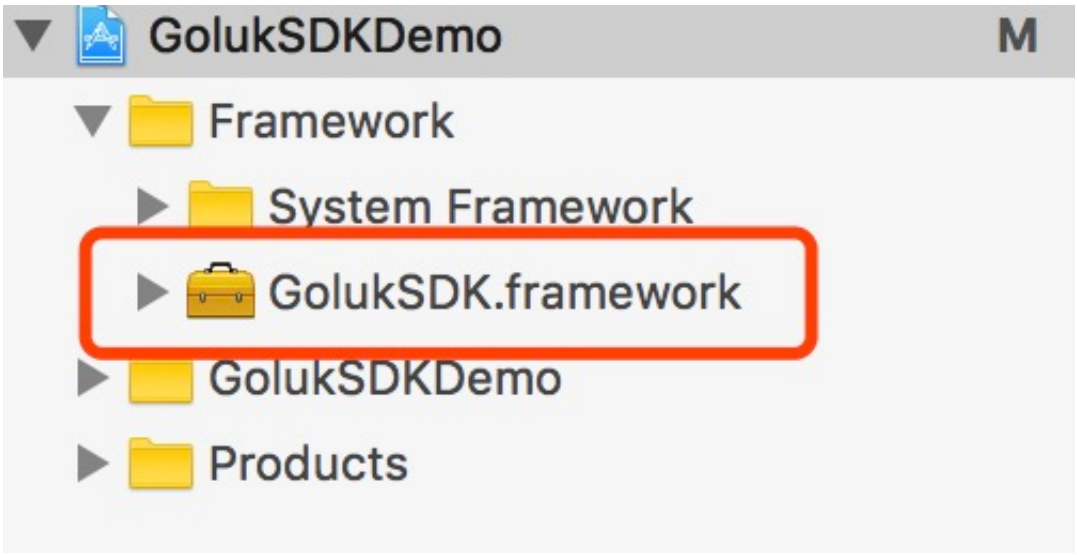
1. Framework Dependence

There are three databases that must be put into the program for it to functionproperly, as shown below:

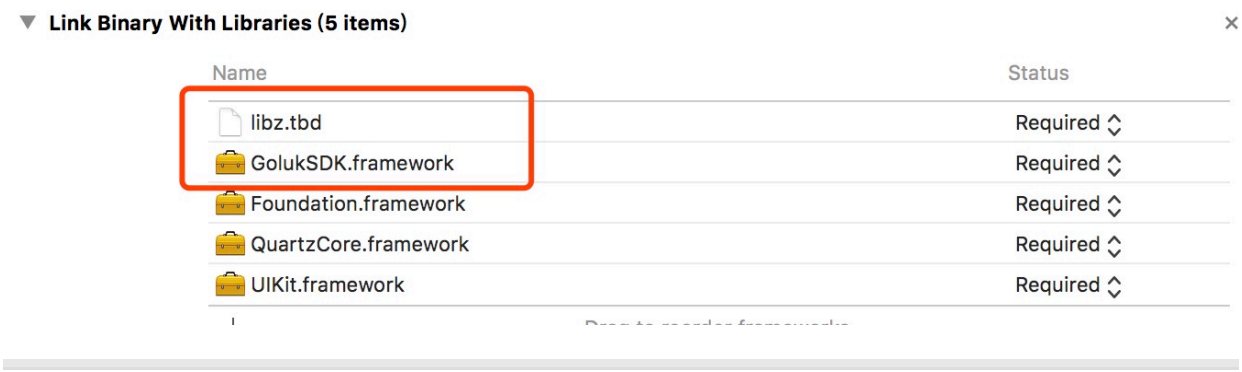


Project Settings

You must change the `Enable Bitcode` to `No` under the `Build Settings` as shown below

At the same time, you must add `-ObjC` to the `Other Linker Flags` under the `Build Settings`

You must also add the following dependence to the `Link Binary With Libraries` under the `Build Phases` as shown below:



2. SDK Initialization

Register APP

- **Function** : Initialize SDK, obtain APPID's corresponding permissions
- **Related Types** ： GKSDKEngine
- **Transfer Instructions**

Method	Param Instructions
+ (void)registerApp:(NSString*)appId	appId: published by the marketing department, every appId corresponds to an APP

- **Note : SDK Initialization failed, cannot provide service**
- **Reference Code** ：

```
    - (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [GKSDKEngine registerApp:@"100010XX"];
    return YES;
}
```

