

# Visual Programming

ver:1.1.0.0

制作者 小米机器人事业部-尚子涵 (Doxygen 1.8.17 2023.04.07)



<b>1 cyberdog_vp</b>	<b>1</b>
1.1 许可证 (License) . . . . .	1
1.2 cyberdog_vp 设计文档 - 中文版 . . . . .	1
1.2.1 1. 修订 . . . . .	1
1.2.2 2. 概述 . . . . .	1
1.2.3 3. 设计 . . . . .	1
1.2.3.1 3.1. 操作流程设计 . . . . .	1
1.2.3.2 3.2 技术架构 . . . . .	4
1.2.3.3 3.3 交互时序 . . . . .	5
1.2.3.4 3.4 文件架构 . . . . .	5
1.2.3.5 3.5 模块架构 . . . . .	6
1.2.4 4 附录 . . . . .	13
1.2.4.1 4.1 自定义动作说明 . . . . .	13
<b>2 Bug 列表</b>	<b>15</b>
<b>3 命名空间索引</b>	<b>17</b>
3.1 命名空间列表 . . . . .	17
<b>4 继承关系索引</b>	<b>19</b>
4.1 类继承关系 . . . . .	19
<b>5 类索引</b>	<b>23</b>
5.1 类列表 . . . . .	23
<b>6 文件索引</b>	<b>25</b>
6.1 文件列表 . . . . .	25
<b>7 命名空间文档</b>	<b>29</b>
7.1 cyberdog_visual_programming_abilityset 命名空间参考 . . . . .	29
7.1.1 类说明 . . . . .	33
7.1.1.1 struct cyberdog_visual_programming_abilityset::DefaultAndMaximum . . . . .	33
7.1.1.2 struct cyberdog_visual_programming_abilityset::GestureData . . . . .	34
7.1.1.3 struct cyberdog_visual_programming_abilityset::ObstacleMeta . . . . .	35
7.1.1.4 struct cyberdog_visual_programming_abilityset::RPY . . . . .	36
7.1.1.5 struct cyberdog_visual_programming_abilityset::TofObstacle . . . . .	37
7.1.2 类型定义说明 . . . . .	37
7.1.2.1 ActNavigation . . . . .	37
7.1.2.2 CyberdogJson . . . . .	38
7.1.2.3 CyberdogToml . . . . .	38
7.1.2.4 MsgAlgoStatus . . . . .	38
7.1.2.5 MsgAudioPlay . . . . .	38
7.1.2.6 MsgAudioPlayExtend . . . . .	38
7.1.2.7 MsgBmsStatus . . . . .	38

---

7.1.2.8 MsgBool . . . . .	38
7.1.2.9 MsgConnectorStatus . . . . .	39
7.1.2.10 MsgFaceRes . . . . .	39
7.1.2.11 MsgGesture . . . . .	39
7.1.2.12 MsgGpsPayload . . . . .	39
7.1.2.13 MsgHeader . . . . .	39
7.1.2.14 MsgHeadTofPayload . . . . .	39
7.1.2.15 MsgImu . . . . .	39
7.1.2.16 MsgLaserScan . . . . .	40
7.1.2.17 MsgMotionID . . . . .	40
7.1.2.18 MsgMotionSequenceGait . . . . .	40
7.1.2.19 MsgMotionSequencePace . . . . .	40
7.1.2.20 MsgMotionServoCmd . . . . .	40
7.1.2.21 MsgMotionServoResponse . . . . .	40
7.1.2.22 MsgMotionStatus . . . . .	40
7.1.2.23 MsgOdometry . . . . .	41
7.1.2.24 MsgPersonnel . . . . .	41
7.1.2.25 MsgPoint . . . . .	41
7.1.2.26 MsgPose . . . . .	41
7.1.2.27 MsgPoseStamped . . . . .	41
7.1.2.28 MsgPreset . . . . .	41
7.1.2.29 MsgRange . . . . .	41
7.1.2.30 MsgRearTofPayload . . . . .	42
7.1.2.31 MsgSingleTofPayload . . . . .	42
7.1.2.32 MsgSport . . . . .	42
7.1.2.33 MsgString . . . . .	42
7.1.2.34 MsgTouchStatus . . . . .	42
7.1.2.35 MsgTrainingWords . . . . .	42
7.1.2.36 MsgVisualProgrammingOperate . . . . .	42
7.1.2.37 SrvAudioGetVolume . . . . .	43
7.1.2.38 SrvAudioSetVolume . . . . .	43
7.1.2.39 SrvAudioTextPlay . . . . .	43
7.1.2.40 SrvCancelNavigation . . . . .	43
7.1.2.41 SrvFaceRec . . . . .	43
7.1.2.42 SrvGesture . . . . .	43
7.1.2.43 SrvGetPreset . . . . .	43
7.1.2.44 SrvLedExecute . . . . .	44
7.1.2.45 SrvMotionResultCmd . . . . .	44
7.1.2.46 SrvMotionSequenceShow . . . . .	44
7.1.2.47 SrvPersonnel . . . . .	44
7.1.2.48 SrvSetBool . . . . .	44
7.1.2.49 SrvSport . . . . .	44

