Cyberon DSpotter SDK (Arduino IDE Edition)

Programming Guide

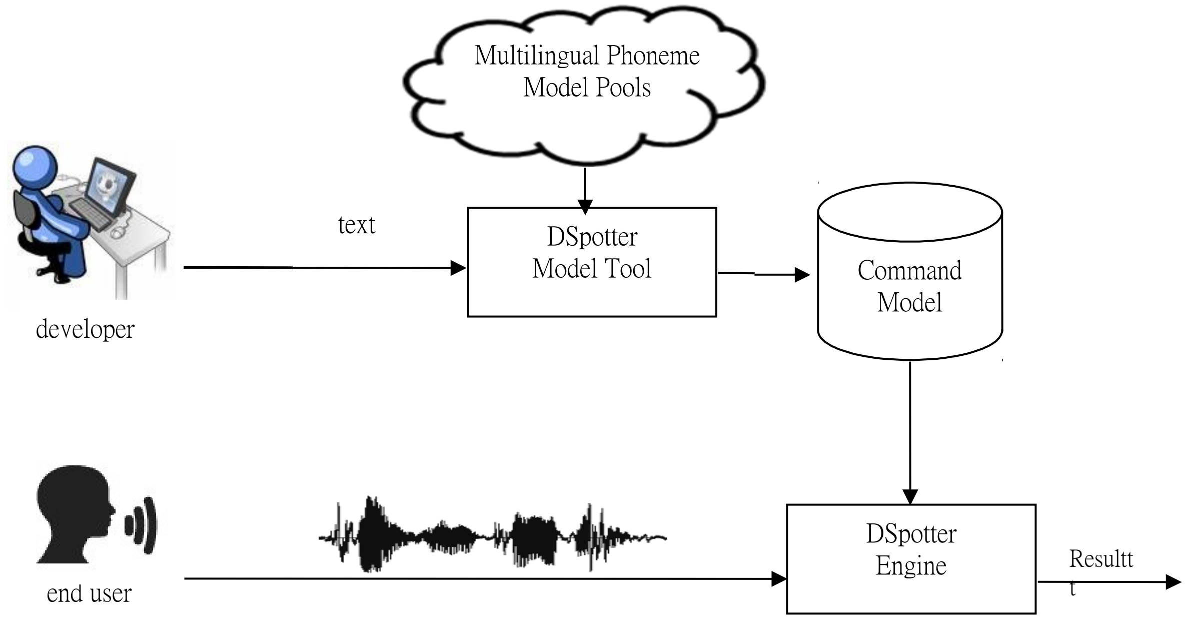
Ver no: 2.2.12
Release no: 2021090901
Date of issue: Sep 09, 2021

Leading Speech Solution provider
http://www.cyberon.com.tw/

Table of Contents

1. About Cyberon DSpotter SDK

DSpotter SDK is Cyberon's flagship high-performance embedded voice recognition solution specially optimized for mobile phones, automotives, smart home devices, consumer products, and interactive toys. Based on phoneme acoustic models, it enables developers to create applications of speaker-independent (SI) voice recognition capability without requiring costly data collection process for specific commands. With Win32-based DSpotter Model Tool, developers can easily and quickly create their own voice command models simply by text input. Other important features include always-on keyword-spotting capability, highly noise immune, adjustable sensitivity, voice quality assessment, and more than 30 commonly used language versions available.



2. Release History

Date	Version No.	Release No.	Author	Description
2021/09/09	2.2.12	2021090901	Tom	**Purpose:** First release.