And also need call GKSetModeCommand after registered. And must set the seriesType to GKIPCSeriesT if using Goluk_T1*XXXXXX or Goluk_T2*XXXXXX, otherwise please set value to GKIPCSeriesG

```
[GKSDKEngine registerApp:@"10000005"];
GKSetModeCommand * cmd = [GKSetModeCommand new];
// Goluk_T1_XXXXXX, Goluk_T2_XXXXXX belong to T series
// Goluk_T3_XXXXXX belong to G series.
cmd.seriesType = GKIPCSeriesG;
[cmd send];
```

3. Recorder Connect and Disconnect

- **Function** : Controls recorder connection and disconnection
- **Commands** : GKConnectCommand (connect to recorder), GKDisconnectCommand (disconnect from recorder)
- **Param** ：

Command	Param	Type	Instructions
GKConnectCommand	None		

Command	Param	Type	Instructions
GKDisconnectCommand	None		

- **Note** ：

Transfer `[[GKConnectCommand new] send]` After the APP connects with the recorder, the SDK will send the notification GK_IPC_CONNECTED_NOTIFICATION

Transfer `[[GKDisconnectCommand new] send]` After the APP disconnects from the recorder, the SDK will send the notification GK_IPC_DISCONNECT_NOTIFICATION

- **ReferenceCode**：

```
    // Connect Goluk
    [[GKConnectCommand new] send];
```

```
    // Disconnect Goluk
    [[GKDisconnectCommand new] send];
```

4. Video Configure Setting

- **Function** : Controls the video configure setting
- **Commands** : GKGetVideoConfigCommand（get video configure）、GKSetVideoConfigCommand（set video configure）
- **Param**：

Command	Param	Type	Instructions
GKGetVideoConfigCommand	bitStreamType	GKBitStreamType	-

Command	Param	Type	Instructions
GKSetVideoConfigCommand	resolution	GKVideoQualityType	
	bitStreamType	GKBitStreamType	
	frameRate	NSInteger	10 <= frameRate <= 30
	bitrate	NSInteger	-

- **Return value instructions**：

Command	Return Value Type	Instructions
GKGetVideoConfigCommand	NSDictionary	See GKGetVideoConfigCommand return value instructions

Command	Return Value Type	Instructions
GKSetVideoConfigCommand	nil	None

GKGetVideoConfigCommand return value instructions

Key	Value Examples	Value Type	Instructions
AudioEnable	1	NSNumber	0 for off, 1 for on
bitrate		NSNumber	the bit rate of video
bitstreams	0 or 1	NSNumber	0 for primary bitstream, 1 for sub-bitstream
frameRate		NSNumber	the frame rate of video
resolution		NSString	the resolution of video, 1080P, 720P

- **Reference Code**：

```
    // get config
    GKGetVideoConfigCommand * cmd = [GKGetVideoConfigCommand new];
    // Command executes success
    [cmd setCommandFinish:^(id info) {
        // success
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Failed
    }];
    [cmd send];
```

```
    // set config
  GKSetVideoConfigCommand * cmd = [GKSetVideoConfigCommand new];
  cmd.resolution = GKVideoQuality_1080P;
  cmd.bitStreamType = GKBitStreamPrimary;
  cmd.frameRate = 30;
  cmd.bitrate = 12288;

    // Command executes success
    [cmd setCommandFinish:^(id info) {
        // success
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Failed
    }];
    [cmd send];
```

5. Time Settings

- **Function** : Configure recorder time
- **Commands** : GKGetTimeCommand (get recorder time), GKSetTimeCommand(set recorder

time)

- **Params**：

Command	Param	Type	Instructions
GKGetTimeCommand	None		

Command	Param	Type	Instructions
GKSetTimeCommand	time	NSDate	time

- **Return Value Instructions**：

Command	Return Value Type	Instructions
GKGetTimeCommand	NSDictionary	See GKGetTimeCommand Return value Instructions

Command	Return Value Type	Instructions
GKSetTimeCommand	nil	Return Value: nil

GKGetTimeCommand Return Value Instructions

Key	Value Examples	Value Type	Instructions
IPCTime	1464264251	NSNumber	Recorder time
zone	"Asia/Shanghai"	NSString	Recorder time zone