---

7.1.2.50 SrvTrainingWords . . . . .	44
7.1.2.51 SrvVisualProgrammingOperate . . . . .	45
7.1.3 枚举类型说明 . . . . .	45
7.1.3.1 GestureType . . . . .	45
7.1.3.2 LedConstraint . . . . .	45
7.1.3.3 MotionId . . . . .	47
7.1.3.4 Processor . . . . .	48
7.1.3.5 RPYType . . . . .	49
7.1.3.6 SkeletonType . . . . .	49
7.1.3.7 StateCode . . . . .	49
7.1.3.8 TimeMode . . . . .	50
7.1.4 函数说明 . . . . .	51
7.1.4.1 Angle2Radian() . . . . .	51
7.1.4.2 covariance36() . . . . .	51
7.1.4.3 covariance9() . . . . .	51
7.1.4.4 Format() . . . . .	52
7.1.4.5 GetTime() . . . . .	52
7.1.4.6 GetTimeNs() . . . . .	53
7.1.4.7 GetWorkspace() . . . . .	54
7.1.4.8 int2binary() . . . . .	54
7.1.4.9 intVectorToString() . . . . .	55
7.1.4.10 JudgeToml() . . . . .	55
7.1.4.11 msgFaceResMap() . . . . .	55
7.1.4.12 msgFaceResVector() . . . . .	55
7.1.4.13 msgPersonnelVector() . . . . .	55
7.1.4.14 msgPresetMap() . . . . .	56
7.1.4.15 msgPresetVector() . . . . .	56
7.1.4.16 msgTrainingWordsMap() . . . . .	56
7.1.4.17 msgTrainingWordsVector() . . . . .	56
7.1.4.18 Qrientation2RPY() . . . . .	56
7.1.4.19 Radian2Angle() . . . . .	57
7.1.4.20 RPY2Qrientation() . . . . .	57
7.1.4.21 sequenceGaitVector() . . . . .	57
7.1.4.22 sequencePaceVector() . . . . .	58
7.1.4.23 stringVector() . . . . .	58
7.1.4.24 Timeout() . . . . .	58
7.1.5 变量说明 . . . . .	59
7.1.5.1 bms_data_cv_ . . . . .	59
7.1.5.2 bms_data_cvm_ . . . . .	59
7.1.5.3 compensation_frame_mtx_ . . . . .	59
7.1.5.4 gesture_recognition_state_cv_ . . . . .	59
7.1.5.5 gesture_recognition_state_cvm_ . . . . .	59