3. Related Files

3.1. Related Files

Library

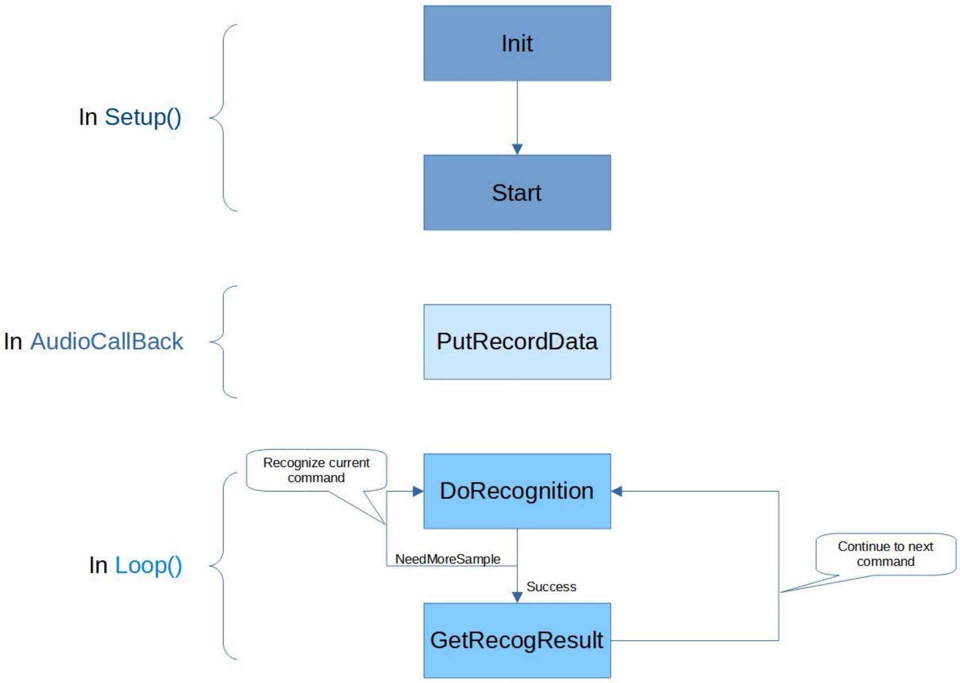
- **libDSpotterSDK.a**, the library for DSpotter SDK arduino edition.
- **CDSpotter.h,** the cpp api header for DSpotter SDK arduino edition.

Data

- **HeyRAKStar_OpenCamera_pack_withTxt_Enc_L0.h**: the header file that contains a level 0 model array which packs both trigger group and command group models together. To use the model array, developers only need to assign it to Init function. Note that this header file is in UTF8 format. The type of the model array is uint32_t and in little-endian manner.
- **HeyRAKStar_OpenCamera_pack_withTxt_Enc_L1.h**: the header file that contains a level 1 model array which packs both trigger group and command group models together. In comparison with the level 0 model under the same parameter settings, the level 1 model provides better recognition stability but also requires more computing and storage requirements.

4. DSpotter SDK API Standard Version

4.1. Calling Flow Chart of Standard API



4.2. APIs

GetMemoryUsage

Purpose

Get the memory usage of DSpotter.

Prototype

int CDSpotter::GetMemoryUsage(const uint32_t * lpdwModel, int nRecordCacheTimeMS)

Parameters

lpdwModel(in): The voice model.

nRecordCacheTimeMS(in): The cache length of record data, unit is millisecond. It must be greater than or equal to 60.

Return value

Return the memory usage.

GetVerInfor

Purpose

Get the version information of DSpotter.

Prototype

const char* CDSpotter::GetVerInfor()

Parameters

none.

Return value

Return the version string.

GetSerialNumber

Purpose

Get the serial number of Arduino device.

Prototype

const char* CDSpotter::GetSerialNumber()

Parameters

none.

Return value

Return the serial number string.

Init

Purpose

Initialize DSpotter.

Prototype

int Instance.Init(const uint32_t *lpdwLicense, int nLicenseSize, const uint32_t *lpdwModel, int nRecordCacheTimeMS, unsigned char *lpbyMemPool, int nMemSize)

Parameters

lpdwLicense(in): The license data.

nLicenseSize(in): The size of the license data.

lpdwModel(in): The voice model.

nRecordCacheTimeMS(in): The cache length of record data, unit is millisecond.

It must be greater than or equal to 60.

lpbyMemPool(in): The memory buffer that will be used by Dspotter engine.

nMemSize(in): The size of the memory buffer.

Return value

Return success or error code.

Start

Purpose

Start the process of recognition. Please call Start() before starting the recording device.

Prototype

int Instance.Start()

Parameters

none.

Return value

Return success or error code.

Stop

Purpose

Stop the process of recognition. PutRecordData() and DoRecognition() will do nothing after calling Stop().

Prototype

int Instance.Stop()

Parameters

none.

Return value

Return success or error code.

Release

Purpose

To release DSpotter, then the memory buffer(lpbyMemPool) can be reused by others.

Prototype

int Instance.Release()

Parameters

none.

Return value

Return success or error code.