- **Reference Code**：

```
    // get config
    GKGetTimeCommand * getTimeCmd = [GKGetTimeCommand new];
    // Command executes success
    [cmd setCommandFinish:^(id info) {
        // success
        if ([info isKindOfClass:[NSDictionary class]]) {
            NSNumber *IPCTime = info[@"IPCTime"];
            NSDate *deviceTime = [NSDate
dateWithTimeIntervalSince1970:IPCTime.doubleValue];
        }
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Failed
    }];
    [cmd send];
```

```
// set config
GKSetTimeCommand *cmd = [GKSetTimeCommand new];
cmd.time = [NSDate date];
// Command executes success
[cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

6. Parking/Sleep Mode

- **Function** : Turns Park/Sleep Mode on and off
- **Commands** : GKGetFunctionCommand (obtain Park/Sleep Mode status)GKSetDormantCommand (configure Park/Sleep Mode)
- **Params** :

Command	Param	Type	Instructions
GKGetFunctionCommand	None		

Command	Param	Type	Instructions
GKSetDormantCommand	enable	BOOL	YES for on, NO for off

- **Return Value Instructions** :

Command	Return Value Type	Instructions
GKGetFunctionCommand	NSDictionary	See GKGetFunctionCommand Return value Instructions

Command	Return Value Type	Instructions
GKSetDormantCommand	nil	Return Value: nil

GKGetFunctionCommand return value instructions

Key	Value Examples	Value Type	Instructions
Dormant	1	NSNumber	0 for off, 1 for on

- **Reference Code** :

```
// get config
GKGetFunctionCommand * cmd = [GKGetFunctionCommand new];
// Command executes success
[cmd setCommandFinish:^(id info) {
    // success
    if ([info isKindOfClass:[NSDictionary class]]) {
        BOOL value = [info[@"Dormant"] boolValue];
    }
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

```
// set config
GKSetDormantCommand *cmd = [GKSetDormantCommand new];
cmd.enable = YES;
// Command executes success
 [cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

7. Language Settings

- **Function** : Changes language setting
- **Commands** GKGetVoiceTypeCommand (obtain language)GKSetVoiceTypeCommand (set language)
- **Params** :

Command	Param	Type	Instructions
GKGetVoiceTypeCommand	None		

Command	Param	Type	Instructions
GKSetVoiceTypeCommand	voiceType	NSNumber	0 for Chinese, 1 for English

- **Return Value Instructions** :

Command	Return Value Type	Instructions
GKGetVoiceTypeCommand	NSDictionary	See GKGetVoiceTypeCommand Return value Instructions

Command	Return Value Type	Instructions
GKSetVoiceTypeCommand	nil	Return Value: nil

GKGetVoiceTypeCommand return value instructions

Key	Value Examples	Value Type	Instructions
type	1	NSNumber	0 for Chinese, 1 for English

- **Reference Code**：

```
// get config
GKGetVoiceTypeCommand * cmd = [GKGetVoiceTypeCommand new];
// Command executes success
[cmd setCommandFinish:^(id info) {
    // success
    if ([info isKindOfClass:[NSDictionary class]]) {
        NSInteger language = [info[@"type"] integerValue];
    }
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

```
// set config
GKSetVoiceTypeCommand *cmd = [GKSetVoiceTypeCommand new];
cmd.voiceType = @(1);
// Command executes success
 [cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

8. Repeat Mode Switch

- **Function** : Turns auto-record mode
- **Commands** : GKGetRecordStateCommand (obtain record status)GKSetAutoRecordCommand (configure auto-record mode)
- **Params**

Command	Param	Type	Instructions
GKGetRecordStateCommand	None		

Command	Param	Type	Instructions
GKSetAutoRecordCommand	autoRecord	BOOL	YES for on, NO for off

- **Return Value Instructions**：

Command	Return Value Type	Instructions
GKGetRecordStateCommand	NSDictionary	See GKGetVoiceTypeCommand Return value Instructions

Command	Return Value Type	Instructions
GKSetAutoRecordCommand	nil	Return Value: nil

GKGetVoiceTypeCommand Return value Instructions

Key	Value Examples	Value Type	Instructions
status	1	NSNumber	0 for off, 1 for on

- **Reference Code**：

```
// get config
GKGetRecordStateCommand * cmd = [GKGetRecordStateCommand new];
// Command executes success
[cmd setCommandFinish:^(id info) {
    // success
    if ([info isKindOfClass:[NSDictionary class]]) {
        BOOL value = [info[@"status"] boolValue];
    }
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

```
// set config
GKSetAutoRecordCommand *cmd = [GKSetAutoRecordCommand new];
cmd.autoRecord = YES;
// Command executes success
 [cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

9. Check Version Info

- **Function** : Displays the recorder's current version
- **Commands** : GKGetVersionCommand
- **Params**

Command	Param	Type	Instructions
GKGetVersionCommand	None		

- **Return Value Instructions**：

Command	Return Value Type	Instructions
GKGetVersionCommand	NSDictionary	See GKGetVersionCommand Return value Instructions

GKGetVersionCommand Return value Instructions

Key	Value Examples	Value Type	Instructions
productname	T1	NSString	model
serial	ZTSDxxxxxxxx228G	NSString	serial number
version	"T1_XX.3.0425.XXXX"	NSString	ipc software version

- **Reference Code**：

