMAT
35	UNABLE_TO_GET_FEATURES
36	UNABLE_TO_GET_SPECTRAL_COEFS
37	UNABLE_TO_GET_SEGMENTATION
38	UNABLE_TO_GET_DTMF_SEGMENTATION
39	UNABLE_TO_GET_QS_SEGMENTATION
43	UNABLE_TO_LOAD_MODEL
48	CLUSTERING_ERROR

<b>Код</b>	<b>Значение</b>
49	DIR_IS_EMPTY
51	UNABLE_TO_CALC_ENERGY
56	OUT_OF_RANGE
57	CACHE_SIZE_TO_SMALL

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В настоящем документе приняты следующие условные обозначения:

ПО	Программное обеспечение
3i Gender ID SDK	ПО «3i Gender ID SDK»
API	сокр. англ. Application Programming Interface, интерфейс программирования приложений – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах
CPU	сокр. англ. Central Processing Unit – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ).
DNN	сокр. англ. Deep Neural Network, искусственная нейронная сеть с несколькими скрытыми слоями.
DTMF	сокр. англ. Dual-Tone Multi-Frequency, двухтональный многочастотный аналоговый сигнал, используемый для набора телефонного номера.
GMM	сокр. англ. Gaussian Mixture Model – статистическая модель плотности вероятности, выраженная суммой нормальных многомерных распределений.
MFCC	сокр. англ. Mel-Frequency Cepstrum Coefficients – представление кратковременного спектра, основанное на линейном косинусном преобразовании логарифмированного амплитудного спектра по мел-частотной шкале.
PCA	сокр. англ. Principal Component Analysis – метод сокращения размерности статистических данных.
PCM	сокр. англ. Pulse Code Modulation, импульсно-кодовая модуляция, термин применяется в смысле типа

	кодирования аудио-сигнала.
SDK	сокр. англ. Source Development Kit - комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения.
SO	сокр. англ. Shared object – динамическая библиотека в ОС на базе ядра Linux, позволяющая многократное использование различными программными приложениями.
UBM	сокр. англ. Universal Background Model – статистическая модель пространства признаков голоса.
WFST	сокр. англ. Weighted Finite State Transducer, взвешенный конечно-автоматный преобразователь, автомат Мили со взвешенными дугами.