---

---

7.1.5.6 gps_data_cv_ . . . . .	59
7.1.5.7 gps_data_cvm_ . . . . .	60
7.1.5.8 imu_data_cv_ . . . . .	60
7.1.5.9 imu_data_cvm_ . . . . .	60
7.1.5.10 lidar_data_cv_ . . . . .	60
7.1.5.11 lidar_data_cvm_ . . . . .	60
7.1.5.12 odometer_data_cv_ . . . . .	60
7.1.5.13 odometer_data_cvm_ . . . . .	60
7.1.5.14 skeleton_recognition_state_cv_ . . . . .	60
7.1.5.15 skeleton_recognition_state_cvm_ . . . . .	61
7.1.5.16 task_state_cv_ . . . . .	61
7.1.5.17 task_state_cvm_ . . . . .	61
7.1.5.18 tof_data_cv_ . . . . .	61
7.1.5.19 tof_data_cvm_ . . . . .	61
7.1.5.20 touch_data_cv_ . . . . .	61
7.1.5.21 touch_data_cvm_ . . . . .	61
7.1.5.22 training_words_state_cv_ . . . . .	61
7.1.5.23 training_words_state_cvm_ . . . . .	62
7.1.5.24 ultrasonic_data_cv_ . . . . .	62
7.1.5.25 ultrasonic_data_cvm_ . . . . .	62
7.2 cyberdog_visual_programming_abilityset_py 命名空间参考 . . . . .	62
7.2.1 函数说明 . . . . .	64
7.2.1.1 DefineActNavigationResult() . . . . .	64
7.2.1.2 DefineAudio() . . . . .	64
7.2.1.3 DefineAudioGetVolumeSeviceResponse() . . . . .	65
7.2.1.4 DefineAudioPlaySeviceResponse() . . . . .	65
7.2.1.5 DefineAudioSetVolumeSeviceResponse() . . . . .	65
7.2.1.6 DefineBase() . . . . .	65
7.2.1.7 DefineBms() . . . . .	66
7.2.1.8 DefineBuiltInInterfaces() . . . . .	66
7.2.1.9 DefineCommonType() . . . . .	67
7.2.1.10 DefineCyberdog() . . . . .	67
7.2.1.11 DefineFaceRecognizedSeviceResponse() . . . . .	67
7.2.1.12 DefineFaceSeviceResponse() . . . . .	68
7.2.1.13 DefineFollow() . . . . .	68
7.2.1.14 DefineGeometryMsgs() . . . . .	68
7.2.1.15 DefineGesture() . . . . .	69
7.2.1.16 DefineGestureData() . . . . .	69
7.2.1.17 DefineGestureRecognizedMessageResponse() . . . . .	69
7.2.1.18 DefineGestureRecognizedSeviceResponse() . . . . .	69
7.2.1.19 DefineGestureType() . . . . .	70
7.2.1.20 DefineGps() . . . . .	70

---

7.2.1.21 DefineHeader()	70
7.2.1.22 DefineImu()	71
7.2.1.23 DefineLed()	71
7.2.1.24 DefineLedConstraint()	72
7.2.1.25 DefineLedServiceResponse()	72
7.2.1.26 DefineLidar()	72
7.2.1.27 DefineMapPresetServiceResponse()	72
7.2.1.28 DefineMotion()	73
7.2.1.29 DefineMotionId()	73
7.2.1.30 DefineMotionParams()	73
7.2.1.31 DefineMotionResultServiceResponse()	73
7.2.1.32 DefineMotionSequence()	74
7.2.1.33 DefineMotionSequenceServiceResponse()	74
7.2.1.34 DefineMotionServoCmdResponse()	74
7.2.1.35 DefineMsgBmsStatus()	74
7.2.1.36 DefineMsgConnectorStatus()	75
7.2.1.37 DefineMsgFaceRes()	75
7.2.1.38 DefineMsgGpsPayload()	75
7.2.1.39 DefineMsgHeadTofPayload()	76
7.2.1.40 DefineMsgImu()	76
7.2.1.41 DefineMsgLaserScan()	77
7.2.1.42 DefineMsgMotionSequenceGait()	77
7.2.1.43 DefineMsgMotionSequenceGaitList()	77
7.2.1.44 DefineMsgMotionSequencePace()	78
7.2.1.45 DefineMsgMotionSequencePaceList()	78
7.2.1.46 DefineMsgOdometry()	78
7.2.1.47 DefineMsgPersonnel()	79
7.2.1.48 DefineMsgPersonnelList()	79
7.2.1.49 DefineMsgPreset()	79
7.2.1.50 DefineMsgRange()	79
7.2.1.51 DefineMsgRearTofPayload()	80
7.2.1.52 DefineMsgSingleTofPayload()	80
7.2.1.53 DefineMsgSport()	80
7.2.1.54 DefineMsgTouchStatus()	81
7.2.1.55 DefineMsgTrainingWords()	81
7.2.1.56 DefineNavigation()	81
7.2.1.57 DefineNavigationActionResponse()	82
7.2.1.58 DefineNavMsgs()	82
7.2.1.59 DefineNetwork()	82
7.2.1.60 DefineObstacleMeta()	83
7.2.1.61 DefineOdometer()	83
7.2.1.62 DefinePersonnel()	83

