

IP-телестудия

Создано системой Doxygen 1.8.13



# Оглавление

1	Алфавитный указатель групп	1
1.1	Группы	1
2	Алфавитный указатель пространств имен	3
2.1	Пространства имен	3
3	Иерархический список классов	5
3.1	Иерархия классов	5
4	Алфавитный указатель классов	7
4.1	Классы	7
5	Список файлов	9
5.1	Файлы	9
6	Группы	11
6.1	Рекордер	11
6.1.1	Подробное описание	11
6.2	Сетка видеопотоков	12
6.2.1	Подробное описание	12
6.3	Сетевой монитор	13
6.3.1	Подробное описание	13

7	Пространства имен	15
7.1	Пространство имен from-gsuite	15
7.1.1	Функции	15
7.1.1.1	main()	15
7.1.2	Переменные	15
7.1.2.1	SCOPES	15
7.2	Пространство имен gpio	16
7.2.1	Функции	16
7.2.1.1	keypress()	16
7.2.1.2	main()	16
7.2.1.3	menu_cb()	17
7.2.1.4	rec_cb()	17
7.2.1.5	stop_cb()	17
7.2.2	Переменные	18
7.2.2.1	key_m	18
7.2.2.2	key_r	18
7.2.2.3	key_s	18
7.2.2.4	menu	18
7.2.2.5	rec	18
7.2.2.6	stop	18
7.3	Пространство имен ui	18
7.3.1	Функции	19
7.3.1.1	runPing()	19
7.3.2	Переменные	19
7.3.2.1	adresses	20
7.3.2.2	alpha	20
7.3.2.3	app	20
7.3.2.4	button	20
7.3.2.5	camera	20
7.3.2.6	cams	20

7.3.2.7	display0	21
7.3.2.8	display1	21
7.3.2.9	font	21
7.3.2.10	graph	21
7.3.2.11	grapher	21
7.3.2.12	i	21
7.3.2.13	label	21
7.3.2.14	labelStyle	21
7.3.2.15	leftAxis	22
7.3.2.16	menu	22
7.3.2.17	pinger	22
7.3.2.18	pingTimer	22
7.3.2.19	plot	22
7.3.2.20	rightAxis	22
7.3.2.21	stylesheet	22
7.3.2.22	threadpool	22
7.3.2.23	tickFont	23
7.3.2.24	True	23
7.3.2.25	units	23
7.3.2.26	widget	23
7.3.2.27	x	23
7.3.2.28	y	23
7.4	Пространство имен video-upload	23
7.4.1	Переменные	24
7.4.1.1	args	24
7.4.1.2	creds	24
7.4.1.3	DRIVE	24
7.4.1.4	filename	24
7.4.1.5	flow	24
7.4.1.6	help	24
7.4.1.7	metadata	24
7.4.1.8	parser	25
7.4.1.9	res	25
7.4.1.10	SCOPES	25
7.4.1.11	store	25
7.4.1.12	type	25

8	Классы	27
8.1	Структура AudioSource	27
8.1.1	Данные класса	27
8.1.1.1	name	27
8.1.1.2	uri	27
8.2	Класс ui.Camera	28
8.2.1	Подробное описание	28
8.2.2	Конструктор(ы)	28
8.2.2.1	__init__()	28
8.2.3	Методы	28
8.2.3.1	drawMe()	28
8.2.3.2	setPing()	29
8.2.4	Данные класса	29
8.2.4.1	button	29
8.2.4.2	grapher	29
8.2.4.3	ip	29
8.2.4.4	label	29
8.2.4.5	pingHistory	29
8.3	Структура Camera	30
8.3.1	Данные класса	30
8.3.1.1	button	30
8.3.1.2	drawingArea	30
8.3.1.3	fullName	31
8.3.1.4	name	31
8.3.1.5	player	31
8.3.1.6	recImage	31
8.3.1.7	record	31
8.3.1.8	uri	31
8.4	Класс Config	31
8.4.1	Подробное описание	32

8.4.2	Конструктор(ы)	33
8.4.2.1	Config() [1/4]	33
8.4.2.2	Config() [2/4]	33
8.4.2.3	Config() [3/4]	33
8.4.2.4	Config() [4/4]	34
8.4.3	Методы	34
8.4.3.1	get() [1/2]	34
8.4.3.2	get() [2/2]	34
8.4.3.3	getCustomRooms() [1/2]	34
8.4.3.4	getCustomRooms() [2/2]	34
8.4.3.5	getGSuiteRooms() [1/2]	35
8.4.3.6	getGSuiteRooms() [2/2]	35
8.4.3.7	getParam() [1/2]	35
8.4.3.8	getParam() [2/2]	35
8.4.3.9	getParamInt() [1/2]	35
8.4.3.10	getParamInt() [2/2]	35
8.4.3.11	getRooms() [1/2]	36
8.4.3.12	getRooms() [2/2]	36
8.4.3.13	makeGSuiteRequest() [1/2]	36
8.4.3.14	makeGSuiteRequest() [2/2]	36
8.4.3.15	operator=() [1/2]	36
8.4.3.16	operator=() [2/2]	36
8.4.3.17	readRoomsFromFile() [1/2]	36
8.4.3.18	readRoomsFromFile() [2/2]	37
8.4.3.19	setFile() [1/2]	37
8.4.3.20	setFile() [2/2]	37
8.4.4	Данные класса	37
8.4.4.1	configuration	37
8.4.4.2	rooms	37
8.5	Структура UI::display_player_data	38

8.5.1	Данные класса . . . . .	38
8.5.1.1	cam . . . . .	38
8.5.1.2	player . . . . .	38
8.5.1.3	playerLabel . . . . .	39
8.5.1.4	playingCamName . . . . .	39
8.6	Структура UI::gdrive_status_data . . . . .	39
8.6.1	Данные класса . . . . .	39
8.6.1.1	GDriveIcon . . . . .	40
8.6.1.2	recorder . . . . .	40
8.7	Класс ui.Grapher . . . . .	40
8.7.1	Подробное описание . . . . .	40
8.7.2	Конструктор(ы) . . . . .	40
8.7.2.1	__init__() . . . . .	40
8.7.3	Методы . . . . .	40
8.7.3.1	changeCam() . . . . .	41
8.7.3.2	setData() . . . . .	41
8.7.4	Данные класса . . . . .	41
8.7.4.1	activeCam . . . . .	41
8.7.4.2	curve . . . . .	41
8.7.4.3	data . . . . .	41
8.7.4.4	plot . . . . .	41
8.8	Структура PadData . . . . .	42
8.8.1	Данные класса . . . . .	42
8.8.1.1	delay . . . . .	42
8.8.1.2	src . . . . .	42
8.9	Класс ui.Pinger . . . . .	42
8.9.1	Подробное описание . . . . .	43
8.9.2	Конструктор(ы) . . . . .	43
8.9.2.1	__init__() . . . . .	43
8.9.3	Методы . . . . .	44



8.9.3.1	<code>_runCmd()</code>	44
8.9.3.2	<code>run()</code>	44
8.9.4	Данные класса	44
8.9.4.1	<code>camList</code>	44
8.10	Класс <code>Player</code>	45
8.10.1	Подробное описание	46
8.10.2	Конструктор(ы)	46
8.10.2.1	<code>Player()</code> [1/2]	46
8.10.2.2	<code>~Player()</code> [1/2]	47
8.10.2.3	<code>Player()</code> [2/2]	47
8.10.2.4	<code>~Player()</code> [2/2]	47
8.10.3	Методы	47
8.10.3.1	<code>buildPipeline()</code> [1/2]	47
8.10.3.2	<code>buildPipeline()</code> [2/2]	48
8.10.3.3	<code>busSyncHandler()</code> [1/2]	48
8.10.3.4	<code>busSyncHandler()</code> [2/2]	48
8.10.3.5	<code>isPlaying()</code>	48
8.10.3.6	<code>pad_added_handler()</code> [1/2]	48
8.10.3.7	<code>pad_added_handler()</code> [2/2]	49
8.10.3.8	<code>playStream()</code> [1/2]	49
8.10.3.9	<code>playStream()</code> [2/2]	49
8.10.3.10	<code>stopStream()</code>	49
8.10.3.11	<code>videoWidgetDraw_cb()</code> [1/2]	50
8.10.3.12	<code>videoWidgetDraw_cb()</code> [2/2]	50
8.10.3.13	<code>videoWidgetRealize_cb()</code> [1/2]	50
8.10.3.14	<code>videoWidgetRealize_cb()</code> [2/2]	50
8.10.4	Данные класса	50
8.10.4.1	<code>bus</code>	50
8.10.4.2	<code>camName</code>	50
8.10.4.3	<code>config</code>	51

8.10.4.4	dec	51
8.10.4.5	delay	51
8.10.4.6	parse	51
8.10.4.7	pipeline	51
8.10.4.8	platform	51
8.10.4.9	playing	51
8.10.4.10	scale	51
8.10.4.11	sink	52
8.10.4.12	src	52
8.10.4.13	uri	52
8.10.4.14	videoWindow	52
8.10.4.15	videoWindowHandle	52
8.11	Класс Recorder	52
8.11.1	Подробное описание	53
8.11.2	Конструктор(ы)	53
8.11.2.1	Recorder()	53
8.11.2.2	~Recorder()	54
8.11.3	Методы	54
8.11.3.1	checkIfRecStopped()	54
8.11.3.2	getRunningRecordings()	54
8.11.3.3	isGDriveUploadActive()	54
8.11.3.4	startRecording()	54
8.11.3.5	stopRecording()	55
8.11.3.6	uploadVideo()	55
8.11.4	Данные класса	55
8.11.4.1	config	55
8.11.4.2	runningGDriveUploads	55
8.11.4.3	runningRecordings	55
8.12	Класс Recording	56
8.12.1	Подробное описание	57

8.12.2	Конструктор(ы)	57
8.12.2.1	Recording()	57
8.12.2.2	~Recording()	57
8.12.3	Методы	57
8.12.3.1	buildPipeline()	58
8.12.3.2	busSyncHandler()	58
8.12.3.3	data_probe()	58
8.12.3.4	elementSinkLinked()	58
8.12.3.5	elementSrcLinked()	58
8.12.3.6	freeze_check()	59
8.12.3.7	getFileName()	59
8.12.3.8	getStatus()	59
8.12.3.9	pad_added_handler()	59
8.12.3.10	probe_block_stream()	60
8.12.3.11	probe_eos_in_stream()	60
8.12.3.12	probe_idle_relink()	60
8.12.3.13	relinkElements()	61
8.12.3.14	start()	61
8.12.3.15	stop()	61
8.12.4	Данные класса	61
8.12.4.1	camName	61
8.12.4.2	clock	61
8.12.4.3	depay	62
8.12.4.4	enc	62
8.12.4.5	fakesink	62
8.12.4.6	fileName	62
8.12.4.7	folder	62
8.12.4.8	freeze_check_id	62
8.12.4.9	in_idle_probe	62
8.12.4.10	lastBufferTime	62

8.12.4.11 mux . . . . .	63
8.12.4.12 parse . . . . .	63
8.12.4.13 pipeline . . . . .	63
8.12.4.14 sink . . . . .	63
8.12.4.15 src . . . . .	63
8.12.4.16 status . . . . .	63
8.12.4.17 streamcapsfilter . . . . .	63
8.12.4.18 streamLinked . . . . .	63
8.12.4.19 testcapsfilter . . . . .	64
8.12.4.20 testsrc . . . . .	64
8.12.4.21 timeout . . . . .	64
8.12.4.22 uri . . . . .	64
8.12.4.23 videoTimeLimit . . . . .	64
8.13 Класс Room . . . . .	64
8.13.1 Подробное описание . . . . .	65
8.13.2 Конструктор(ы) . . . . .	65
8.13.2.1 Room() [1/4] . . . . .	65
8.13.2.2 Room() [2/4] . . . . .	66
8.13.2.3 Room() [3/4] . . . . .	66
8.13.2.4 Room() [4/4] . . . . .	66
8.13.3 Методы . . . . .	66
8.13.3.1 getAudioSource() [1/2] . . . . .	66
8.13.3.2 getAudioSource() [2/2] . . . . .	66
8.13.3.3 getCameras() [1/2] . . . . .	66
8.13.3.4 getCameras() [2/2] . . . . .	66
8.13.3.5 getName() [1/2] . . . . .	67
8.13.3.6 getName() [2/2] . . . . .	67
8.13.3.7 setAudioSource() [1/2] . . . . .	67
8.13.3.8 setAudioSource() [2/2] . . . . .	67
8.13.3.9 setCameras() [1/2] . . . . .	67

8.13.3.10 setCameras() [2/2]	67
8.13.3.11 setName() [1/2]	67
8.13.3.12 setName() [2/2]	68
8.13.4 Данные класса	68
8.13.4.1 audio	68
8.13.4.2 cameras	68
8.13.4.3 name	68
8.13.4.4 type	68
8.14 Структура UI::switch_state_changed_data	69
8.14.1 Данные класса	69
8.14.1.1 cam	69
8.14.1.2 recorder	69
8.15 Класс UI	70
8.15.1 Подробное описание	71
8.15.2 Конструктор(ы)	71
8.15.2.1 UI() [1/2]	72
8.15.2.2 ~UI() [1/2]	72
8.15.2.3 UI() [2/2]	72
8.15.2.4 ~UI() [2/2]	72
8.15.3 Методы	72
8.15.3.1 displayPlayer() [1/2]	73
8.15.3.2 displayPlayer() [2/2]	73
8.15.3.3 displayRecordingStatus()	73
8.15.3.4 editButtonClicked()	73
8.15.3.5 findIP() [1/2]	73
8.15.3.6 findIP() [2/2]	74
8.15.3.7 initCamWidgets() [1/2]	74
8.15.3.8 initCamWidgets() [2/2]	74
8.15.3.9 initMenuWidgets() [1/2]	74
8.15.3.10 initMenuWidgets() [2/2]	75

8.15.3.11	<code>initPlayerWidgets()</code>	75
8.15.3.12	<code>initRoomTab()</code> [1/2]	75
8.15.3.13	<code>initRoomTab()</code> [2/2]	75
8.15.3.14	<code>initStyles()</code> [1/2]	75
8.15.3.15	<code>initStyles()</code> [2/2]	76
8.15.3.16	<code>keyPress()</code> [1/2]	76
8.15.3.17	<code>keyPress()</code> [2/2]	76
8.15.3.18	<code>on_show()</code>	76
8.15.3.19	<code>pageSwitched()</code>	76
8.15.3.20	<code>switchStateChanged()</code>	77
8.15.3.21	<code>updateGDriveStatus()</code>	77
8.15.3.22	<code>windowInit()</code> [1/2]	77
8.15.3.23	<code>windowInit()</code> [2/2]	77
8.15.4	Данные класса	77
8.15.4.1	<code>config</code>	78
8.15.4.2	<code>editButton</code>	78
8.15.4.3	<code>menuBuilder</code>	78
8.15.4.4	<code>menuWindow</code>	78
8.15.4.5	<code>player</code>	78
8.15.4.6	<code>playerBuilder</code>	78
8.15.4.7	<code>playerLabel</code>	78
8.15.4.8	<code>players</code>	78
8.15.4.9	<code>playerWidget</code>	79
8.15.4.10	<code>playerWindow</code>	79
8.15.4.11	<code>playingCamName</code>	79
8.15.4.12	<code>recorder</code>	79
8.15.4.13	<code>rooms</code>	79
8.15.4.14	<code>switchGridV</code>	79
8.16	Класс <code>ui.UIGraph</code>	80
8.16.1	Конструктор(ы)	80
8.16.1.1	<code>__init__()</code>	80
8.17	Класс <code>ui.UIMenu</code>	81
8.17.1	Конструктор(ы)	81
8.17.1.1	<code>__init__()</code>	81

9	Файлы	83
9.1	Файл groups.dox	83
9.2	Файл /home/jetson/controlroom/Studio/Network/src/ui.py	83
9.3	Файл /home/jetson/controlroom/Studio/Operator/Grid/include/config.h	84
9.4	Файл /home/jetson/controlroom/Studio/Recorder/include/config.h	85
9.5	Файл /home/jetson/controlroom/Studio/Operator/Grid/include/player.h	86
9.6	Файл /home/jetson/controlroom/Studio/Recorder/include/player.h	87
9.7	Файл /home/jetson/controlroom/Studio/Operator/Grid/include/room.h	88
9.7.1	Перечисления	90
9.7.1.1	room_t	90
9.8	Файл /home/jetson/controlroom/Studio/Recorder/include/room.h	90
9.8.1	Перечисления	91
9.8.1.1	room_t	91
9.9	Файл /home/jetson/controlroom/Studio/Operator/Grid/include/ui.h	91
9.10	Файл /home/jetson/controlroom/Studio/Recorder/include/ui.h	92
9.11	Файл /home/jetson/controlroom/Studio/Operator/Grid/src/config.cpp	93
9.12	Файл /home/jetson/controlroom/Studio/Recorder/src/config.cpp	93
9.13	Файл /home/jetson/controlroom/Studio/Operator/Grid/src/from-gsuite.py	94
9.14	Файл /home/jetson/controlroom/Studio/Recorder/src/from-gsuite.py	94
9.15	Файл /home/jetson/controlroom/Studio/Operator/Grid/src/main.cpp	95
9.15.1	Функции	95
9.15.1.1	main()	95
9.16	Файл /home/jetson/controlroom/Studio/Recorder/src/main.cpp	96
9.16.1	Функции	96
9.16.1.1	main()	96
9.17	Файл /home/jetson/controlroom/Studio/Operator/Grid/src/player.cpp	97
9.18	Файл /home/jetson/controlroom/Studio/Recorder/src/player.cpp	97
9.19	Файл /home/jetson/controlroom/Studio/Operator/Grid/src/ui.cpp	97
9.20	Файл /home/jetson/controlroom/Studio/Recorder/src/ui.cpp	98
9.21	Файл /home/jetson/controlroom/Studio/Recorder/include/recorder.h	98
9.22	Файл /home/jetson/controlroom/Studio/Recorder/include/recording.h	99
9.22.1	Перечисления	100
9.22.1.1	status_t	100
9.23	Файл /home/jetson/controlroom/Studio/Recorder/src/gpio.py	101
9.24	Файл /home/jetson/controlroom/Studio/Recorder/src/recorder.cpp	101
9.25	Файл /home/jetson/controlroom/Studio/Recorder/src/recording.cpp	102
9.26	Файл /home/jetson/controlroom/Studio/Recorder/src/video-upload.py	102
	Алфавитный указатель	103





# Глава 1

## Алфавитный указатель групп

### 1.1 Группы

Полный список групп.