SetAGC

Purpose

Set the option of auto gain control(AGC).

Prototype

int Instance.SetAGC(bool bEnableAGC = false, int nScalePercentage = 100)

Parameters

bEnableAGC(in): To enable or disable AGC. The default value is false.

nScalePercentage(in): The scale percentage of gain is 100 ~ 1600. The default value is 100.

Return value

Return success or error code.

SetCommandStageProperty

Purpose

Set the flow property at command stage.

Prototype

int Instance.SetCommandStageProperty(int nTimeout = 6000,

bool bCommandStageRepeatUntilTimeout = false)

Parameters

nTimeout(in): The maximum recording time in ms when there is no result at command stage. The valid range is 1000 to 30000. The default value is 6000.

bCommandStageRepeatUntilTimeout(in): If false, the recognition flow will switch to trigger stage immediately after recognizing the command. If true, it will recognize repeatedly at command stage until timeout. The default value is false.

Return value

Return success or error code.

PutRecordData

Purpose

Put the record data to the cached record buffer.

Prototype

int Instance.PutRecordData(const short *lpsSample, int nNumSample)

Parameters

lpsSample(in): The record data buffer.

nNumSample(in): The number of samples in the record data buffer.

Return value

Return success or error code

GetRecordLostCount

Purpose

Get the lost count when putting record data.

Prototype

int Instance.GetRecordLostCount()

Parameters

none.

Return value

Return the lost count.

DoRecognition

Purpose

Get 10 milliseconds of data from the cached record buffer and process it.

Prototype

int Instance.DoRecognition(int *pnCurrentState)

Parameters

pnCurrentState(out): The current stage.

Return value

Return success if get the recognition result, error code otherwise.

GetRecogResult

Purpose

Get the information of recognition result.

Prototype

int Instance.GetRecogResult(int *pnID, char *lpszCommand, int nCmdLength, int *pnConfi, int *pnSGDiff, int *pnCmdEnergy)

Parameters

pnID(out): The ID of command.

lpszCommand(out): The command buffer.

nCmdLength(in): The length of the command buffer.

pnConfi(out): The confidence score of command.

pnSGDiff(out): The human voice similarity of command.

pnCmdEnergy(out): The energy of command.

Return value

Return success or error code.

GetCommandCount

Purpose

Get the count of recognition command at different stage.

Prototype

int Instance.GetCommandCount(int nStage)

Parameters

nStage(in): CDSpotter::TriggerStage or CDSpotter::CommandStage.

Return value

Return the command counts in the given stage.

GetCommand

Purpose

Get the information of command.

Prototype

int Instance.GetCommand(int nStage, int nIndex, char *lpszCommand,

int nCmdLength, int *pnID)

Parameters

nStage(in): CDSpotter::TriggerStage or CDSpotter::CommandStage.

nIndex(in): From 0 to GetCommandCount(nStage) - 1.

lpszCommand(out): The command buffer.

nCmdLength(in): The length of command buffer.

pnID(out): The ID of command.

Return value

Return success or error code.

5. DSpotter SDK Error Code Table

Error Symbol	Error Code
CDSpotter::InitStage	-1
CDSpotter::TriggerStage	0
CDSpotter::CommandStage	1
CDSpotter::Success	0
CDSpotter::NotInit	-2001
CDSpotter::IllegalParam	-2002
CDSpotter::LeaveNoMemory	-2003
CDSpotter::LoadModelFailed	-2005
CDSpotter::NeedMoreSample	-2009
CDSpotter::Stopped	-2030
CDSpotter::LicenseFailed	-2200

6. DSpotter Supported Languages

Arabic
Bahasa(Indonesia)
Bahasa(Melayu)
Cantonese(HK)
Chinese(CHN)
Chinese(CHN)/English
Chinese(TWN)
Czech
Danish
Dutch
English(AU)
English(IN)
English(PHI)
English(SEA)
English(SG)
English(TWN)
English(UK)
English(US)
English(Worldwide)
Finnish
French
German
Greek
Hindi
Hungarian
Italian
Japanese
Japanese/English
Korean
Norwegian
Polish
Portuguese(BRA)
Portuguese(EU)
Russian
Slovak

Spanish(EU)
Spanish(LA)
Swedish
Taiwanese
Thai
Turkish
Ukrainian
Vietnamese