---

7.2.1.63 DefinePoint()	84
7.2.1.64 DefinePose()	84
7.2.1.65 DefinePoseWithCovariance()	85
7.2.1.66 DefineProtocol()	85
7.2.1.67 DefineQuaternion()	86
7.2.1.68 DefineSensorMsgs()	87
7.2.1.69 DefineSkeleton()	87
7.2.1.70 DefineSkeletonRecognizedMessageResponse()	88
7.2.1.71 DefineSkeletonRecognizedSeviceResponse()	88
7.2.1.72 DefineSkeletonType()	88
7.2.1.73 DefineSrvAudioTextPlayResponse()	88
7.2.1.74 DefineSrvFaceRecResponse()	89
7.2.1.75 DefineSrvLedExecuteResponse()	89
7.2.1.76 DefineSrvMotionResultCmdResponse()	89
7.2.1.77 DefineSrvMotionSequenceShowResponse()	90
7.2.1.78 DefineSrvPersonnelResponse()	90
7.2.1.79 DefineSrvSportResponse()	90
7.2.1.80 DefineSrvTrainingWordsResponse()	91
7.2.1.81 DefineState()	91
7.2.1.82 DefineStdMsgs()	91
7.2.1.83 DefineTask()	92
7.2.1.84 DefineTimer()	92
7.2.1.85 DefineTof()	93
7.2.1.86 DefineTofObstacle()	93
7.2.1.87 DefineTofPayload()	93
7.2.1.88 DefineTouch()	94
7.2.1.89 DefineTrain()	94
7.2.1.90 DefineTrainingWordsRecognizedMessageResponse()	95
7.2.1.91 DefineTrainingWordsRecognizedSeviceResponse()	95
7.2.1.92 DefineTwist()	95
7.2.1.93 DefineTwistWithCovariance()	95
7.2.1.94 DefineUltrasonic()	96
7.2.1.95 DefineVector3()	96
7.2.1.96 DefineVoiceprintRecognizedResponse()	97
7.2.1.97 PYBIND11_MODULE()	97
7.3 cyberdog_visual_programming_engine 命名空间参考	98
7.3.1 类说明	100
7.3.1.1 struct cyberdog_visual_programming_engine::BackendMsg	100
7.3.2 类型定义说明	100
7.3.2.1 ASRMsg	100
7.3.2.2 AudioPlayExtendMsg	101
7.3.2.3 CyberdogJson	101