Рекордер . . . . .	<a href="#">11</a>
Сетка видеопотоков . . . . .	<a href="#">12</a>
Сетевой монитор . . . . .	<a href="#">13</a>



## Глава 2

# Алфавитный указатель пространств имен

### 2.1 Пространства имен

Полный список пространств имен.

<a href="#">from-gsuite</a>	15
<a href="#">gpio</a>	16
<a href="#">ui</a>	18
<a href="#">video-upload</a>	23



## Глава 3

# Иерархический список классов

### 3.1 Иерархия классов

Иерархия классов.

AudioSource . . . . .	27
ui.Camera . . . . .	28
Camera . . . . .	30
Config . . . . .	31
UI::display_player_data . . . . .	38
UI::gdrive_status_data . . . . .	39
ui.Grapher . . . . .	40
PadData . . . . .	42
Player . . . . .	45
QMainWindow	
ui.UIMenu . . . . .	81
QWidget	
ui.UIGraph . . . . .	80
Recorder . . . . .	52
Recording . . . . .	56
Room . . . . .	64
UI::switch_state_changed_data . . . . .	69
UI . . . . .	70
QRunnable	
ui.Pinger . . . . .	42



## Глава 4

# Алфавитный указатель классов

### 4.1 Классы

Классы с их кратким описанием.

AudioSource	27
ui.Camera	28
Camera	30
Config	
Класс для хранения конфигурации	31
UI::display_player_data	38
UI::gdrive_status_data	39
ui.Grapher	40
PadData	42
ui.Pinger	
Пингует камеры	42
Player	
The Player class	45
Recorder	
Управляет процессом записи	52
Recording	
The Recording class	56
Room	
The Room class	64
UI::switch_state_changed_data	69
UI	
The UI class	70
ui.UIGraph	80
ui.UIMenu	81





## Глава 5

# Список файлов

### 5.1 Файлы

Полный список файлов.

/home/jetson/controlroom/Studio/Network/src/ <a href="#">ui.py</a> . . . . .	83
/home/jetson/controlroom/Studio/Operator/Grid/include/ <a href="#">config.h</a> . . . . .	84
/home/jetson/controlroom/Studio/Operator/Grid/include/ <a href="#">player.h</a> . . . . .	86
/home/jetson/controlroom/Studio/Operator/Grid/include/ <a href="#">room.h</a> . . . . .	88
/home/jetson/controlroom/Studio/Operator/Grid/include/ <a href="#">ui.h</a> . . . . .	91
/home/jetson/controlroom/Studio/Operator/Grid/src/ <a href="#">config.cpp</a> . . . . .	93
/home/jetson/controlroom/Studio/Operator/Grid/src/ <a href="#">from-gsuite.py</a> . . . . .	94
/home/jetson/controlroom/Studio/Operator/Grid/src/ <a href="#">main.cpp</a> . . . . .	95
/home/jetson/controlroom/Studio/Operator/Grid/src/ <a href="#">player.cpp</a> . . . . .	97
/home/jetson/controlroom/Studio/Operator/Grid/src/ <a href="#">ui.cpp</a> . . . . .	97
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">config.h</a> . . . . .	85
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">player.h</a> . . . . .	87
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">recorder.h</a> . . . . .	98
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">recording.h</a> . . . . .	99
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">room.h</a> . . . . .	90
/home/jetson/controlroom/Studio/Recorder/include/ <a href="#">ui.h</a> . . . . .	92
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">config.cpp</a> . . . . .	93
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">from-gsuite.py</a> . . . . .	94
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">gpio.py</a> . . . . .	101
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">main.cpp</a> . . . . .	96
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">player.cpp</a> . . . . .	97
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">recorder.cpp</a> . . . . .	101
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">recording.cpp</a> . . . . .	102
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">ui.cpp</a> . . . . .	98
/home/jetson/controlroom/Studio/Recorder/src/ <a href="#">video-upload.py</a> . . . . .	102



## Глава 6

# Группы

### 6.1 Рекордер

Выполняет запись видеопотоков, и их загрузку в Google Drive.

Граф связей класса Рекордер:



#### Классы

- class [Config](#)  
Класс для хранения конфигурации
- class [Recorder](#)  
Управляет процессом записи
- class [Recording](#)  
The [Recording](#) class.
- class [Room](#)  
The [Room](#) class.
- class [UI](#)  
The [UI](#) class.

#### 6.1.1 Подробное описание

Выполняет запись видеопотоков, и их загрузку в Google Drive.

## 6.2 Сетка видеопотоков

Отображает сетку изображений с камер

Граф связей класса Сетка видеопотоков:



### Классы

- class [Player](#)  
The [Player](#) class.
- class [Room](#)  
The [Room](#) class.
- class [UI](#)  
The [UI](#) class.

### 6.2.1 Подробное описание

Отображает сетку изображений с камер

## 6.3 Сетевой монитор

Выполняет ping камер и отображает график изменения значений для выбранной камеры

Классы

- class [ui.Pinger](#)  
Пингует камеры

### 6.3.1 Подробное описание

Выполняет ping камер и отображает график изменения значений для выбранной камеры



## Глава 7

# Пространства имен

### 7.1 Пространство имен from-gsuite

#### Функции

- `def main ()`

#### Переменные

- `list SCOPES = ['https://www.googleapis.com/auth/admin.directory.resource.calendar']`

#### 7.1.1 Функции

##### 7.1.1.1 main()

```
def from-gsuite.main ( )
```

Fetches cameras from GSuite and writes them to JSON file grouping by room

#### 7.1.2 Переменные

##### 7.1.2.1 SCOPES

```
list from-gsuite.SCOPEs = ['https://www.googleapis.com/auth/admin.directory.resource.calendar']
```

## 7.2 Пространство имен gpio

### Функции

- def `keypress` (key)
- def `stop_cb` (channel=0)
- def `rec_cb` (channel=0)
- def `menu_cb` (channel=0)
- def `main` ()

### Переменные

- int `stop` = 12
- int `rec` = 11
- int `menu` = 18
- string `key_r` = "key R "
- string `key_s` = "key S "
- string `key_m` = "key M "

### 7.2.1 Функции

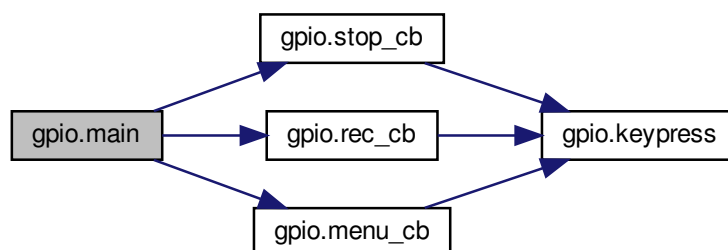
#### 7.2.1.1 keypress()

```
def gpio.keypress (  
    key )
```

#### 7.2.1.2 main()

```
def gpio.main ( )
```

Граф вызовов:





## 7.2.1.3 menu\_cb()

```
def gpio.menu_cb (  
    channel = 0 )
```

Граф вызовов:



## 7.2.1.4 rec\_cb()

```
def gpio.rec_cb (  
    channel = 0 )
```

Граф вызовов:



## 7.2.1.5 stop\_cb()

```
def gpio.stop_cb (  
    channel = 0 )
```

Граф вызовов:



## 7.2.2 Переменные

### 7.2.2.1 key\_m

```
string gpio.key_m = "key M "
```

### 7.2.2.2 key\_r

```
string gpio.key_r = "key R "
```

### 7.2.2.3 key\_s

```
string gpio.key_s = "key S "
```

### 7.2.2.4 menu

```
int gpio.menu = 18
```

### 7.2.2.5 rec

```
int gpio.rec = 11
```

### 7.2.2.6 stop

```
int gpio.stop = 12
```

## 7.3 Пространство имен ui

### Классы

- class [Camera](#)
- class [Grapher](#)
- class [Pinger](#)

Пингует камеры

- class [UIGraph](#)
- class [UIMenu](#)

## Функции

- `def runPing ()`

## Переменные

- dictionary `addresses`
- `app = QtWidgets.QApplication([])`
- `menu = UIMenu()`
- `stylesheet = open("../ui/stylesheet.css", "r")`
- `display0 = QtWidgets.QDesktopWidget().screenGeometry(0)`
- `graph = UIGraph()`
- `display1 = QtWidgets.QDesktopWidget().screenGeometry(1)`
- `widget = graph.findChild(pg.PlotWidget, "plotwidget")`
- `plot = widget.getPlotItem()`
- `rightAxis = plot.getAxis('right')`
- `leftAxis = plot.getAxis('left')`
- dictionary `labelStyle = {'color': '#FFF', 'font-size': '20pt'}`
- `units`
- `font = QFont()`
- `tickFont`
- `x`
- `True`
- `y`
- `alpha`
- `grapher = Grapher(plot)`
- list `cams = []`
- `int i = 0`
- `button = menu.findChild(QtWidgets.QPushButton, "b" + str(i))`
- `label = menu.findChild(QtWidgets.QLabel, "l" + str(i))`
- `camera = Camera(addresses[c], button, label, grapher)`
- `threadpool = QThreadPool()`
- `pingTimer = QTimer()`
- `pinger = Pinger(cams)`

### 7.3.1 Функции

#### 7.3.1.1 runPing()

```
def ui.runPing ()
```

### 7.3.2 Переменные

### 7.3.2.1 addresses

dictionary `ui.addresses`

Инициализатор

```
1 = {  
2   '51': '172.18.200.51',  
3   '52': '172.18.200.52',  
4   '53': '172.18.200.53',  
5   '54': '172.18.200.54',  
6   '55': '172.18.200.55',  
7   '56': '172.18.200.56',  
8   '30': '172.18.199.30',  
9   '32': '172.18.199.32',  
10  '42': '192.168.15.42',  
11 }
```

### 7.3.2.2 alpha

`ui.alpha`

### 7.3.2.3 app

`ui.app = QtWidgets.QApplication([])`

### 7.3.2.4 button

`ui.button = menu.findChild(QtWidgets.QPushButton, "b" + str(i))`

### 7.3.2.5 camera

`ui.camera = Camera(addresses[c], button, label, grapher)`

### 7.3.2.6 cams

`list ui.cams = []`

## 7.3.2.7 display0

```
ui.display0 = QtWidgets.QDesktopWidget().screenGeometry(0)
```

## 7.3.2.8 display1

```
ui.display1 = QtWidgets.QDesktopWidget().screenGeometry(1)
```

## 7.3.2.9 font

```
ui.font = QFont()
```

## 7.3.2.10 graph

```
ui.graph = UIGraph()
```

## 7.3.2.11 grapher

```
ui.grapher = Grapher(plot)
```

## 7.3.2.12 i

```
int ui.i = 0
```

## 7.3.2.13 label

```
ui.label = menu.findChild(QtWidgets.QLabel, "l" + str(i))
```

## 7.3.2.14 labelStyle

```
dictionary ui.labelStyle = {'color': '#FFF', 'font-size': '20pt'}
```

#### 7.3.2.15 leftAxis

```
ui.leftAxis = plot.GetAxis( 'left ' )
```

#### 7.3.2.16 menu

```
ui.menu = UIMenu\(\)
```

#### 7.3.2.17 pinger

```
ui.pinger = Pinger(cams)
```

#### 7.3.2.18 pingTimer

```
ui.pingTimer = QTimer()
```

#### 7.3.2.19 plot

```
ui.plot = widget.getPlotItem()
```

#### 7.3.2.20 rightAxis

```
ui.rightAxis = plot.GetAxis( 'right ' )
```

#### 7.3.2.21 stylesheet

```
ui.stylesheet = open("../ui/stylesheet.css", "r")
```

#### 7.3.2.22 threadpool

```
ui.threadpool = QThreadPool()
```

## 7.3.2.23 tickFont

```
ui.tickFont
```

## 7.3.2.24 True

```
ui.True
```

## 7.3.2.25 units

```
ui.units
```

## 7.3.2.26 widget

```
ui.widget = graph.findChild(pg.PlotWidget, "plotwidget")
```

## 7.3.2.27 x

```
ui.x
```

## 7.3.2.28 y

```
ui.y
```

## 7.4 Пространство имен video-upload

### Переменные

- `parser` = `argparse.ArgumentParser("Videofile name and location")`
- `help`
- `type`
- `args` = `parser.parse_args()`
- `string SCOPES` = `'https://www.googleapis.com/auth/drive.file'`
- `store` = `file.Storage('storage.json')`
- `creds` = `store.get()`
- `flow` = `client.flow_from_clientsecrets('client_secret.json', scope = SCOPES)`
- `DRIVE` = `build('drive', 'v3', http = creds.authorize(Http()))`
- `filename` = `args.location + args.filename`
- `dictionary metadata` = `{'name': args.filename}`
- `res` = `DRIVE.files().create(body = metadata, media_body = filename).execute()`

### 7.4.1 Переменные

#### 7.4.1.1 args

```
video-upload.args = parser.parse_args()
```

#### 7.4.1.2 creds

```
video-upload.creds = store.get()
```

#### 7.4.1.3 DRIVE

```
video-upload.DRIVE = build('drive', 'v3', http = creds.authorize(Http()))
```

#### 7.4.1.4 filename

```
video-upload.filename = args.location + args.filename
```

#### 7.4.1.5 flow

```
video-upload.flow = client.flow_from_clientsecrets('client_secret.json', scope = SCOPES)
```

#### 7.4.1.6 help

```
video-upload.help
```

#### 7.4.1.7 metadata

```
dictionary video-upload.metadata = {'name': args.filename}
```



## 7.4.1.8 parser

```
video-upload.parser = argparse.ArgumentParser("Videofile name and location")
```

## 7.4.1.9 res

```
video-upload.res = DRIVE.files().create(body = metadata, media_body = filename).execute()
```

## 7.4.1.10 SCOPES

```
string video-upload.SCOPES = 'https://www.googleapis.com/auth/drive.file'
```

## 7.4.1.11 store

```
video-upload.store = file.Storage('storage.json')
```

## 7.4.1.12 type

```
video-upload.type
```



## Глава 8

# Классы

### 8.1 Структура AudioSource

```
#include <room.h>
```

Открытые атрибуты

- string [name](#)
- string [uri](#)

#### 8.1.1 Данные класса

##### 8.1.1.1 name

```
string AudioSource::name
```

##### 8.1.1.2 uri

```
string AudioSource::uri
```

Объявления и описания членов структуры находятся в файле:

- `/home/jetson/controlroom/Studio/Operator/Grid/include/room.h`

## 8.2 Класс ui.Camera

### Открытые члены

- `def __init__ (self, ip, button, label, grapher)`
- `def setPing (self, error, time)`
- `def drawMe (self)`