```
GKGetVersionCommand *cmd = [GKGetVersionCommand new];
// Command executes success
[cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

10. Recover IPC Settings

- **Function** : Recover IPC Settings to defaults.
- **Commands** : GKRestoreCommand
- **Params** : :
- **Return Value Instructions** :

Command	Return Value Type	Instructions
GKRestoreCommand	nil	return value nil

- **Reference Code** :

```
GKRestoreCommand *cmd = [GKRestoreCommand new];
// Command executes success
 [cmd setCommandFinish:^(id info) {
    // success
}];
// Command executes Failed
[cmd setCommandError:^(NSError * error) {
    // Failed
}];
[cmd send];
```

11. Storage Capacity: Total, Used, Available

- **Function** : Obtain IPC storage capacity (total, used and available)
- **Commands** : GKGetRecUsageCommand
- **Params** : None
- **Return Value Instructions** :

: Obtain IPC storage capacity (total, used and available)

Key	Value Examples	Value Type	Instructions
totalSdSize	30668	MB	IPC total storage size
userFilesSize	25873	MB	Amount of memory used by IPC

Use totalSdSize and userFilesSize to calculate available space.

- **Reference Code** :

```
    GKGetRecUsageCommand * cmd = [GKGetRecUsageCommand new];
    // Command executes success
    [cmd setCommandFinish:^(NSDictionary * info) {
        // Success. Do your things
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Do error things.
    }];
    [cmd send];
```

The info process is as follows:

```
{
    SDCardActive = 1;
    isSpaceTooSmall = 0;
    leftSize = 4794;
    normalRecQuota = 23000;
    normalRecSize = 21601;
    picQuota = 1533;
    picSize = 49;
    totalSdSize = 30668;
    urgentRecQuota = 3680;
    urgentRecSize = 3472;
    userFilesSize = 25873;
    wonderfulRecQuota = 2453;
    wonderfulRecSize = 749;
}
```

12. View File Info

- **Function** : View file info: name, size, length, time created, style, etc.
- **Commands** : GKQuerySingleCommand
- **Params** :

Param	Type	Instructions
videoName	NSString	

- **Return Value Instructions** : In the setCommandFinish block, the info(NSDictionary) info packet is as follows

Key	Value Examples	Value Type	Instructions
location	URG_event*20160526165027*1*TX*1_0016.mp4		File path
period	16	second	Duration
resolution	1080p	1080p,720p,480p	Split Rate
time	1464252626	timeinterval	time
type	2	1.Auto-record 2.emergency video 4.hi-def video	Type
withSnapshot	1	1.screenshot 0.no-screenshot	Screenshot

- **Reference Code**：

```
    GKQuerySingleCommand * cmd = [GKQuerySingleCommand new];
    cmd.videoName = @"URG_event_20160526165027_1_TX_1_0016.mp4";
    // Command executes success
    [cmd setCommandFinish:^(NSDictionary * info) {
        // Success. Do your things
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Do error things.
    }];
    [cmd send];
```

The info process is as follows:

```
{
    id = 3245235;
    location = "URG_event_20160526165027_1_TX_1_0016.mp4";
    period = 16;
    resolution = 1080p;
    size = "25.5";
    time = 1464252626;
    timestamp = 20160526165026;
    total = 1;
    type = 2;
    withGps = 0;
    withSnapshot = 1;
    withThumb = 1;
}
```

13. View File List

- **Function** : list files by date (most recent to oldest)
- **Commands**： GKQueryCommand
- **Params**：

Param	Type	Instructions
type	NSInteger	1.Auto-record 2.emergency video 4.hi-def video
limitCount	NSInteger	View largest file
startTime	NSDate	Start time
endTime	NSDate	End time

- **Return Value Instructions**

In the setCommandFinish block, the info(NSDictionary) info packet is as follows:

Key	Value Examples	Value Type	Instructions
items	[item,item]	NSArray	file description array, see definition of item below
totol	1	NSInteger	file integer

item contents:

Key	Value Examples	Value Type	Instructions
location	URG_event*2016052616502*71*TX*1_0016.mp4		File path
period	16	second	duration
resolution	1080p	1080p,720p,480p	Split Rate
time	1464252626	timeinterval	time
type	2	1.Auto-record 2.emergency video 4.hi-def video	Type
withSnapshot	1	1.screenshot 0.no-screenshot	Screenshot

- **Reference Code**：

```
    GKQueryCommand * cmd = [GKQueryCommand new];
    cmd.type = 2;
    cmd.limitCount = 20;
    cmd.startTime = [NSDate dateWithTimeIntervalSince1970:0];
    cmd.endTime = [NSDate date];
    // Command executes success
    [cmd setCommandFinish:^(NSDictionary * info) {
        // Success. Do your things
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Do error things.
    }];
    [cmd send];
```

The info process is as follows:

```
{
    items = (
        {
            id = 3245235;
            location = "URG_event_20160526180326_1_TX_1_0016.mp4";
            period = 16;
            resolution = 1080p;
            size = "25.5";
            time = 1464257006;
            timestamp = 20160526180326;
            type = 2;
            withGps = 0;
            withSnapshot = 1;
            withThumb = 1;
        },
        {
            id = 3245236;
            location = "URG_event_20160526181326_1_TX_1_0016.mp4";
            period = 16;
            resolution = 1080p;
            size = "25.5";
            time = 1464257106;
            timestamp = 20160526181326;
            type = 2;
            withGps = 0;
            withSnapshot = 1;
            withThumb = 1;
        }
    );
    total = 2;
}
```

13. Download Files

- **Function** : download files and show download progress (if downloading pictures, cannot show download progress)
- **Commands**： GKAddDownloadFileCommand
- **Params**：

Param	Type	Instructions
videoType	NSInteger	1.Auto-record 2.emergency video 4.hi-def video
videoName	NSString	Name of downloaded video
picName	NSString	Name of downloaded pic

Note: you can only save one videoName and picName

- **Return Value Instructions**

In the setCommandProgress block, the info(NSDictionary) info packet is as follows:

Key	Value Examples	Value Type	Instructions
filerecvsize	262144		number of downloaded strings
filesize	26738688		total strings to be downloaded

In the setCommandFinish block, the info(NSDictionary) info packet is as follows:

Key	Value Examples	Value Type	Instructions
filename	URG_event*201605262002421TX*1_0016.mp4	NSString	file name
tag	downloadvideo	fixed value downloadvideo	tag

- **Reference Code**：

Download video, the commandProgress return call only works when downloading videos

```
    GKAddDownloadFileCommand * cmd = [GKAddDownloadFileCommand new];
    cmd.videoName = @"URG_event_20160526165027_1_TX_1_0016.mp4";
    // progress ONLY avaliable when download video.
    [cmd setCommandProgress:^(NSDictionary *progressInfo) {
        // Progress. Do your progress thing.
    }];
    // Command executes success
    [cmd setCommandFinish:^(NSDictionary * info) {
        // Success. Do your things
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Do error things.
    }];
    [cmd send];
```

Download picture

```
    GKAddDownloadFileCommand * cmd = [GKAddDownloadFileCommand new];
    cmd.picName = @"URG_event_20160526180122_1_TX_1_0016.jpg";
    // Command executes success
    [cmd setCommandFinish:^(NSDictionary * info) {
        // Success. Do your things
    }];
    // Command executes Failed
    [cmd setCommandError:^(NSError * error) {
        // Do error things.
    }];
    [cmd send];
```

The progressinfo process is as follows:

```
{
    fileid = 0;
    filename = "URG_event_20160526200242_1_TX_1_0016.mp4";
    filerecvsize = 262144;
    filesize = 26738688;
    tag = downloadvideo;
}
```

The info process is as follows:

```
{
    filename = "URG_event_20160526180318_1_TX_1_0016.jpg";
    tag = downloadvideoshot;
}
```

14. Cancel File Download

- **Function** : cancel download
- **Commands** : GKStopDownloadFileCommand
- **Params** : None
- **Return Value Instructions** : None
- **Reference Code** :

```
GKStopDownloadFileCommand * cmd = [GKStopDownloadFileCommand new];
[cmd send];
```

15. Remote preview camera

If you want to preview the camera, We supply the rtsp stream url for T and G series

SeriesType	RTSP stream url
T	rtsp://admin:123456@192.168.62.1/stream1
G	rtsp://admin:123456@192.168.62.1/sub