---

7.3.2.4 GRPCMsg . . . . .	101
7.3.2.5 HTPSrv . . . . .	101
7.3.2.6 OperateMsg . . . . .	101
7.3.2.7 OperateSrv . . . . .	101
7.3.2.8 PersonnelMsg . . . . .	101
7.3.2.9 PersonnelSrv . . . . .	102
7.3.2.10 PresetMsg . . . . .	102
7.3.2.11 PresetSrv . . . . .	102
7.3.2.12 ShellEnum . . . . .	102
7.3.2.13 TimeMode . . . . .	102
7.3.2.14 TrainingWordsMsg . . . . .	102
7.3.2.15 TrainingWordsSrv . . . . .	102
7.3.3 枚举类型说明 . . . . .	102
7.3.3.1 ShellEnum . . . . .	102
7.3.3.2 StateEnum . . . . .	103
7.3.3.3 TimeMode . . . . .	103
7.3.4 函数说明 . . . . .	104
7.3.4.1 GetTime() . . . . .	104
7.3.4.2 GetVector() . . . . .	104
7.3.4.3 GetWorkspace() . . . . .	105
7.3.4.4 JudgeConfileFile() . . . . .	105
7.3.4.5 Mkdir() . . . . .	106
7.3.4.6 Shell() . . . . .	106
7.3.4.7 Subreplace() . . . . .	106
7.3.5 变量说明 . . . . .	106
7.3.5.1 generate_derivative_file . . . . .	107
7.3.5.2 get_module_header . . . . .	107
7.3.5.3 get_task_header . . . . .	107
7.3.5.4 IPythonDemo . . . . .	107
7.4 cyberdog_visual_programming_terminal 命名空间参考 . . . . .	107
7.4.1 类型定义说明 . . . . .	108
7.4.1.1 CyberdogJson . . . . .	108
7.4.1.2 CyberdogToml . . . . .	108
7.4.1.3 GRPCMsg . . . . .	108
7.4.1.4 OperateMsg . . . . .	108
7.4.2 枚举类型说明 . . . . .	108
7.4.2.1 TimeMode . . . . .	108
7.4.3 函数说明 . . . . .	109
7.4.3.1 CoutJson() . . . . .	109
7.4.3.2 GetTime() . . . . .	109
7.4.3.3 GetVector() . . . . .	109
7.4.3.4 JudgeConfileFile() . . . . .	110

---

---

7.4.4 变量说明 . . . . .	110
7.4.4.1 cursor_mutex . . . . .	110
7.5 cyberdog_visual_programming_terminal.py 命名空间参考 . . . . .	110
7.5.1 函数说明 . . . . .	111
7.5.1.1 DefineDebugAbilityset() . . . . .	111
7.5.1.2 DefineDebugEngine() . . . . .	112
7.5.1.3 DefineDebugger() . . . . .	112
7.5.1.4 DefineInterface() . . . . .	113
7.5.1.5 DefineInterfaceAudio() . . . . .	113
7.5.1.6 DefineInterfaceBms() . . . . .	114
7.5.1.7 DefineInterfaceCyberdog() . . . . .	114
7.5.1.8 DefineInterfaceFollow() . . . . .	114
7.5.1.9 DefineInterfaceGesture() . . . . .	115
7.5.1.10 DefineInterfaceGps() . . . . .	115
7.5.1.11 DefineInterfaceImu() . . . . .	115
7.5.1.12 DefineInterfaceLed() . . . . .	116
7.5.1.13 DefineInterfaceLidar() . . . . .	116
7.5.1.14 DefineInterfaceMotion() . . . . .	116
7.5.1.15 DefineInterfaceNavigation() . . . . .	117
7.5.1.16 DefineInterfaceNetwork() . . . . .	117
7.5.1.17 DefineInterfaceOdometer() . . . . .	117
7.5.1.18 DefineInterfacePersonnel() . . . . .	118
7.5.1.19 DefineInterfaceSkeleton() . . . . .	118
7.5.1.20 DefineInterfaceTask() . . . . .	118
7.5.1.21 DefineInterfaceTof() . . . . .	119
7.5.1.22 DefineInterfaceTouch() . . . . .	119
7.5.1.23 DefineInterfaceTrain() . . . . .	119
7.5.1.24 DefineInterfaceType() . . . . .	120
7.5.1.25 DefineInterfaceTypeClass() . . . . .	120
7.5.1.26 DefineInterfaceTypeEnum() . . . . .	120
7.5.1.27 DefineInterfaceUltrasonic() . . . . .	121
7.5.1.28 DefineInterfaceVisual() . . . . .	121
7.5.1.29 DefineInterfaceVisualDebugAbilityset() . . . . .	121
7.5.1.30 DefineInterfaceVisualDebugger() . . . . .	122
7.5.1.31 DefineInterfaceVisualDebuggerEngine() . . . . .	122
7.5.1.32 DefineInterfaceVisualInterface() . . . . .	122
7.5.1.33 DefineVisual() . . . . .	123
7.5.1.34 PYBIND11_MODULE() . . . . .	123
8 类说明 . . . . .	125
8.1 cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset结构体 参考 . . . . .	125
8.1.1 构造及析构函数说明 . . . . .	126