### Открытые атрибуты

- `ip`
- `button`
- `label`
- `grapher`
- `pingHistory`

### 8.2.1 Подробное описание

Contains all camera information and ui objects

### 8.2.2 Конструктор(ы)

#### 8.2.2.1 \_\_init\_\_()

```
def ui.Camera.__init__ (
    self,
    ip,
    button,
    label,
    grapher )
```

### 8.2.3 Методы

#### 8.2.3.1 drawMe()

```
def ui.Camera.drawMe (
    self )
```

## 8.2.3.2 setPing()

```
def ui.Camera.setPing (
    self,
    error,
    time )
```

## 8.2.4 Данные класса

## 8.2.4.1 button

```
ui.Camera.button
```

## 8.2.4.2 grapher

```
ui.Camera.grapher
```

## 8.2.4.3 ip

```
ui.Camera.ip
```

## 8.2.4.4 label

```
ui.Camera.label
```

## 8.2.4.5 pingHistory

```
ui.Camera.pingHistory
```

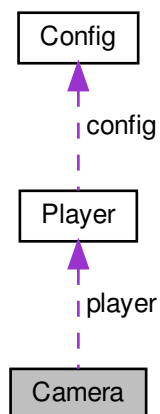
Объявления и описания членов класса находятся в файле:

- /home/jetson/controlroom/Studio/Network/src/[ui.py](#)

## 8.3 Структура Camera

```
#include <room.h>
```

Граф связей класса Camera:



Открытые атрибуты

- string `name`
- string `fullName`
- string `uri`
- `Player` \* `player`
- GtkWidget \* `button`
- GtkWidget \* `drawingArea`
- GtkWidget \* `recImage`
- bool `record` = false

### 8.3.1 Данные класса

#### 8.3.1.1 button

```
GtkWidget * Camera::button
```

#### 8.3.1.2 drawingArea

```
GtkWidget* Camera::drawingArea
```

## 8.3.1.3 fullName

```
string Camera::fullName
```

## 8.3.1.4 name

```
string Camera::name
```

## 8.3.1.5 player

```
Player* Camera::player
```

## 8.3.1.6 recImage

```
GtkWidget* Camera::recImage
```

## 8.3.1.7 record

```
bool Camera::record = false
```

## 8.3.1.8 uri

```
string Camera::uri
```

Объявления и описания членов структуры находятся в файле:

- /home/jetson/controlroom/Studio/Operator/Grid/include/[room.h](#)

## 8.4 Класс Config

Класс для хранения конфигурации

```
#include <config.h>
```

## Открытые члены

- void `setFile` (string configPath)
- string `getParam` (string name)
- int `getParamInt` (string name)
- vector< `Room` \* > \* `getRooms` ()
- void `setFile` (string configPath)
- string `getParam` (string name)
- int `getParamInt` (string name)
- vector< `Room` \* > \* `getRooms` ()

## Открытые статические члены

- static `Config` & `get` ()
- static `Config` & `get` ()

## Закрытые члены

- `Config` ()
- `Config` (const `Config` &)
- `Config` & `operator=` (const `Config` &)
- void `getGSuiteRooms` ()
- void `getCustomRooms` ()
- string `makeGSuiteRequest` ()
- vector< `Room` \* > `readRoomsFromFile` (string fileName)
- `Config` ()
- `Config` (const `Config` &)
- `Config` & `operator=` (const `Config` &)
- void `getGSuiteRooms` ()
- void `getCustomRooms` ()
- string `makeGSuiteRequest` ()
- vector< `Room` \* > `readRoomsFromFile` (string fileName)

## Закрытые данные

- map< string, string > `configuration`
- vector< `Room` \* > `rooms`

## 8.4.1 Подробное описание

## Класс для хранения конфигурации

Данный класс выполнен по шаблону singleton. Читает конфигурацию из файла и хранит параметры в словаре (std::map). Вызывает from\_gsutie.py для получения данных о камерах из GSutie. Читает файл gsuite\_rooms.json и создает объекты `Room` для каждой комнаты, которые затем хранит в массиве (std::vector)

## Форматирование файла конфигурации

Файл должен содержать параметры из таблицы ниже, отформатированные в виде #Комментарий  
Параметр = значение

## Допустимые параметры



Наименование	Значение	Описание
screens	1 или 2	Количество используемых экранов
saveToFolder	/путь	Путь к каталогу, в который будут сохраняться видео-файлы, начиная от корневого каталога системы (/)
windowWidth	Число (в пикселях)	Ширина и высота окон программы. Используется в случае, если параметр screens=1
windowHeight	^	^
videoTimeLimit	Число(в минутах)	Ограничение на длительность видео в одном файле.

Использование класса:

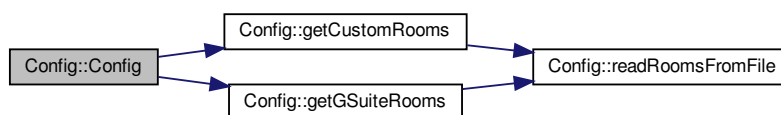
```
Config *config = &Config::get(); config->setFile("recorder.conf"); config->getRooms()
```

## 8.4.2 Конструктор(ы)

### 8.4.2.1 Config() [1/4]

```
Config::Config ( ) [private]
```

Граф вызовов:



### 8.4.2.2 Config() [2/4]

```
Config::Config (
    const Config & ) [private]
```

### 8.4.2.3 Config() [3/4]

```
Config::Config ( ) [private]
```

#### 8.4.2.4 Config() [4/4]

```
Config::Config (  
    const Config & ) [private]
```

### 8.4.3 Методы

#### 8.4.3.1 get() [1/2]

```
static Config& Config::get ( ) [inline], [static]
```

#### 8.4.3.2 get() [2/2]

```
static Config& Config::get ( ) [inline], [static]
```

#### 8.4.3.3 getCustomRooms() [1/2]

```
void Config::getCustomRooms ( ) [private]
```

Граф вызовов:



#### 8.4.3.4 getCustomRooms() [2/2]

```
void Config::getCustomRooms ( ) [private]
```

## 8.4.3.5 getGSuiteRooms() [1/2]

```
void Config::getGSuiteRooms ( ) [private]
```

Граф вызовов:



## 8.4.3.6 getGSuiteRooms() [2/2]

```
void Config::getGSuiteRooms ( ) [private]
```

## 8.4.3.7 getParam() [1/2]

```
string Config::getParam (
    string name )
```

## 8.4.3.8 getParam() [2/2]

```
string Config::getParam (
    string name )
```

## 8.4.3.9 getParamInt() [1/2]

```
int Config::getParamInt (
    string name )
```

## 8.4.3.10 getParamInt() [2/2]

```
int Config::getParamInt (
    string name )
```

## 8.4.3.11 getRooms() [1/2]

```
vector< Room * > * Config::getRooms ( )
```

## 8.4.3.12 getRooms() [2/2]

```
vector<Room*>* Config::getRooms ( )
```

## 8.4.3.13 makeGSuiteRequest() [1/2]

```
string Config::makeGSuiteRequest ( ) [private]
```

## 8.4.3.14 makeGSuiteRequest() [2/2]

```
string Config::makeGSuiteRequest ( ) [private]
```

## 8.4.3.15 operator=() [1/2]

```
Config& Config::operator= (
    const Config & ) [private]
```

## 8.4.3.16 operator=() [2/2]

```
Config& Config::operator= (
    const Config & ) [private]
```

## 8.4.3.17 readRoomsFromFile() [1/2]

```
vector< Room * > Config::readRoomsFromFile (
    string fileName ) [private]
```

## 8.4.3.18 readRoomsFromFile() [2/2]

```
vector<Room*> Config::readRoomsFromFile (
    string fileName ) [private]
```

## 8.4.3.19 setFile() [1/2]

```
void Config::setFile (
    string configPath )
```

## 8.4.3.20 setFile() [2/2]

```
void Config::setFile (
    string configPath )
```

## 8.4.4 Данные класса

## 8.4.4.1 configuration

```
map< string, string > Config::configuration [private]
```

## 8.4.4.2 rooms

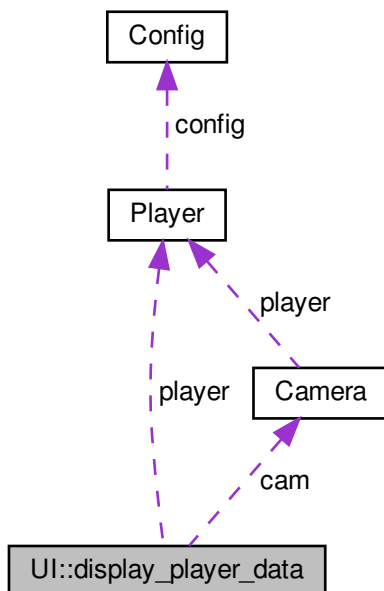
```
vector< Room * > Config::rooms [private]
```

Объявления и описания членов классов находятся в файлах:

- /home/jetson/controlroom/Studio/Operator/Grid/include/[config.h](#)
- /home/jetson/controlroom/Studio/Operator/Grid/src/[config.cpp](#)

## 8.5 Структура UI::display\_player\_data

Граф связей класса UI::display\_player\_data:



Открытые атрибуты

- [Camera](#) [cam](#)
- string \* [playingCamName](#)
- GtkWidget \* [playerLabel](#)
- [Player](#) \* [player](#)

### 8.5.1 Данные класса

#### 8.5.1.1 cam

[Camera](#) UI::display\_player\_data::cam

#### 8.5.1.2 player

[Player](#)\* UI::display\_player\_data::player

## 8.5.1.3 playerLabel

```
GtkWidget* UI::display_player_data::playerLabel
```

## 8.5.1.4 playingCamName

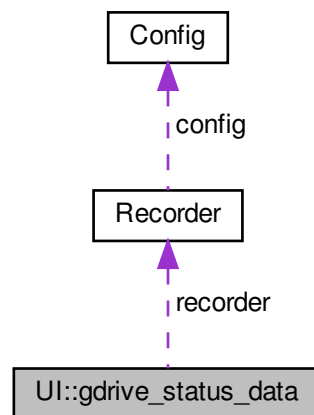
```
string * UI::display_player_data::playingCamName
```

Объявления и описания членов структуры находятся в файле:

- `/home/jetson/controlroom/Studio/Operator/Grid/include/ui.h`

## 8.6 Структура UI::gdrive\_status\_data

Граф связей класса UI::gdrive\_status\_data:



Открытые атрибуты

- `Recorder * recorder`
- `GtkWidget * GDriveIcon`

## 8.6.1 Данные класса

#### 8.6.1.1 GDriveIcon

GtkWidget\* UI::gdrive\_status\_data::GDriveIcon

#### 8.6.1.2 recorder

Recorder\* UI::gdrive\_status\_data::recorder

Объявления и описания членов структуры находятся в файле:

- /home/jetson/controlroom/Studio/Recorder/include/ui.h

### 8.7 Класс ui.Grapher

Открытые члены

- def `__init__` (self, `plot`)
- def `setData` (self, cam, `data`)
- def `changeCam` (self, cam)

Открытые атрибуты

- `plot`
- `curve`
- `activeCam`
- `data`

#### 8.7.1 Подробное описание

Draws a graph on PlotWidget

#### 8.7.2 Конструктор(ы)

##### 8.7.2.1 `__init__`()

```
def ui.Grapher.__init__ (
    self,
    plot )
```

#### 8.7.3 Методы



## 8.7.3.1 changeCam()

```
def ui.Grapher.changeCam (
    self,
    cam )
```

Set name or ip of camera, which data should be drawn

## 8.7.3.2 setData()

```
def ui.Grapher.setData (
    self,
    cam,
    data )
```

## 8.7.4 Данные класса

## 8.7.4.1 activeCam

ui.Grapher.activeCam

## 8.7.4.2 curve

ui.Grapher.curve

## 8.7.4.3 data

ui.Grapher.data

## 8.7.4.4 plot

ui.Grapher.plot

Объявления и описания членов класса находятся в файле:

- /home/jetson/controlroom/Studio/Network/src/[ui.py](#)

## 8.8 Структура PadData

```
#include <player.h>
```

Открытые атрибуты

- GstElement \* [src](#)
- GstElement \* [depay](#)

### 8.8.1 Данные класса

#### 8.8.1.1 depay

```
GstElement * PadData::depay
```

#### 8.8.1.2 src

```
GstElement * PadData::src
```

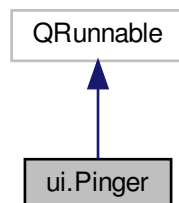
Объявления и описания членов структуры находятся в файле:

- [/home/jetson/controlroom/Studio/Operator/Grid/include/player.h](#)

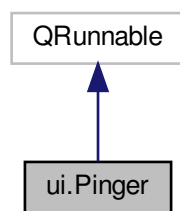
## 8.9 Класс ui.Pinger

Пингует камеры

Граф наследования: ui.Pinger:



Граф связей класса ui.Pinger:



Открытые члены

- `def __init__ (self, camList)`
- `def run (self)`

Открытые атрибуты

- `camList`

Закрытые члены

- `def _runCmd (self, cmd, stderr=STDOUT)`

### 8.9.1 Подробное описание

Пингует камеры

Updates ping information

### 8.9.2 Конструктор(ы)

#### 8.9.2.1 `__init__()`

```
def ui.Pinger.__init__ (
    self,
    camList )
```

### 8.9.3 Методы

#### 8.9.3.1 `_runCmd()`

```
def ui.Pinger._runCmd (
    self,
    cmd,
    stderr = STDOUT ) [private]
```

Run command and get output

#### 8.9.3.2 `run()`

```
def ui.Pinger.run (
    self )
```

Function for `QThreadPool` to run in separate thread

Граф вызовов:



### 8.9.4 Данные класса

#### 8.9.4.1 `camList`

`ui.Pinger.camList`

Объявления и описания членов класса находятся в файле:

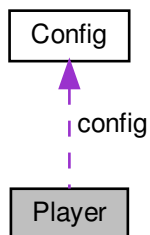
- `/home/jetson/controlroom/Studio/Network/src/ui.py`

## 8.10 Класс Player

The `Player` class.

```
#include <player.h>
```

Граф связей класса `Player`:



### Открытые члены

- `Player` (`GtkWidget *videoWindow`, `string platform`, `string uri`, `string camName`)
- `~Player` ()
- `void playStream` ()
- `void stopStream` ()
- `bool isPlaying` ()
- `Player` (`GtkWidget *videoWindow`, `Config *config`)
- `~Player` ()
- `void playStream` (`string uri`)

### Открытые атрибуты

- `GstElement * pipeline`
- `GstElement * src`
- `GstElement * depay`
- `GstElement * parse`
- `GstElement * dec`
- `GstElement * scale`
- `GstElement * sink`
- `Config * config`

### Закрытые члены

- `void buildPipeline` ()
- `void buildPipeline` ()

## Закрытые статические члены

- static void `videoWidgetRealize_cb` (GtkWidget \*widget, `Player` \*player)
- static gboolean `videoWidgetDraw_cb` (GtkWidget \*widget, cairo\_t \*cr, gpointer user\_data)
- static GstBusSyncReply `busSyncHandler` (GstBus \*bus, GstMessage \*message, `Player` \*player)
- static void `pad_added_handler` (GstElement \*src, GstPad \*new\_pad, `Player` \*player)
- static void `videoWidgetRealize_cb` (GtkWidget \*widget, `Player` \*player)
- static gboolean `videoWidgetDraw_cb` (GtkWidget \*widget, cairo\_t \*cr, gpointer user\_data)
- static GstBusSyncReply `busSyncHandler` (GstBus \*bus, GstMessage \*message, `Player` \*player)
- static void `pad_added_handler` (GstElement \*src, GstPad \*new\_pad, `Player` \*player)

## Закрытые данные

- bool `playing`
- string `uri`
- string `camName`
- GtkWidget \* `videoWindow`
- GstBus \* `bus`
- string `platform`
- guintptr `videoWindowHandle` = 0

### 8.10.1 Подробное описание

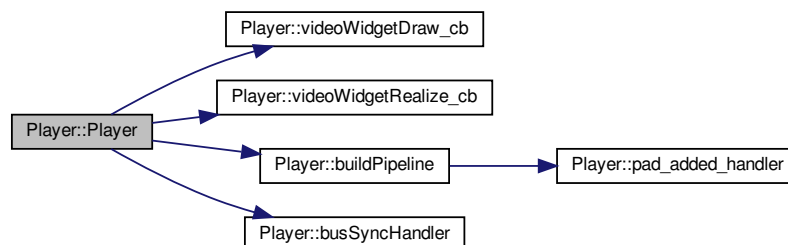
The `Player` class.

### 8.10.2 Конструктор(ы)

#### 8.10.2.1 `Player()` [1/2]

```
Player::Player (
    GtkWidget * videoWindow,
    string platform,
    string uri,
    string camName )
```

Граф вызовов:



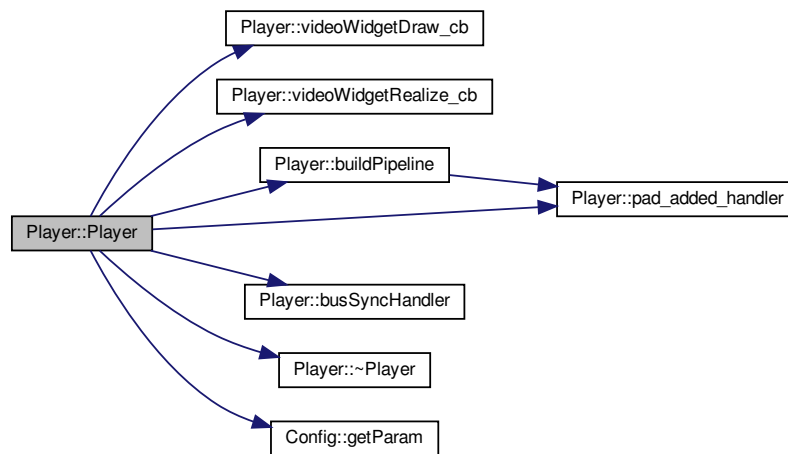
## 8.10.2.2 ~Player() [1/2]

```
Player::~Player ( )
```

## 8.10.2.3 Player() [2/2]

```
Player::Player (
    GtkWidget * videoWindow,
    Config * config )
```

Граф вызовов:



## 8.10.2.4 ~Player() [2/2]

```
Player::~Player ( )
```

## 8.10.3 Методы

## 8.10.3.1 buildPipeline() [1/2]

```
void Player::buildPipeline ( ) [private]
```

## 8.10.3.2 buildPipeline() [2/2]

```
void Player::buildPipeline ( ) [private]
```

Граф вызовов:



## 8.10.3.3 busSyncHandler() [1/2]

```
static GstBusSyncReply Player::busSyncHandler (
    GstBus * bus,
    GstMessage * message,
    Player * player ) [static], [private]
```

## 8.10.3.4 busSyncHandler() [2/2]

```
GstBusSyncReply Player::busSyncHandler (
    GstBus * bus,
    GstMessage * message,
    Player * player ) [static], [private]
```

## 8.10.3.5 isPlaying()

```
bool Player::isPlaying ( ) [inline]
```

## 8.10.3.6 pad\_added\_handler() [1/2]

```
static void Player::pad_added_handler (
    GstElement * src,
    GstPad * new_pad,
    Player * player ) [static], [private]
```



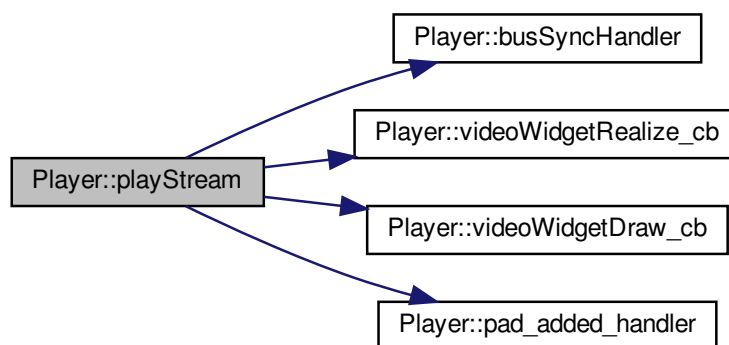
## 8.10.3.7 pad\_added\_handler() [2/2]

```
void Player::pad_added_handler (
    GstElement * src,
    GstPad * new_pad,
    Player * player ) [static], [private]
```

## 8.10.3.8 playStream() [1/2]

```
void Player::playStream (
    string uri )
```

Граф вызовов:



## 8.10.3.9 playStream() [2/2]

```
void Player::playStream ( )
```

## 8.10.3.10 stopStream()

```
void Player::stopStream ( )
```

## 8.10.3.11 videoWidgetDraw\_cb() [1/2]

```
static gboolean Player::videoWidgetDraw_cb (  
    GtkWidget * widget,  
    cairo_t * cr,  
    gpointer user_data ) [static], [private]
```

## 8.10.3.12 videoWidgetDraw\_cb() [2/2]

```
gboolean Player::videoWidgetDraw_cb (  
    GtkWidget * widget,  
    cairo_t * cr,  
    gpointer user_data ) [static], [private]
```

## 8.10.3.13 videoWidgetRealize\_cb() [1/2]

```
static void Player::videoWidgetRealize_cb (  
    GtkWidget * widget,  
    Player * player ) [static], [private]
```

## 8.10.3.14 videoWidgetRealize\_cb() [2/2]

```
void Player::videoWidgetRealize_cb (  
    GtkWidget * widget,  
    Player * player ) [static], [private]
```

## 8.10.4 Данные класса

## 8.10.4.1 bus

```
GstBus * Player::bus [private]
```

## 8.10.4.2 camName

```
string Player::camName [private]
```

## 8.10.4.3 config

[Config](#)\* Player::config

## 8.10.4.4 dec

GstElement \* Player::dec

## 8.10.4.5 depay

GstElement \* Player::depay

## 8.10.4.6 parse

GstElement \* Player::parse

## 8.10.4.7 pipeline

GstElement \* Player::pipeline

## 8.10.4.8 platform

string Player::platform [private]

## 8.10.4.9 playing

bool Player::playing [private]

## 8.10.4.10 scale

GstElement \* Player::scale

## 8.10.4.11 sink

```
GstElement * Player::sink
```

## 8.10.4.12 src

```
GstElement * Player::src
```

## 8.10.4.13 uri

```
string Player::uri [private]
```

## 8.10.4.14 videoWindow

```
GtkWidget * Player::videoWindow [private]
```

## 8.10.4.15 videoWindowHandle

```
guintptr Player::videoWindowHandle = 0 [private]
```

Объявления и описания членов классов находятся в файлах:

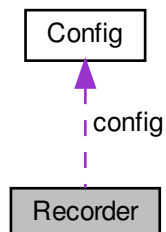
- /home/jetson/controlroom/Studio/Operator/Grid/include/[player.h](#)
- /home/jetson/controlroom/Studio/Operator/Grid/src/[player.cpp](#)

## 8.11 Класс Recorder

Управляет процессом записи

```
#include <recorder.h>
```

Граф связей класса Recorder:



## Открытые члены

- [Recorder](#) ([Config](#) \*config)
- [~Recorder](#) ()
- [map](#)< [Camera](#) \*, [Recording](#) \* > [getRunningRecordings](#) ()
- [bool](#) [startRecording](#) ([Camera](#) \*cam)
- [bool](#) [stopRecording](#) ([Camera](#) \*cam)
- [bool](#) [isGDriveUploadActive](#) ()

## Закрытые члены

- [void](#) [uploadVideo](#) (string uri, string fileName)

## Закрытые статические члены

- [static gboolean](#) [checkIfRecStopped](#) (gpointer data)

## Закрытые данные

- [map](#)< [Camera](#) \*, [Recording](#) \* > [runningRecordings](#)
- [Config](#) \* config
- [int](#) [runningGDriveUploads](#) = 0

## 8.11.1 Подробное описание

Управляет процессом записи

Forks gstreamer pipelines for each recording Needs a stream id to start a recording Stores pids of running recorder processes Stops recordings by stream id

## 8.11.2 Конструктор(ы)

## 8.11.2.1 Recorder()

```
Recorder::Recorder (
    Config * config )
```

Граф вызовов:



## 8.11.2.2 ~Recorder()

```
Recorder::~~Recorder ( )
```

## 8.11.3 Методы

## 8.11.3.1 checkIfRecStopped()

```
gboolean Recorder::checkIfRecStopped (
    gpointer data ) [static], [private]
```

## 8.11.3.2 getRunningRecordings()

```
map<Camera*, Recording*> Recorder::getRunningRecordings ( ) [inline]
```

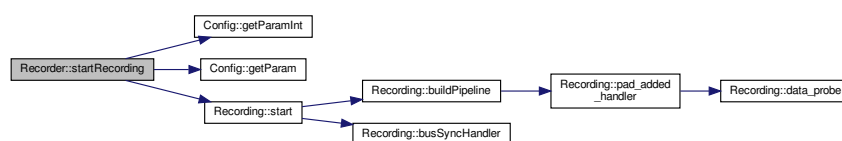
## 8.11.3.3 isGDriveUploadActive()

```
bool Recorder::isGDriveUploadActive ( ) [inline]
```

## 8.11.3.4 startRecording()

```
bool Recorder::startRecording (
    Camera * cam )
```

Граф вызовов:



## 8.11.3.5 stopRecording()

```
bool Recorder::stopRecording (
    Camera * cam )
```

## 8.11.3.6 uploadVideo()

```
void Recorder::uploadVideo (
    string uri,
    string fileName ) [private]
```

Граф вызовов:



## 8.11.4 Данные класса

## 8.11.4.1 config

```
Config* Recorder::config [private]
```

## 8.11.4.2 runningGDriveUploads

```
int Recorder::runningGDriveUploads = 0 [private]
```

## 8.11.4.3 runningRecordings

```
map<Camera*, Recording*> Recorder::runningRecordings [private]
```

Объявления и описания членов классов находятся в файлах:

- /home/jetson/controlroom/Studio/Recorder/include/recorder.h
- /home/jetson/controlroom/Studio/Recorder/src/recorder.cpp

## 8.12 Класс Recording

The [Recording](#) class.

```
#include <recording.h>
```

### Открытые члены

- [Recording](#) (string [uri](#), string [folder](#), string [camName](#), long [timeout](#), long [videoTimeLimit](#))
- [~Recording](#) ()
- string [getFileName](#) ()
- [status\\_t](#) [getStatus](#) ()
- bool [start](#) ()
- bool [stop](#) ()

### Открытые атрибуты

- [status\\_t](#) [status](#)
- string [uri](#)
- string [camName](#)
- string [fileName](#)
- string [folder](#)

### Закрытые члены

- bool [buildPipeline](#) ()

### Закрытые статические члены

- static void [pad\\_added\\_handler](#) (GstElement \*src, GstPad \*new\_pad, [Recording](#) \*recording)
- static GstBusSyncReply [busSyncHandler](#) (GstBus \*bus, GstMessage \*message, [Recording](#) \*recording)
- static GstPadProbeReturn [probe\\_block\\_stream](#) (GstPad \*pad, GstPadProbeInfo \*info, gpointer user\_data)
- static GstPadProbeReturn [probe\\_eos\\_in\\_stream](#) (GstPad \*pad, GstPadProbeInfo \*info, gpointer user\_data)
- static GstPadProbeReturn [probe\\_idle\\_relink](#) (GstPad \*pad, GstPadProbeInfo \*info, gpointer user\_data)
- static GstPadProbeReturn [data\\_probe](#) (GstPad \*pad, GstPadProbeInfo \*info, gpointer user\_data)
- static gboolean [freeze\\_check](#) (gpointer user\_data)
- static bool [elementSrcLinked](#) (GstElement \*elem)
- static bool [elementSinkLinked](#) (GstElement \*elem)
- static bool [relinkElements](#) (GstElement \*wrong\_src, GstElement \*right\_src, GstElement \*sink)



## Закрытые данные

- long `timeout`
- long `videoTimeLimit`
- int `freeze_check_id`
- bool `streamLinked`
- GstElement \* `pipeline`
- GstElement \* `src`
- GstElement \* `testsrc`
- GstElement \* `depay`
- GstElement \* `parse`
- GstElement \* `streamcapsfilter`
- GstElement \* `mux`
- GstElement \* `sink`
- GstElement \* `enc`
- GstElement \* `fakesink`
- GstElement \* `testcapsfilter`
- GstClock \* `clock`
- GstClockTime `lastBufferTime`
- gint `in_idle_probe`

### 8.12.1 Подробное описание

The `Recording` class.

### 8.12.2 Конструктор(ы)

#### 8.12.2.1 Recording()

```
Recording::Recording (
    string uri,
    string folder,
    string camName,
    long timeout,
    long videoTimeLimit )
```

#### 8.12.2.2 ~Recording()

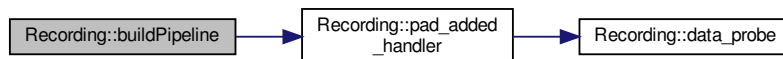
```
Recording::~Recording ( )
```

### 8.12.3 Методы

## 8.12.3.1 buildPipeline()

```
bool Recording::buildPipeline ( ) [private]
```

Граф вызовов:



## 8.12.3.2 busSyncHandler()

```
GstBusSyncReply Recording::busSyncHandler (
    GstBus * bus,
    GstMessage * message,
    Recording * recording ) [static], [private]
```

## 8.12.3.3 data\_probe()

```
GstPadProbeReturn Recording::data_probe (
    GstPad * pad,
    GstPadProbeInfo * info,
    gpointer user_data ) [static], [private]
```

## 8.12.3.4 elementSinkLinked()

```
bool Recording::elementSinkLinked (
    GstElement * elem ) [static], [private]
```

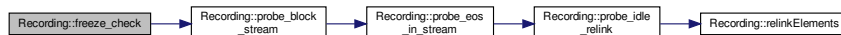
## 8.12.3.5 elementSrcLinked()

```
bool Recording::elementSrcLinked (
    GstElement * elem ) [static], [private]
```

## 8.12.3.6 freeze\_check()

```
gboolean Recording::freeze_check (
    gpointer user_data ) [static], [private]
```

Граф вызовов:



## 8.12.3.7 getFileName()

```
string Recording::getFileName ( ) [inline]
```

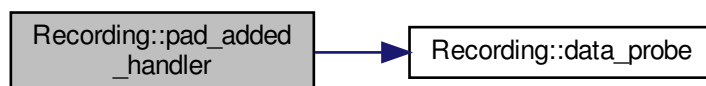
## 8.12.3.8 getStatus()

```
status_t Recording::getStatus ( ) [inline]
```

## 8.12.3.9 pad\_added\_handler()

```
void Recording::pad_added_handler (
    GstElement * src,
    GstPad * new_pad,
    Recording * recording ) [static], [private]
```

Граф вызовов:



## 8.12.3.10 probe\_block\_stream()

```
GstPadProbeReturn Recording::probe_block_stream (  
    GstPad * pad,  
    GstPadProbeInfo * info,  
    gpointer user_data ) [static], [private]
```

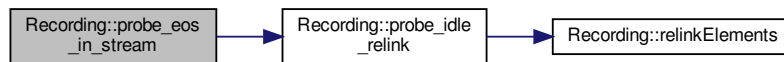
Граф вызовов:



## 8.12.3.11 probe\_eos\_in\_stream()

```
GstPadProbeReturn Recording::probe_eos_in_stream (  
    GstPad * pad,  
    GstPadProbeInfo * info,  
    gpointer user_data ) [static], [private]
```

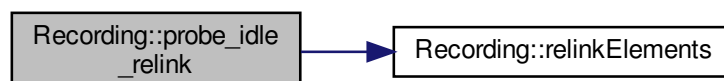
Граф вызовов:



## 8.12.3.12 probe\_idle\_relink()

```
GstPadProbeReturn Recording::probe_idle_relink (  
    GstPad * pad,  
    GstPadProbeInfo * info,  
    gpointer user_data ) [static], [private]
```

Граф вызовов:



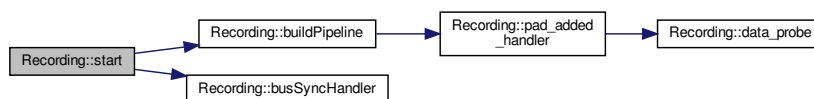
## 8.12.3.13 relinkElements()

```
bool Recording::relinkElements (
    GstElement * wrong_src,
    GstElement * right_src,
    GstElement * sink ) [static], [private]
```

## 8.12.3.14 start()

```
bool Recording::start ( )
```

Граф вызовов:



## 8.12.3.15 stop()

```
bool Recording::stop ( )
```

## 8.12.4 Данные класса

## 8.12.4.1 camName

```
string Recording::camName
```

## 8.12.4.2 clock

```
GstClock* Recording::clock [private]
```

## 8.12.4.3 depay

GstElement \* Recording::depay [private]

## 8.12.4.4 enc

GstElement\* Recording::enc [private]

## 8.12.4.5 fakesink

GstElement \* Recording::fakesink [private]

## 8.12.4.6 fileName

string Recording::fileName

## 8.12.4.7 folder

string Recording::folder

## 8.12.4.8 freeze\_check\_id

int Recording::freeze\_check\_id [private]

## 8.12.4.9 in\_idle\_probe

gint Recording::in\_idle\_probe [private]

## 8.12.4.10 lastBufferTime

GstClockTime Recording::lastBufferTime [private]

## 8.12.4.11 mux

GstElement \* Recording::mux [private]

## 8.12.4.12 parse

GstElement \* Recording::parse [private]

## 8.12.4.13 pipeline

GstElement\* Recording::pipeline [private]

## 8.12.4.14 sink

GstElement \* Recording::sink [private]

## 8.12.4.15 src

GstElement \* Recording::src [private]

## 8.12.4.16 status

[status\\_t](#) Recording::status

## 8.12.4.17 streamcapsfilter

GstElement \* Recording::streamcapsfilter [private]

## 8.12.4.18 streamLinked

bool Recording::streamLinked [private]

## 8.12.4.19 testcapsfilter

```
GstElement * Recording::testcapsfilter [private]
```

## 8.12.4.20 testsrc

```
GstElement * Recording::testsrc [private]
```

## 8.12.4.21 timeout

```
long Recording::timeout [private]
```

## 8.12.4.22 uri

```
string Recording::uri
```

## 8.12.4.23 videoTimeLimit

```
long Recording::videoTimeLimit [private]
```

Объявления и описания членов классов находятся в файлах:

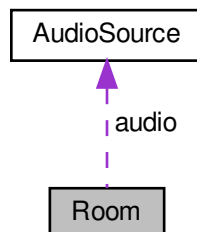
- `/home/jetson/controlroom/Studio/Recorder/include/recording.h`
- `/home/jetson/controlroom/Studio/Recorder/src/recording.cpp`

## 8.13 Класс Room

The [Room](#) class.

```
#include <room.h>
```

Граф связей класса Room:





## Открытые члены

- [Room](#) (string [name](#), vector< [Camera](#) > [cameras](#), [AudioSource](#) [audio](#))
- [Room](#) ()
- string [getName](#) ()
- void [setName](#) (string [name](#))
- vector< [Camera](#) > \* [getCameras](#) ()
- void [setCameras](#) (vector< [Camera](#) > [cams](#))
- [AudioSource](#) [getAudioSource](#) ()
- void [setAudioSource](#) ([AudioSource](#) [audio](#))
- [Room](#) (string [name](#), vector< [Camera](#) > [cameras](#), [AudioSource](#) [audio](#))
- [Room](#) ()
- string [getName](#) ()
- void [setName](#) (string [name](#))
- vector< [Camera](#) > \* [getCameras](#) ()
- void [setCameras](#) (vector< [Camera](#) > [cams](#))
- [AudioSource](#) [getAudioSource](#) ()
- void [setAudioSource](#) ([AudioSource](#) [audio](#))

## Открытые атрибуты

- [room\\_t](#) type

## Закрытые данные

- string [name](#)
- [AudioSource](#) [audio](#)
- vector< [Camera](#) > [cameras](#)

## 8.13.1 Подробное описание

The [Room](#) class.