---

8.1.1.1 Abilityset()	126
8.1.2 成员函数说明	126
8.1.2.1 Abilityset()	126
8.1.3 类成员变量说明	126
8.1.3.1 show_info	126
8.2 cyberdog_visual_programming_engine::ArtificialIntelligence类 参考	126
8.2.1 构造及析构函数说明	127
8.2.1.1 ArtificialIntelligence()	127
8.2.1.2 ~ArtificialIntelligence()	127
8.2.2 成员函数说明	127
8.2.2.1 Init()	127
8.2.2.2 RespondToRequests()	127
8.3 cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio结构体 参考	128
8.3.1 构造及析构函数说明	128
8.3.1.1 Audio()	128
8.3.2 成员函数说明	128
8.3.2.1 Audio_()	128
8.3.3 类成员变量说明	129
8.3.3.1 show_info	129
8.4 cyberdog_visual_programming_abilityset::Audio类 参考	129
8.4.1 构造及析构函数说明	132
8.4.1.1 Audio()	132
8.4.1.2 ~Audio()	132
8.4.2 成员函数说明	132
8.4.2.1 GetVolume()	132
8.4.2.2 OfflineInstantlyPlay()	133
8.4.2.3 OfflinePlay()	133
8.4.2.4 OnlineInstantlyPlay()	134
8.4.2.5 OnlinePlay()	134
8.4.2.6 SetVolume()	135
8.5 cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse类 参考	135
8.5.1 详细描述	136
8.5.2 构造及析构函数说明	136
8.5.2.1 AudioGetVolumeSeviceResponse()	137
8.5.2.2 ~AudioGetVolumeSeviceResponse()	137
8.5.3 类成员变量说明	137
8.5.3.1 response	137
8.5.3.2 state	137
8.6 cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse类 参考	137
8.6.1 详细描述	138
8.6.2 构造及析构函数说明	138
8.6.2.1 AudioPlaySeviceResponse()	138

---

8.6.2.2 ~AudioPlaySeviceResponse()	139
8.6.3 类成员变量说明	139
8.6.3.1 response	139
8.6.3.2 state	139
8.7 cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse类 参考	139
8.7.1 详细描述	140
8.7.2 构造及析构函数说明	140
8.7.2.1 AudioSetVolumeSeviceResponse()	141
8.7.2.2 ~AudioSetVolumeSeviceResponse()	141
8.7.3 类成员变量说明	141
8.7.3.1 response	141
8.7.3.2 state	141
8.8 cyberdog_visual_programming_engine::BackendMessage类 参考	141
8.8.1 成员类型定义说明	143
8.8.1.1 CommonEnum	143
8.8.1.2 HttpStatusCode	143
8.8.2 构造及析构函数说明	143
8.8.2.1 BackendMessage()	143
8.8.2.2 ~BackendMessage()	143
8.8.3 成员函数说明	143
8.8.3.1 getRobotendMsg()	143
8.8.4 类成员变量说明	144
8.8.4.1 backend_	144
8.8.4.2 describe_	144
8.8.4.3 errors	144
8.8.4.4 keys	144
8.8.4.5 state_	144
8.9 cyberdog_visual_programming_abilityset::Base类 参考	145
8.9.1 构造及析构函数说明	147
8.9.1.1 Base() [1/2]	147
8.9.1.2 Base() [2/2]	147
8.9.1.3 ~Base()	147
8.9.2 成员函数说明	147
8.9.2.1 Debug() [1/2]	148
8.9.2.2 Debug() [2/2]	148
8.9.2.3 DebugIf() [1/2]	149
8.9.2.4 DebugIf() [2/2]	149
8.9.2.5 Error() [1/2]	149
8.9.2.6 Error() [2/2]	150
8.9.2.7 ErrorIf() [1/2]	151
8.9.2.8 ErrorIf() [2/2]	151
8.9.2.9 Fatal() [1/2]	151