## 8.13.2 Конструктор(ы)

## 8.13.2.1 Room() [1/4]

```
Room::Room (
    string name,
    vector< Camera > cameras,
    AudioSource audio ) [inline]
```

#### 8.13.2.2 Room() [2/4]

Room::Room ( ) [inline]

#### 8.13.2.3 Room() [3/4]

Room::Room (  
    string name,  
    vector< Camera > cameras,  
    AudioSource audio ) [inline]

#### 8.13.2.4 Room() [4/4]

Room::Room ( ) [inline]

### 8.13.3 Методы

#### 8.13.3.1 getAudioSource() [1/2]

AudioSource Room::getAudioSource ( ) [inline]

#### 8.13.3.2 getAudioSource() [2/2]

AudioSource Room::getAudioSource ( ) [inline]

#### 8.13.3.3 getCameras() [1/2]

vector<Camera>\* Room::getCameras ( ) [inline]

#### 8.13.3.4 getCameras() [2/2]

vector<Camera>\* Room::getCameras ( ) [inline]

## 8.13.3.5 getName() [1/2]

```
string Room::getName ( ) [inline]
```

## 8.13.3.6 getName() [2/2]

```
string Room::getName ( ) [inline]
```

## 8.13.3.7 setAudioSource() [1/2]

```
void Room::setAudioSource (
    AudioSource audio ) [inline]
```

## 8.13.3.8 setAudioSource() [2/2]

```
void Room::setAudioSource (
    AudioSource audio ) [inline]
```

## 8.13.3.9 setCameras() [1/2]

```
void Room::setCameras (
    vector< Camera > cams ) [inline]
```

## 8.13.3.10 setCameras() [2/2]

```
void Room::setCameras (
    vector< Camera > cams ) [inline]
```

## 8.13.3.11 setName() [1/2]

```
void Room::setName (
    string name ) [inline]
```

#### 8.13.3.12 setName() [2/2]

```
void Room::setName (  
    string name ) [inline]
```

### 8.13.4 Данные класса

#### 8.13.4.1 audio

```
AudioSource Room::audio [private]
```

#### 8.13.4.2 cameras

```
vector< Camera > Room::cameras [private]
```

#### 8.13.4.3 name

```
string Room::name [private]
```

#### 8.13.4.4 type

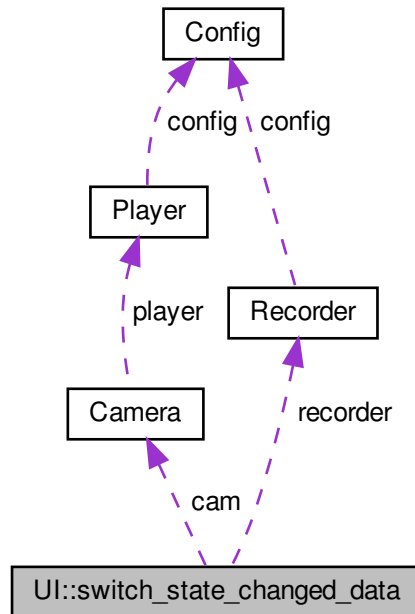
```
room\_t Room::type
```

Объявления и описания членов класса находятся в файле:

- /home/jetson/controlroom/Studio/Operator/Grid/include/[room.h](#)

## 8.14 Структура UI::switch\_state\_changed\_data

Граф связей класса UI::switch\_state\_changed\_data:



Открытые атрибуты

- [Camera](#) \* [cam](#)
- [Recorder](#) \* [recorder](#)

### 8.14.1 Данные класса

#### 8.14.1.1 cam

[Camera](#)\* UI::switch\_state\_changed\_data::cam

#### 8.14.1.2 recorder

[Recorder](#)\* UI::switch\_state\_changed\_data::recorder

Объявления и описания членов структуры находятся в файле:

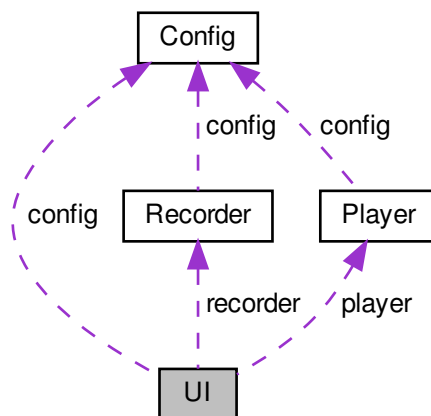
- `/home/jetson/controlroom/Studio/Recorder/include/ui.h`

## 8.15 Класс UI

The `UI` class.

```
#include <ui.h>
```

Граф связей класса `UI`:



### Классы

- struct `display_player_data`
- struct `gdrive_status_data`
- struct `switch_state_changed_data`

### Открытые члены

- `UI (Config *config)`
- `~UI ()`
- `UI (Config *config)`
- `~UI ()`
- void `displayRecordingStatus` (string cam\_id, bool status)

### Открытые атрибуты

- `Config * config`
- string `playingCamName` = ""
- `Recorder * recorder`
- vector< GtkWidget \* > `switchGridV`

## Закрытые члены

- int `initStyles` ()
- GtkWidget \* `windowInit` (GtkBuilder \*\*builder, string gladeFile, string windowName)
- void `initMenuWidgets` ()
- void `initCamWidgets` (int room\_n, vector< Camera > \*cams)
- void `initRoomTab` (int room\_n, string room\_name)
- string `findIP` ()
- int `initStyles` ()
- GtkWidget \* `windowInit` (GtkBuilder \*\*builder, string gladeFile, string windowName)
- void `initPlayerWidgets` ()
- void `initMenuWidgets` ()
- void `initCamWidgets` (int room\_n, vector< Camera > \*cams)
- void `initRoomTab` (int room\_n, string room\_name)
- string `findIP` ()

## Закрытые статические члены

- static int `on_show` (gpointer ui\_ptr)
- static void `pageSwitched` (GtkWidget \*widget, GParamSpec \*property, vector< Room \*> \*rooms)
- static void `displayPlayer` (GtkWidget \*widget, gpointer data)
- static gboolean `keyPress` (GtkWidget \*widget, GdkEventKey \*event, UI \*ui)
- static gboolean `updateGDriveStatus` (gpointer user\_data)
- static void `displayPlayer` (GtkWidget \*widget, gpointer data)
- static gboolean `keyPress` (GtkWidget \*widget, GdkEventKey \*event, UI \*ui)
- static void `switchStateChanged` (GtkWidget \*widget, gboolean state, gpointer user\_data)
- static void `editButtonClicked` (GtkWidget \*widget, vector< GtkWidget \*> \*switchGridV)

## Закрытые данные

- vector< Room \* > \* `rooms`
- map< Camera \*, Player \* > `players`
- GtkBuilder \* `menuBuilder`
- GtkWidget \* `menuWindow`
- Player \* `player`
- GtkBuilder \* `playerBuilder`
- GtkWidget \* `playerWindow`
- GtkWidget \* `playerWidget`
- GtkWidget \* `playerLabel`
- GtkWidget \* `editButton`

## 8.15.1 Подробное описание

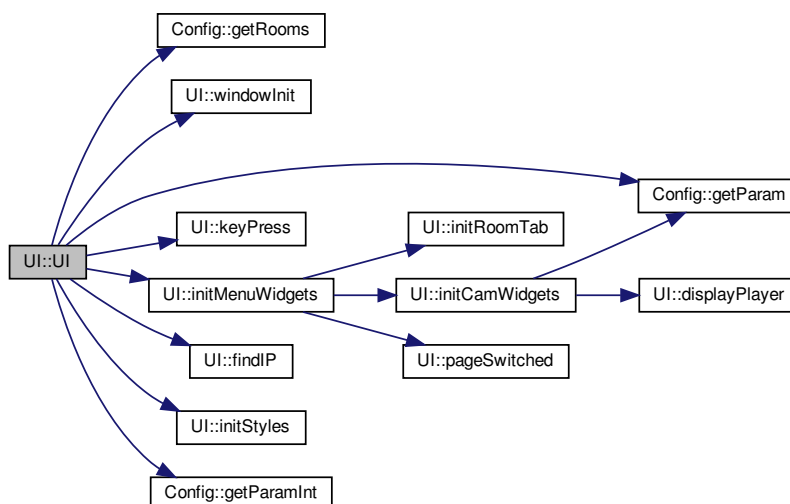
The `UI` class.

## 8.15.2 Конструктор(ы)

## 8.15.2.1 UI() [1/2]

UI::UI (   
Config \* config )

Граф вызовов:



## 8.15.2.2 ~UI() [1/2]

UI::~~UI ( )

## 8.15.2.3 UI() [2/2]

UI::UI (   
Config \* config )

## 8.15.2.4 ~UI() [2/2]

UI::~~UI ( )

## 8.15.3 Методы



## 8.15.3.1 displayPlayer() [1/2]

```
void UI::displayPlayer (
    GtkWidget * widget,
    gpointer data ) [static], [private]
```

## 8.15.3.2 displayPlayer() [2/2]

```
static void UI::displayPlayer (
    GtkWidget * widget,
    gpointer data ) [static], [private]
```

## 8.15.3.3 displayRecordingStatus()

```
void UI::displayRecordingStatus (
    string cam_id,
    bool status )
```

## 8.15.3.4 editButtonClicked()

```
void UI::editButtonClicked (
    GtkWidget * widget,
    vector< GtkWidget *> * switchGridV ) [static], [private]
```

Граф вызовов:



## 8.15.3.5 findIP() [1/2]

```
string UI::findIP ( ) [private]
```

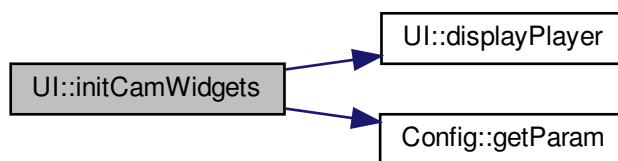
## 8.15.3.6 findIP() [2/2]

```
string UI::findIP ( ) [private]
```

## 8.15.3.7 initCamWidgets() [1/2]

```
void UI::initCamWidgets (
    int room_n,
    vector< Camera > * cams ) [private]
```

Граф вызовов:



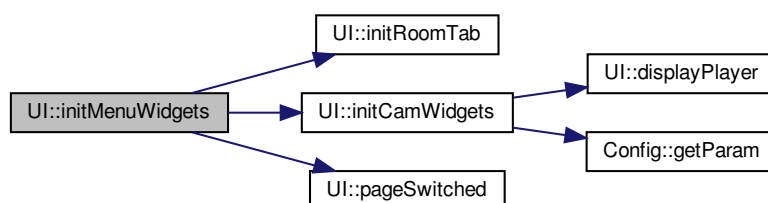
## 8.15.3.8 initCamWidgets() [2/2]

```
void UI::initCamWidgets (
    int room_n,
    vector< Camera > * cams ) [private]
```

## 8.15.3.9 initMenuWidgets() [1/2]

```
void UI::initMenuWidgets ( ) [private]
```

Граф вызовов:



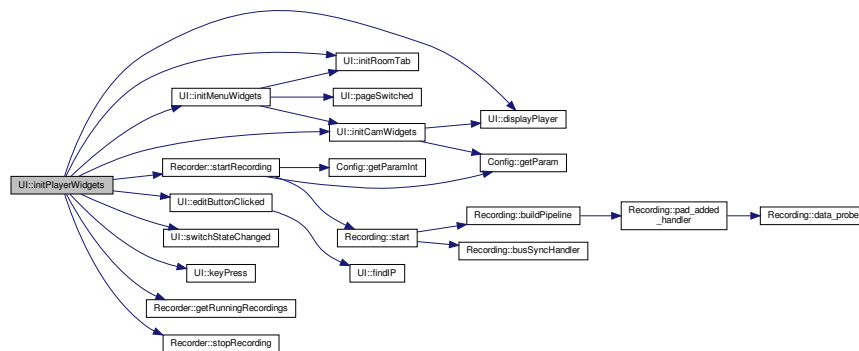
## 8.15.3.10 initMenuWidgets() [2/2]

```
void UI::initMenuWidgets ( ) [private]
```

## 8.15.3.11 initPlayerWidgets()

```
void UI::initPlayerWidgets ( ) [private]
```

Граф вызовов:



## 8.15.3.12 initRoomTab() [1/2]

```
void UI::initRoomTab (
    int room_n,
    string room_name ) [private]
```

## 8.15.3.13 initRoomTab() [2/2]

```
void UI::initRoomTab (
    int room_n,
    string room_name ) [private]
```

## 8.15.3.14 initStyles() [1/2]

```
int UI::initStyles ( ) [private]
```

## 8.15.3.15 initStyles() [2/2]

```
int UI::initStyles ( ) [private]
```

## 8.15.3.16 keyPress() [1/2]

```
gboolean UI::keyPress (
    GtkWidget * widget,
    GdkEventKey * event,
    UI * ui ) [static], [private]
```

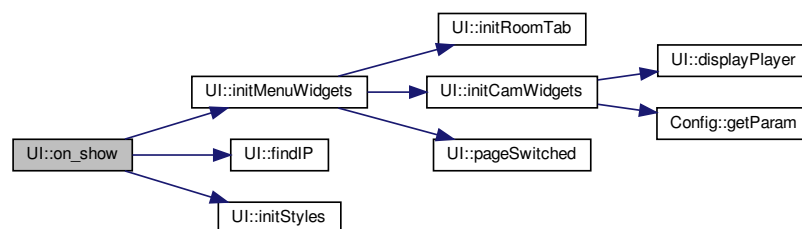
## 8.15.3.17 keyPress() [2/2]

```
static gboolean UI::keyPress (
    GtkWidget * widget,
    GdkEventKey * event,
    UI * ui ) [static], [private]
```

## 8.15.3.18 on\_show()

```
int UI::on_show (
    gpointer ui_ptr ) [static], [private]
```

Граф вызовов:



## 8.15.3.19 pageSwitched()

```
void UI::pageSwitched (
    GtkWidget * widget,
    GParamSpec * property,
    vector< Room *> * rooms ) [static], [private]
```

## 8.15.3.20 switchStateChanged()

```
void UI::switchStateChanged (
    GtkWidget * widget,
    gboolean state,
    gpointer user_data ) [static], [private]
```

## 8.15.3.21 updateGDriveStatus()

```
gboolean UI::updateGDriveStatus (
    gpointer user_data ) [static], [private]
```

Граф вызовов:



## 8.15.3.22 windowInit() [1/2]

```
GtkWidget * UI::windowInit (
    GtkBuilder ** builder,
    string gladeFile,
    string windowName ) [private]
```

## 8.15.3.23 windowInit() [2/2]

```
GtkWidget* UI::windowInit (
    GtkBuilder ** builder,
    string gladeFile,
    string windowName ) [private]
```

## 8.15.4 Данные класса

## 8.15.4.1 config

`Config * UI::config`

## 8.15.4.2 editButton

`GtkWidget* UI::editButton` [private]

## 8.15.4.3 menuBuilder

`GtkBuilder * UI::menuBuilder` [private]

## 8.15.4.4 menuWindow

`GtkWidget * UI::menuWindow` [private]

## 8.15.4.5 player

`Player* UI::player` [private]

## 8.15.4.6 playerBuilder

`GtkBuilder * UI::playerBuilder` [private]

## 8.15.4.7 playerLabel

`GtkWidget * UI::playerLabel` [private]

## 8.15.4.8 players

`map<Camera*, Player*> UI::players` [private]

## 8.15.4.9 playerWidget

```
GtkWidget* UI::playerWidget [private]
```

## 8.15.4.10 playerWindow

```
GtkWidget * UI::playerWindow [private]
```

## 8.15.4.11 playingCamName

```
string UI::playingCamName = ""
```

## 8.15.4.12 recorder

```
Recorder* UI::recorder
```

## 8.15.4.13 rooms

```
vector< Room * > * UI::rooms [private]
```

## 8.15.4.14 switchGridV

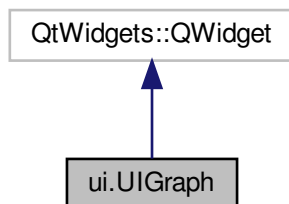
```
vector<GtkWidget*> UI::switchGridV
```

Объявления и описания членов классов находятся в файлах:

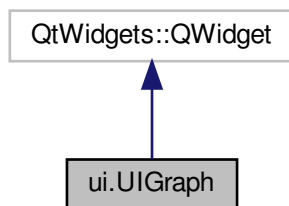
- /home/jetson/controlroom/Studio/Operator/Grid/include/[ui.h](#)
- /home/jetson/controlroom/Studio/Operator/Grid/src/[ui.cpp](#)

## 8.16 Класс `ui.UIGraph`

Граф наследования: `ui.UIGraph`:



Граф связей класса `ui.UIGraph`:



Открытые члены

- `def __init__(self)`

### 8.16.1 Конструктор(ы)

#### 8.16.1.1 `__init__()`

```
def ui.UIGraph.__init__(  
    self )
```

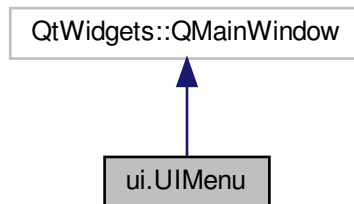
Объявления и описания членов класса находятся в файле:

- `/home/jetson/controlroom/Studio/Network/src/ui.py`

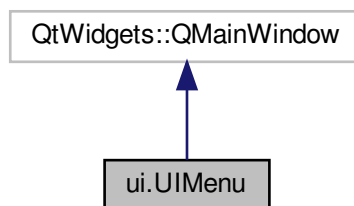


## 8.17 Класс `ui.UIMenu`

Граф наследования: `ui.UIMenu`:



Граф связей класса `ui.UIMenu`:



Открытые члены

- `def __init__(self)`

### 8.17.1 Конструктор(ы)

#### 8.17.1.1 `__init__()`

```
def ui.UIMenu.__init__(
    self )
```

Объявления и описания членов класса находятся в файле:

- `/home/jetson/controlroom/Studio/Network/src/ui.py`



## Глава 9

# Файлы

### 9.1 Файл groups.dox

### 9.2 Файл /home/jetson/controlroom/Studio/Network/src/ui.py

#### Классы

- class [ui.UIMenu](#)
- class [ui.UIGraph](#)
- class [ui.Pinger](#)
  - Пингует камеры
- class [ui.Grapher](#)
- class [ui.Camera](#)

#### Пространства имен

- [ui](#)

#### Функции

- def [ui.runPing](#) ()

#### Переменные

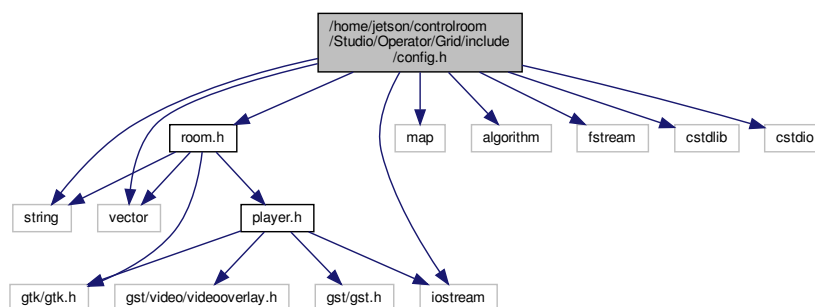
- dictionary [ui.addresses](#)
- [ui.app](#) = QtWidgets.QApplication([])
- [ui.menu](#) = UIMenu()
- [ui.stylesheet](#) = open("../ui/stylesheet.css", "r")
- [ui.display0](#) = QtWidgets.QDesktopWidget().screenGeometry(0)
- [ui.graph](#) = UIGraph()
- [ui.display1](#) = QtWidgets.QDesktopWidget().screenGeometry(1)
- [ui.widget](#) = graph.findChild(pg.PlotWidget, "plotwidget")
- [ui.plot](#) = widget.getPlotItem()
- [ui.rightAxis](#) = plot.getAxis('right')

- `ui.leftAxis = plot.getAxis('left')`
- dictionary `ui.labelStyle = {'color': '#FFF', 'font-size': '20pt'}`
- `ui.units`
- `ui.font = QFont()`
- `ui.tickFont`
- `ui.x`
- `ui.True`
- `ui.y`
- `ui.alpha`
- `ui.grapher = Grapher(plot)`
- list `ui.cams = []`
- int `ui.i = 0`
- `ui.button = menu.findChild(QtWidgets.QPushButton, "b" + str(i))`
- `ui.label = menu.findChild(QtWidgets.QLabel, "l" + str(i))`
- `ui.camera = Camera(adresses[c], button, label, grapher)`
- `ui.threadpool = QThreadPool()`
- `ui.pingTimer = QTimer()`
- `ui.pinger = Pinger(cams)`

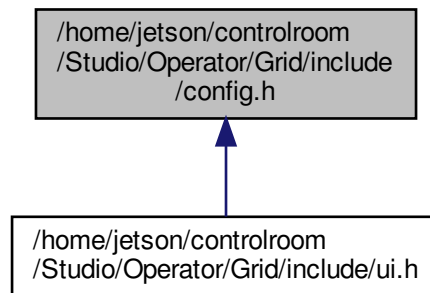
### 9.3 Файл `/home/jetson/controlroom/Studio/Operator/Grid/include/config.h`

```
#include <room.h>
#include <vector>
#include <map>
#include <string>
#include <algorithm>
#include <fstream>
#include <iostream>
#include <cstdlib>
#include <cstdio>
```

Граф включаемых заголовочных файлов для `config.h`:



Граф файлов, в которые включается этот файл:



## Классы

- class [Config](#)

Класс для хранения конфигурации

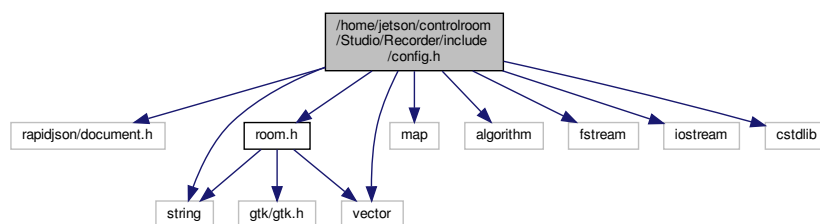
## 9.4 Файл /home/jetson/controlroom/Studio/Recorder/include/config.h

```

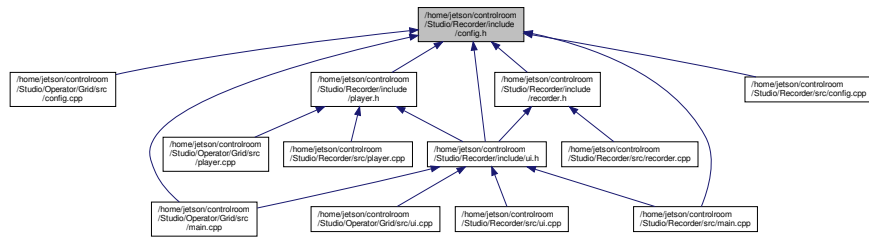
#include "rapidjson/document.h"
#include "room.h"
#include <vector>
#include <map>
#include <string>
#include <algorithm>
#include <fstream>
#include <iostream>
#include <cstdlib>

```

Граф включаемых заголовочных файлов для config.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Config](#)

Класс для хранения конфигурации

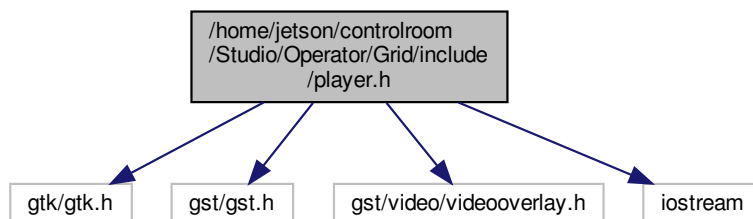
## 9.5 Файл /home/jetson/controlroom/Studio/Operator/Grid/include/player.h

```

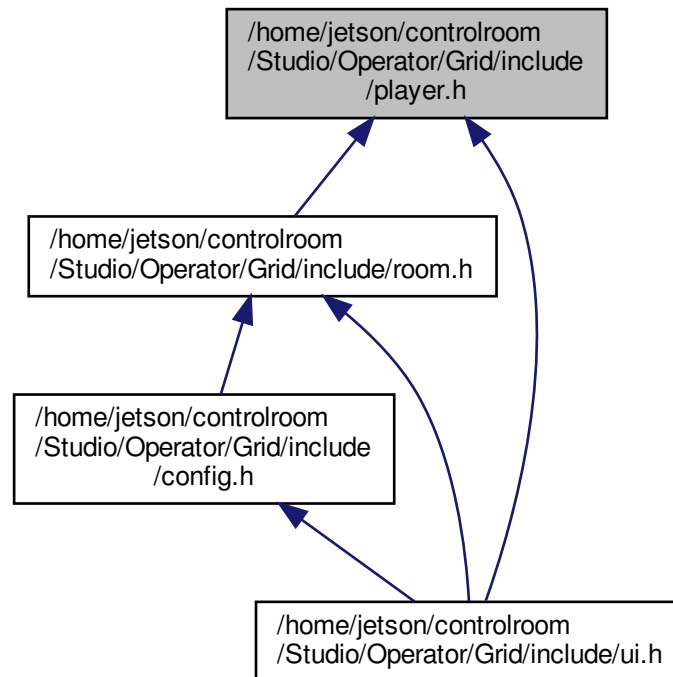
#include <gtk/gtk.h>
#include <gst/gst.h>
#include <gst/video/videooverlay.h>
#include <iostream>

```

Граф включаемых заголовочных файлов для player.h:



Граф файлов, в которые включается этот файл:



## Классы

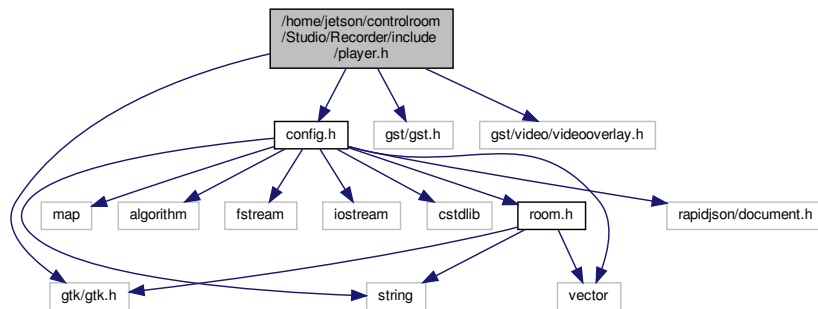
- struct [PadData](#)
- class [Player](#)

The [Player](#) class.

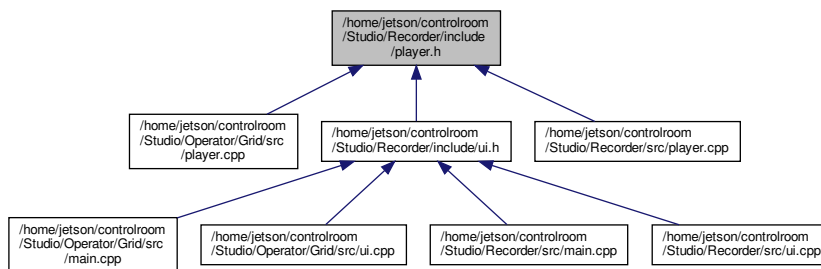
## 9.6 Файл /home/jetson/controlroom/Studio/Recorder/include/player.h

```
#include <gtk/gtk.h>
#include <gst/gst.h>
#include <gst/video/videooverlay.h>
#include "config.h"
```

Граф включаемых заголовочных файлов для `player.h`:



Граф файлов, в которые включается этот файл:



## Классы

- struct `PadData`
- class `Player`

The `Player` class.

## 9.7 Файл `/home/jetson/controlroom/Studio/Operator/Grid/include/room.h`

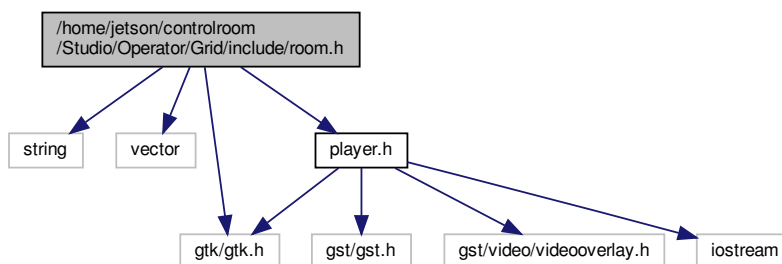
```

#include <string>
#include <vector>
#include <gtk/gtk.h>
#include <player.h>

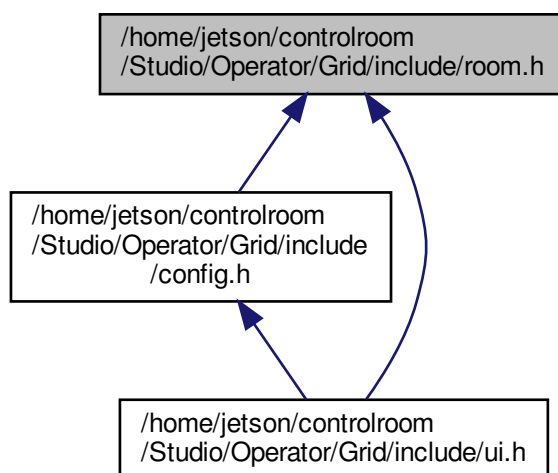
```



Граф включаемых заголовочных файлов для room.h:



Граф файлов, в которые включается этот файл:



## Классы

- struct [Camera](#)
- struct [AudioSource](#)
- class [Room](#)

The [Room](#) class.

## Перечисления

- enum [room\\_t](#) { [CUSTOM](#), [GSUITE](#), [CUSTOM](#), [GSUITE](#) }

## 9.7.1 Перечисления

### 9.7.1.1 room\_t

enum `room_t`

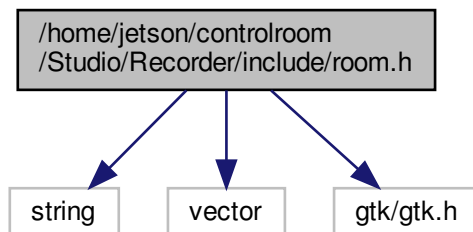
Элементы перечислений

CUSTOM	
GSUITE	
CUSTOM	
GSUITE	

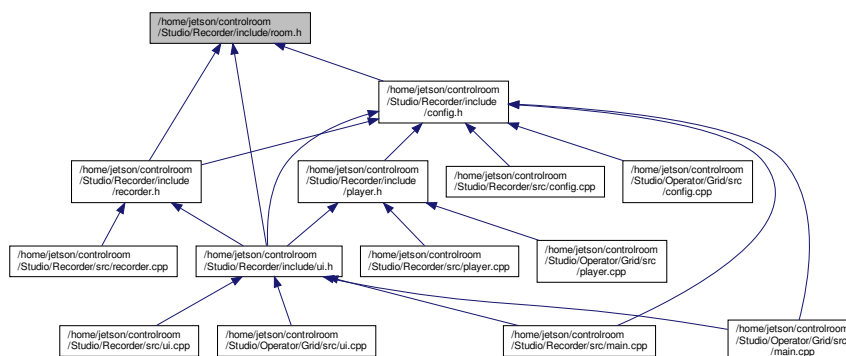
## 9.8 Файл `/home/jetson/controlroom/Studio/Recorder/include/room.h`

```
#include <string>
#include <vector>
#include <gtk/gtk.h>
```

Граф включаемых заголовочных файлов для `room.h`:



Граф файлов, в которые включается этот файл:



## Классы

- struct Camera
- struct AudioSource
- class Room

The `Room` class.

## Перечисления

- enum room\_t { CUSTOM, GSUITE, CUSTOM, GSUITE }

### 9.8.1 Перечисления

#### 9.8.1.1 room t

```
enum room_t
```

## Элементы перечислений

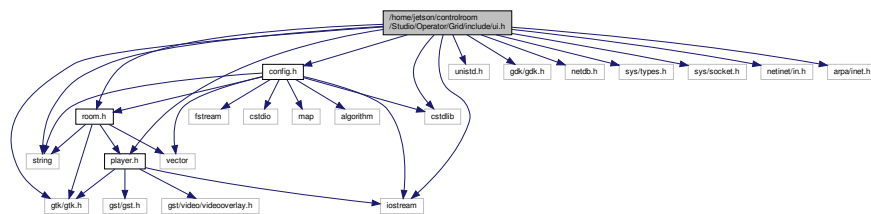
CUSTOM	
GSUITE	
CUSTOM	
GSUITE	

9.9 Файл /home/jetson/controlroom/Studio/Operator/Grid/include/ui.h

```
#include <iostream>
#include <string>
```

```
#include <unistd.h>
#include <gtk/gtk.h>
#include <gdk/gdk.h>
#include "room.h"
#include "config.h"
#include "player.h"
#include <cstdlib>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

Граф включаемых заголовочных файлов для ui.h:



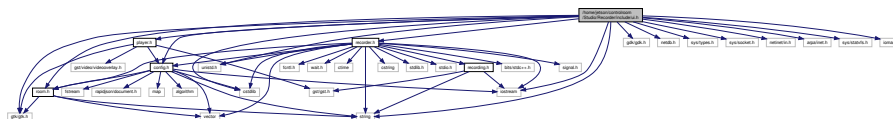
## Классы

- class **UI**  
The **UI** class.
- struct **UI::display\_player\_data**

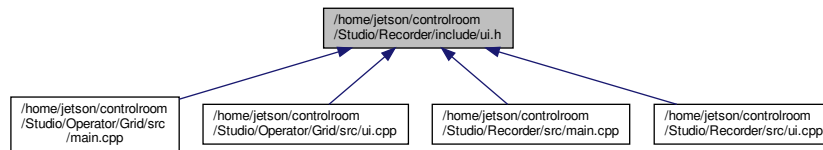
## 9.10 Файл /home/jetson/controlroom/Studio/Recorder/include/ui.h

```
#include <iostream>
#include <string>
#include <unistd.h>
#include <gtk/gtk.h>
#include <gdk/gdk.h>
#include "room.h"
#include "config.h"
#include "player.h"
#include "recorder.h"
#include <cstdlib>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/statvfs.h>
#include <iomanip>
```

Граф включаемых заголовочных файлов для ui.h:



Граф файлов, в которые включается этот файл:



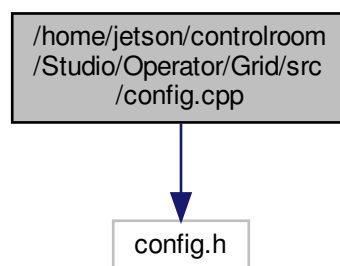
## Классы

- class `UI`  
The `UI` class.
- struct `UI::gdrive_status_data`
- struct `UI::display_player_data`
- struct `UI::switch_state_changed_data`

## 9.11 Файл /home/jetson/controlroom/Studio/Operator/Grid/src/config.cpp

```
#include "config.h"
```

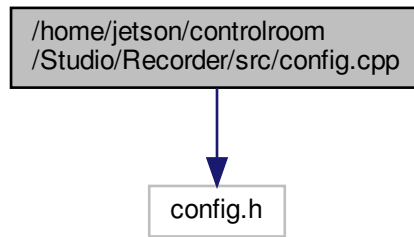
Граф включаемых заголовочных файлов для config.cpp:



## 9.12 Файл /home/jetson/controlroom/Studio/Recorder/src/config.cpp

```
#include "config.h"
```

Граф включаемых заголовочных файлов для config.cpp:



### 9.13 Файл `/home/jetson/controlroom/Studio/Operator/Grid/src/from-gsuite.py`

Пространства имен

- `from-gsuite`

Функции

- `def from-gsuite.main ()`

Переменные

- `list from-gsuite.SCOPEs = ['https://www.googleapis.com/auth/admin.directory.resource.calendar']`

### 9.14 Файл `/home/jetson/controlroom/Studio/Recorder/src/from-gsuite.py`

Пространства имен

- `from-gsuite`

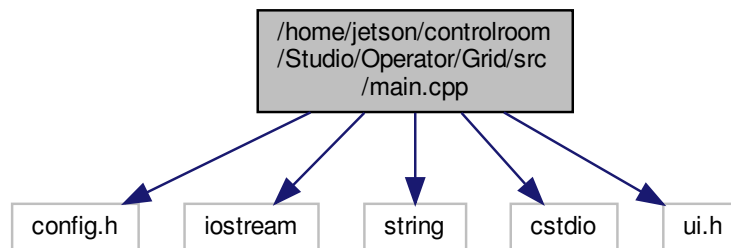
Функции

- `def from-gsuite.main ()`

## 9.15 Файл /home/jetson/controlroom/Studio/Operator/Grid/src/main.cpp

```
#include "config.h"  
#include <iostream>  
#include <string>  
#include <cstdio>  
#include "ui.h"
```

Граф включаемых заголовочных файлов для main.cpp:



## Функции

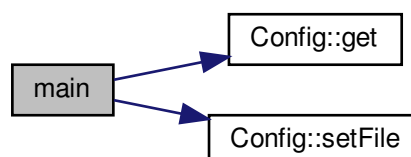
- int `main` ()

## 9.15.1 Функции

9.15.1.1 `main()`

```
int main ( )
```

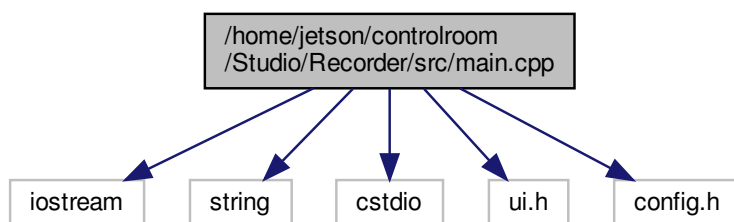
Граф вызовов:



## 9.16 Файл /home/jetson/controlroom/Studio/Recorder/src/main.cpp

```
#include <iostream>
#include <string>
#include <cstdio>
#include "ui.h"
#include "config.h"
```

Граф включаемых заголовочных файлов для main.cpp:



### Функции

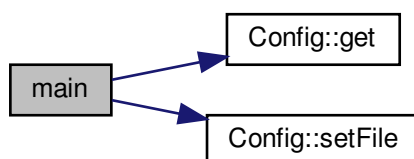
- int `main` ()

#### 9.16.1 Функции

##### 9.16.1.1 main()

```
int main ( )
```

Граф вызовов:

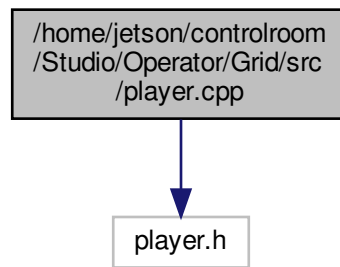




## 9.17 Файл /home/jetson/controlroom/Studio/Operator/Grid/src/player.cpp

```
#include "player.h"
```

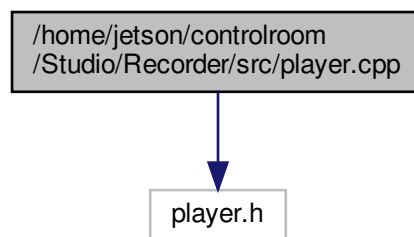
Граф включаемых заголовочных файлов для player.cpp:



## 9.18 Файл /home/jetson/controlroom/Studio/Recorder/src/player.cpp

```
#include "player.h"
```

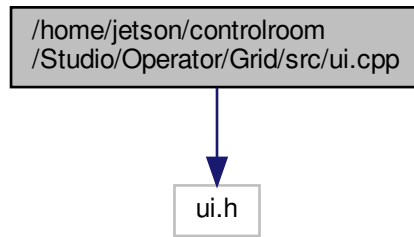
Граф включаемых заголовочных файлов для player.cpp:



## 9.19 Файл /home/jetson/controlroom/Studio/Operator/Grid/src/ui.cpp

```
#include "ui.h"
```

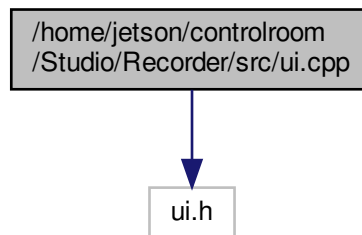
Граф включаемых заголовочных файлов для ui.cpp:



## 9.20 Файл `/home/jetson/controlroom/Studio/Recorder/src/ui.cpp`

```
#include "ui.h"
```

Граф включаемых заголовочных файлов для ui.cpp:

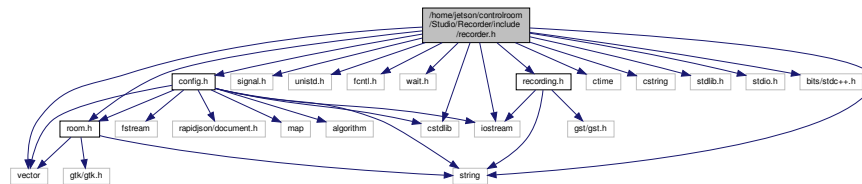


## 9.21 Файл `/home/jetson/controlroom/Studio/Recorder/include/recorder.h`

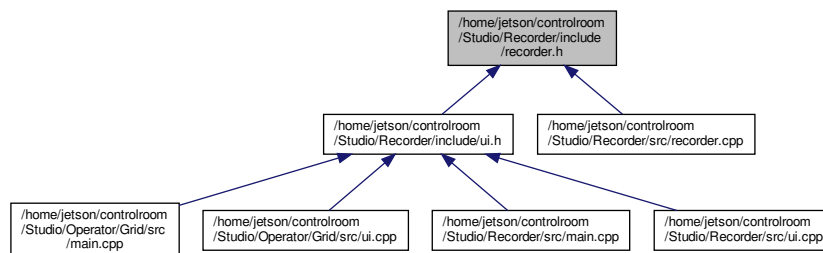
```
#include <iostream>
#include <vector>
#include <string>
#include <signal.h>
#include <unistd.h>
#include <fcntl.h>
#include <wait.h>
#include "config.h"
#include <ctime>
#include <cstring>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <cstdlib>
#include <bits/stdc++.h>
#include "recording.h"
#include "room.h"
```

Граф включаемых заголовочных файлов для recorder.h:



Граф файлов, в которые включается этот файл:



## Классы

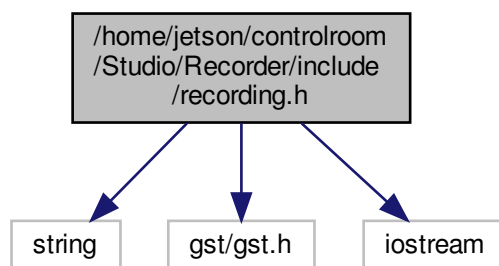
- class [Recorder](#)

Управляет процессом записи

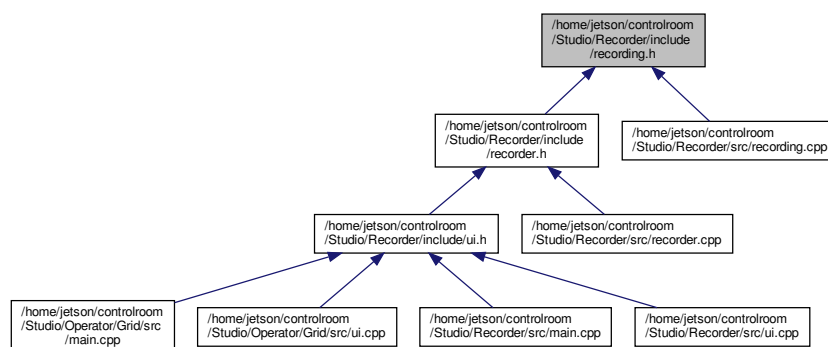
## 9.22 Файл /home/jetson/controlroom/Studio/Recorder/include/recording.h

```
#include <string>
#include <gst/gst.h>
#include <iostream>
```

Граф включаемых заголовочных файлов для `recording.h`:



Граф файлов, в которые включается этот файл:



## Классы

- class `Recording`  
The `Recording` class.

## Перечисления

- enum `status_t` { `RUNNING`, `RELINKING`, `STOPPING`, `STOPPED` }

### 9.22.1 Перечисления

#### 9.22.1.1 `status_t`

```
enum status_t
```

Элементы перечислений

RUNNING	
RELINKING	
STOPPING	
STOPPED	

## 9.23 Файл /home/jetson/controlroom/Studio/Recorder/src/gpio.py

Пространства имен

- `gpio`

Функции

- `def gpio.keypress (key)`
- `def gpio.stop_cb (channel=0)`
- `def gpio.rec_cb (channel=0)`
- `def gpio.menu_cb (channel=0)`
- `def gpio.main ()`

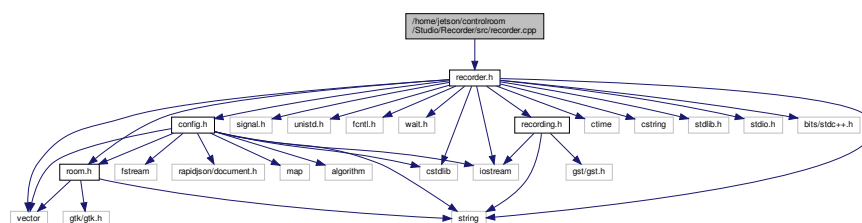
Переменные

- `int gpio.stop = 12`
- `int gpio.rec = 11`
- `int gpio.menu = 18`
- `string gpio.key_r = "key R "`
- `string gpio.key_s = "key S "`
- `string gpio.key_m = "key M "`

## 9.24 Файл /home/jetson/controlroom/Studio/Recorder/src/recorder.cpp

```
#include "recorder.h"
```

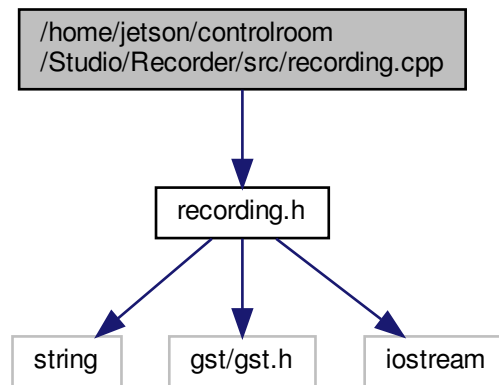
Граф включаемых заголовочных файлов для recorder.cpp:



## 9.25 Файл /home/jetson/controlroom/Studio/Recorder/src/recording.cpp

```
#include "recording.h"
```

Граф включаемых заголовочных файлов для recording.cpp:



## 9.26 Файл /home/jetson/controlroom/Studio/Recorder/src/video-upload.py

Пространства имен

- `video-upload`

Переменные

- `video-upload.parser` = `argparse.ArgumentParser("Videofile name and location")`
- `video-upload.help`
- `video-upload.type`
- `video-upload.args` = `parser.parse_args()`
- `string video-upload.SCOPEES` = `'https://www.googleapis.com/auth/drive.file'`
- `video-upload.store` = `file.Storage('storage.json')`
- `video-upload.creds` = `store.get()`
- `video-upload.flow` = `client.flow_from_clientsecrets('client_secret.json', scope = SCOPEES)`
- `video-upload.DRIVE` = `build('drive', 'v3', http = creds.authorize(Http()))`
- `video-upload.filename` = `args.location + args.filename`
- dictionary `video-upload.metadata` = `{'name': args.filename}`
- `video-upload.res` = `DRIVE.files().create(body = metadata, media_body = filename).execute()`

# Предметный указатель

/home/jetson/controlroom/Studio/Network/src/ui.↵ upload.py, 102  
py, 83  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/include/config.h, 84  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/include/player.h, 86  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/include/room.h, 88  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/include/ui.h, 91  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/src/config.cpp, 93  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/src/from-gsuite.py, 94  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/src/main.cpp, 95  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/src/player.cpp, 97  
/home/jetson/controlroom/Studio/Operator/↵  
Grid/src/ui.cpp, 97  
/home/jetson/controlroom/Studio/Recorder/include/config.↵  
h, 85  
/home/jetson/controlroom/Studio/Recorder/include/player.↵  
h, 87  
/home/jetson/controlroom/Studio/Recorder/include/recon.↵  
h, 98  
/home/jetson/controlroom/Studio/Recorder/include/recon.↵  
h, 99  
/home/jetson/controlroom/Studio/Recorder/include/roomi.↵  
h, 90  
/home/jetson/controlroom/Studio/Recorder/include/ui.↵video-upload, 24  
h, 92  
/home/jetson/controlroom/Studio/Recorder/src/config.↵Room, 68  
cpp, 93  
/home/jetson/controlroom/Studio/Recorder/src/from- name, 27  
gsuite.py, 94 uri, 27  
/home/jetson/controlroom/Studio/Recorder/src/gpio.↵  
py, 101  
/home/jetson/controlroom/Studio/Recorder/src/main.↵ Player, 47  
cpp, 96 Recording, 57  
/home/jetson/controlroom/Studio/Recorder/src/player.↵  
cpp, 97  
/home/jetson/controlroom/Studio/Recorder/src/recon.↵  
cpp, 101  
/home/jetson/controlroom/Studio/Recorder/src/recording.↵  
cpp, 102  
/home/jetson/controlroom/Studio/Recorder/src/ui.↵ Camera, 30  
cpp, 98 ui, 20  
/home/jetson/controlroom/Studio/Recorder/src/video- ui::Camera, 29  
-- init --  
ui::Camera, 28  
ui::Grapher, 40  
ui::Pinger, 43  
ui::UIGraph, 80  
ui::UIMenu, 81  
\_runCmd  
ui::Pinger, 44  
Рекордер, 11  
Сетевой монитор, 13  
Сетка видеопотоков, 12  
~Player  
Player, 46, 47  
~Recorder  
Recorder, 53  
~Recording  
Recording, 57  
~UI  
UI, 72  
activeCam  
ui::Grapher, 41  
adresses  
alpha  
audio  
app  
args  
audio-video-upload, 24  
AudioSource, 27  
name, 27  
uri, 27  
buildPipeline  
Player, 47  
Recording, 57  
Player, 50  
Handler  
Player, 48  
Recording, 58  
button  
Camera, 30  
ui, 20  
ui::Camera, 29

- cam
  - UI::display\_player\_data, [38](#)
  - UI::switch\_state\_changed\_data, [69](#)
- camList
  - ui::Pinger, [44](#)
- camName
  - Player, [50](#)
  - Recording, [61](#)
- Camera, [30](#)
  - button, [30](#)
  - drawingArea, [30](#)
  - fullName, [30](#)
  - name, [31](#)
  - player, [31](#)
  - recImage, [31](#)
  - record, [31](#)
  - uri, [31](#)
- camera
  - ui, [20](#)
- cameras
  - Room, [68](#)
- cams
  - ui, [20](#)
- changeCam
  - ui::Grapher, [40](#)
- checkIfRecStopped
  - Recorder, [54](#)
- clock
  - Recording, [61](#)
- Config, [31](#)
  - Config, [33](#)
  - configuration, [37](#)
  - get, [34](#)
  - getCustomRooms, [34](#)
  - getGSuiteRooms, [34](#), [35](#)
  - getParam, [35](#)
  - getParamInt, [35](#)
  - getRooms, [35](#), [36](#)
  - makeGSuiteRequest, [36](#)
  - operator=, [36](#)
  - readRoomsFromFile, [36](#)
  - rooms, [37](#)
  - setFile, [37](#)
- config
  - Player, [50](#)
  - Recorder, [55](#)
  - UI, [77](#)
- configuration
  - Config, [37](#)
- creds
  - video-upload, [24](#)
- curve
  - ui::Grapher, [41](#)
- DRIVE
  - video-upload, [24](#)
- data
  - ui::Grapher, [41](#)
- data\_probe
  - Recording, [58](#)
- dec
  - Player, [51](#)
- depay
  - PadData, [42](#)
  - Player, [51](#)
  - Recording, [61](#)
- display0
  - ui, [20](#)
- display1
  - ui, [21](#)
- displayPlayer
  - UI, [72](#), [73](#)
- displayRecordingStatus
  - UI, [73](#)
- drawMe
  - ui::Camera, [28](#)
- drawingArea
  - Camera, [30](#)
- editButton
  - UI, [78](#)
- editButtonClicked
  - UI, [73](#)
- elementSinkLinked
  - Recording, [58](#)
- elementSrcLinked
  - Recording, [58](#)
- enc
  - Recording, [62](#)
- fakesink
  - Recording, [62](#)
- fileName
  - Recording, [62](#)
- filename
  - video-upload, [24](#)
- findIP
  - UI, [73](#)
- flow
  - video-upload, [24](#)
- folder
  - Recording, [62](#)
- font
  - ui, [21](#)
- freeze\_check
  - Recording, [58](#)
- freeze\_check\_id
  - Recording, [62](#)
- from-gsuite, [15](#)
  - main, [15](#)
  - SCOPES, [15](#)
- fullName
  - Camera, [30](#)
- GDriveIcon
  - UI::gdrive\_status\_data, [39](#)
- get
  - Config, [34](#)



- getAudioSource
  - Room, [66](#)
- getCameras
  - Room, [66](#)
- getCustomRooms
  - Config, [34](#)
- getFileName
  - Recording, [59](#)
- getGSuiteRooms
  - Config, [34](#), [35](#)
- getName
  - Room, [66](#), [67](#)
- getParam
  - Config, [35](#)
- getParamInt
  - Config, [35](#)
- getRooms
  - Config, [35](#), [36](#)
- getRunningRecordings
  - Recorder, [54](#)
- getStatus
  - Recording, [59](#)
- gpio, [16](#)
  - key\_m, [18](#)
  - key\_r, [18](#)
  - key\_s, [18](#)
  - keypress, [16](#)
  - main, [16](#)
  - menu, [18](#)
  - menu\_cb, [16](#)
  - rec, [18](#)
  - rec\_cb, [17](#)
  - stop, [18](#)
  - stop\_cb, [17](#)
- graph
  - ui, [21](#)
- grapher
  - ui, [21](#)
  - ui::Camera, [29](#)
- groups.dox, [83](#)
- help
  - video-upload, [24](#)
- i
  - ui, [21](#)
- in\_idle\_probe
  - Recording, [62](#)
- initCamWidgets
  - UI, [74](#)
- initMenuWidgets
  - UI, [74](#)
- initPlayerWidgets
  - UI, [75](#)
- initRoomTab
  - UI, [75](#)
- initStyles
  - UI, [75](#)
- ip
  - ui::Camera, [29](#)
- isGDriveUploadActive
  - Recorder, [54](#)
- isPlaying
  - Player, [48](#)
- key\_m
  - gpio, [18](#)
- key\_r
  - gpio, [18](#)
- key\_s
  - gpio, [18](#)
- keyPress
  - UI, [76](#)
- keypress
  - gpio, [16](#)
- label
  - ui, [21](#)
  - ui::Camera, [29](#)
- labelStyle
  - ui, [21](#)
- lastBufferTime
  - Recording, [62](#)
- leftAxis
  - ui, [21](#)
- main
  - from-gsuite, [15](#)
  - gpio, [16](#)
  - Operator/Grid/src/main.cpp, [95](#)
  - Recorder/src/main.cpp, [96](#)
- makeGSuiteRequest
  - Config, [36](#)
- menu
  - gpio, [18](#)
  - ui, [22](#)
- menu\_cb
  - gpio, [16](#)
- menuBuilder
  - UI, [78](#)
- menuWindow
  - UI, [78](#)
- metadata
  - video-upload, [24](#)
- mux
  - Recording, [62](#)
- name
  - AudioSource, [27](#)
  - Camera, [31](#)
  - Room, [68](#)
- on\_show
  - UI, [76](#)
- Operator/Grid/include/room.h
  - room\_t, [90](#)
- Operator/Grid/src/main.cpp
  - main, [95](#)

- operator=
  - Config, 36
- pad\_added\_handler
  - Player, 48
  - Recording, 59
- PadData, 42
  - depay, 42
  - src, 42
- pageSwitched
  - UI, 76
- parse
  - Player, 51
  - Recording, 63
- parser
  - video-upload, 24
- pingHistory
  - ui::Camera, 29
- pingTimer
  - ui, 22
- pinger
  - ui, 22
- pipeline
  - Player, 51
  - Recording, 63
- platform
  - Player, 51
- playStream
  - Player, 49
- Player, 45
  - ~Player, 46, 47
  - buildPipeline, 47
  - bus, 50
  - busSyncHandler, 48
  - camName, 50
  - config, 50
  - dec, 51
  - depay, 51
  - isPlaying, 48
  - pad\_added\_handler, 48
  - parse, 51
  - pipeline, 51
  - platform, 51
  - playStream, 49
  - Player, 46, 47
  - playing, 51
  - scale, 51
  - sink, 51
  - src, 52
  - stopStream, 49
  - uri, 52
  - videoWidgetDraw\_cb, 49, 50
  - videoWidgetRealize\_cb, 50
  - videoWindow, 52
  - videoWindowHandle, 52
- player
  - Camera, 31
  - UI::display\_player\_data, 38
  - UI, 78
- playerBuilder
  - UI, 78
- playerLabel
  - UI::display\_player\_data, 38
  - UI, 78
- playerWidget
  - UI, 78
- playerWindow
  - UI, 79
- players
  - UI, 78
- playing
  - Player, 51
- playingCamName
  - UI::display\_player\_data, 39
  - UI, 79
- plot
  - ui, 22
  - ui::Grapher, 41
- probe\_block\_stream
  - Recording, 59
- probe\_eos\_in\_stream
  - Recording, 60
- probe\_idle\_relink
  - Recording, 60
- readRoomsFromFile
  - Config, 36
- rec
  - gpio, 18
- rec\_cb
  - gpio, 17
- recImage
  - Camera, 31
- record
  - Camera, 31
- Recorder, 52
  - ~Recorder, 53
  - checkIfRecStopped, 54
  - config, 55
  - getRunningRecordings, 54
  - isGDriveUploadActive, 54
  - Recorder, 53
  - runningGDriveUploads, 55
  - runningRecordings, 55
  - startRecording, 54
  - stopRecording, 54
  - uploadVideo, 55
- recorder
  - UI::gdrive\_status\_data, 40
  - UI::switch\_state\_changed\_data, 69
  - UI, 79
- Recorder/include/room.h
  - room\_t, 91
- Recorder/src/main.cpp
  - main, 96
- Recording, 56
  - ~Recording, 57
  - buildPipeline, 57

- busSyncHandler, 58
- camName, 61
- clock, 61
- data\_probe, 58
- depay, 61
- elementSinkLinked, 58
- elementSrcLinked, 58
- enc, 62
- fakesink, 62
- fileName, 62
- folder, 62
- freeze\_check, 58
- freeze\_check\_id, 62
- getFileName, 59
- getStatus, 59
- in\_idle\_probe, 62
- lastBufferTime, 62
- mux, 62
- pad\_added\_handler, 59
- parse, 63
- pipeline, 63
- probe\_block\_stream, 59
- probe\_eos\_in\_stream, 60
- probe\_idle\_relink, 60
- Recording, 57
- relinkElements, 60
- sink, 63
- src, 63
- start, 61
- status, 63
- stop, 61
- streamLinked, 63
- streamcapsfilter, 63
- testcapsfilter, 63
- testsrc, 64
- timeout, 64
- uri, 64
- videoTimeLimit, 64
- recording.h
  - status\_t, 100
- relinkElements
  - Recording, 60
- res
  - video-upload, 25
- rightAxis
  - ui, 22
- Room, 64
  - audio, 68
  - cameras, 68
  - getAudioSource, 66
  - getCameras, 66
  - getName, 66, 67
  - name, 68
  - Room, 65, 66
  - setAudioSource, 67
  - setCameras, 67
  - setName, 67
  - type, 68
- room\_t
  - Operator/Grid/include/room.h, 90
  - Recorder/include/room.h, 91
- rooms
  - Config, 37
  - UI, 79
- run
  - ui::Pinger, 44
- runPing
  - ui, 19
- runningGDriveUploads
  - Recorder, 55
- runningRecordings
  - Recorder, 55
- SCOPES
  - from-gsuite, 15
  - video-upload, 25
- scale
  - Player, 51
- setAudioSource
  - Room, 67
- setCameras
  - Room, 67
- setData
  - ui::Grapher, 41
- setFile
  - Config, 37
- setName
  - Room, 67
- setPing
  - ui::Camera, 28
- sink
  - Player, 51
  - Recording, 63
- src
  - PadData, 42
  - Player, 52
  - Recording, 63
- start
  - Recording, 61
- startRecording
  - Recorder, 54
- status
  - Recording, 63
- status\_t
  - recording.h, 100
- stop
  - gpio, 18
  - Recording, 61
- stop\_cb
  - gpio, 17
- stopRecording
  - Recorder, 54
- stopStream
  - Player, 49
- store
  - video-upload, 25
- streamLinked

- Recording, 63
- streamcapsfilter
  - Recording, 63
- stylesheet
  - ui, 22
- switchGridV
  - UI, 79
- switchStateChanged
  - UI, 76
- testcapsfilter
  - Recording, 63
- testsrc
  - Recording, 64
- threadpool
  - ui, 22
- tickFont
  - ui, 22
- timeout
  - Recording, 64
- True
  - ui, 23
- type
  - Room, 68
  - video-upload, 25
- UI::display\_player\_data, 38
  - cam, 38
  - player, 38
  - playerLabel, 38
  - playingCamName, 39
- UI::gdrive\_status\_data, 39
  - GDriveIcon, 39
  - recorder, 40
- UI::switch\_state\_changed\_data, 69
  - cam, 69
  - recorder, 69
- UI, 70
  - ~UI, 72
  - config, 77
  - displayPlayer, 72, 73
  - displayRecordingStatus, 73
  - editButton, 78
  - editButtonClicked, 73
  - findIP, 73
  - initCamWidgets, 74
  - initMenuWidgets, 74
  - initPlayerWidgets, 75
  - initRoomTab, 75
  - initStyles, 75
  - keyPress, 76
  - menuBuilder, 78
  - menuWindow, 78
  - on\_show, 76
  - pageSwitched, 76
  - player, 78
  - playerBuilder, 78
  - playerLabel, 78
  - playerWidget, 78
  - playerWindow, 79
  - players, 78
  - playingCamName, 79
  - recorder, 79
  - rooms, 79
  - switchGridV, 79
  - switchStateChanged, 76
  - UI, 71, 72
  - updateGDriveStatus, 77
  - windowInit, 77
- ui, 18
  - addresses, 19
  - alpha, 20
  - app, 20
  - button, 20
  - camera, 20
  - cams, 20
  - display0, 20
  - display1, 21
  - font, 21
  - graph, 21
  - grapher, 21
  - i, 21
  - label, 21
  - labelStyle, 21
  - leftAxis, 21
  - menu, 22
  - pingTimer, 22
  - pinger, 22
  - plot, 22
  - rightAxis, 22
  - runPing, 19
  - stylesheet, 22
  - threadpool, 22
  - tickFont, 22
  - True, 23
  - units, 23
  - widget, 23
  - x, 23
  - y, 23
- ui.Camera, 28
- ui.Grapher, 40
- ui.Pinger, 42
- ui.UIGraph, 80
- ui.UIMenu, 81
- ui::Camera
  - \_\_init\_\_, 28
  - button, 29
  - drawMe, 28
  - grapher, 29
  - ip, 29
  - label, 29
  - pingHistory, 29
  - setPing, 28
- ui::Grapher
  - \_\_init\_\_, 40
  - activeCam, 41
  - changeCam, 40

- curve, [41](#)
- data, [41](#)
- plot, [41](#)
- setData, [41](#)
- ui::Pinger
  - \_\_init\_\_, [43](#)
  - \_runCmd, [44](#)
  - camList, [44](#)
  - run, [44](#)
- ui::UIGraph
  - \_\_init\_\_, [80](#)
- ui::UIMenu
  - \_\_init\_\_, [81](#)
- units
  - ui, [23](#)
- updateGDriveStatus
  - UI, [77](#)
- uploadVideo
  - Recorder, [55](#)
- uri
  - AudioSource, [27](#)
  - Camera, [31](#)
  - Player, [52](#)
  - Recording, [64](#)
- video-upload, [23](#)
  - args, [24](#)
  - creds, [24](#)
  - DRIVE, [24](#)
  - filename, [24](#)
  - flow, [24](#)
  - help, [24](#)
  - metadata, [24](#)
  - parser, [24](#)
  - res, [25](#)
  - SCOPES, [25](#)
  - store, [25](#)
  - type, [25](#)
- videoTimeLimit
  - Recording, [64](#)
- videoWidgetDraw\_cb
  - Player, [49](#), [50](#)
- videoWidgetRealize\_cb
  - Player, [50](#)
- videoWindow
  - Player, [52](#)
- videoWindowHandle
  - Player, [52](#)
- widget
  - ui, [23](#)
- windowInit
  - UI, [77](#)
- x
  - ui, [23](#)
- y
  - ui, [23](#)