---

8.9.2.10 Fatal() [2/2] . . . . .	152
8.9.2.11 FatalIf() [1/2] . . . . .	152
8.9.2.12 FatalIf() [2/2] . . . . .	152
8.9.2.13 GetDescribe() . . . . .	153
8.9.2.14 GetState() . . . . .	153
8.9.2.15 GetToml() . . . . .	154
8.9.2.16 Heartbeat() . . . . .	154
8.9.2.17 Info() [1/2] . . . . .	155
8.9.2.18 Info() [2/2] . . . . .	155
8.9.2.19 InfoIf() [1/2] . . . . .	155
8.9.2.20 InfoIf() [2/2] . . . . .	156
8.9.2.21 Init() . . . . .	156
8.9.2.22 SetLog() . . . . .	156
8.9.2.23 SetState() . . . . .	157
8.9.2.24 Warn() [1/2] . . . . .	158
8.9.2.25 Warn() [2/2] . . . . .	159
8.9.2.26 WarnIf() [1/2] . . . . .	160
8.9.2.27 WarnIf() [2/2] . . . . .	160
8.9.3 类成员变量说明 . . . . .	161
8.9.3.1 heartbeat . . . . .	161
8.9.3.2 heartbeat_cb_group . . . . .	161
8.9.3.3 heartbeat_ptr . . . . .	161
8.9.3.4 log_ . . . . .	161
8.9.3.5 log_ptr . . . . .	161
8.9.3.6 logger_ . . . . .	161
8.9.3.7 node_immortal_ptr . . . . .	161
8.9.3.8 node_mortal_ptr . . . . .	162
8.9.3.9 state_ . . . . .	162
8.9.3.10 task_id_ . . . . .	162
8.9.3.11 timens_ . . . . .	162
8.9.3.12 timeout_wait_for_ . . . . .	162
8.9.3.13 timeout_wait_for_service_ . . . . .	162
8.9.3.14 transient_state_ . . . . .	162
8.10 cyberdog_visual_programming_engine::Base类参考 . . . . .	163
8.10.1 构造及析构函数说明 . . . . .	165
8.10.1.1 Base() . . . . .	165
8.10.1.2 ~Base() . . . . .	165
8.10.2 成员函数说明 . . . . .	165
8.10.2.1 BuildFrontendOperate() . . . . .	165
8.10.2.2 ExecuteRequest() . . . . .	166
8.10.2.3 GetList() . . . . .	166
8.10.2.4 GetMeta() [1/2] . . . . .	166

---

---

8.10.2.5 GetMeta() [2/2] . . . . .	167
8.10.2.6 GetRegistryToml() . . . . .	167
8.10.2.7 getRobotendMsg() . . . . .	168
8.10.2.8 GetState() . . . . .	168
8.10.2.9 Init() . . . . .	169
8.10.2.10 InitData() . . . . .	169
8.10.2.11 RespondToRequests() . . . . .	170
8.10.2.12 SetList() . . . . .	170
8.10.2.13 SetRegistryToml() . . . . .	171
8.10.3 类成员变量说明 . . . . .	171
8.10.3.1 base_path_ . . . . .	171
8.10.3.2 describe_ . . . . .	171
8.10.3.3 header_sh_ . . . . .	171
8.10.3.4 judge_python_ . . . . .	171
8.10.3.5 node_ptr_ . . . . .	172
8.10.3.6 params_toml_ . . . . .	172
8.10.3.7 py_interpreter_ptr_ . . . . .	172
8.10.3.8 registry_fil_ . . . . .	172
8.10.3.9 registry_related_fil_ . . . . .	172
8.10.3.10 state_ . . . . .	172
8.10.3.11 type_ . . . . .	172
8.11 cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms结构体 参考 . . . . .	173
8.11.1 构造及析构函数说明 . . . . .	173
8.11.1.1 Bms() . . . . .	173
8.11.2 成员函数说明 . . . . .	173
8.11.2.1 Bms_() . . . . .	173
8.11.3 类成员变量说明 . . . . .	174
8.11.3.1 show_info_ . . . . .	174
8.12 cyberdog_visual_programming_abilityset::Bms类 参考 . . . . .	174
8.12.1 构造及析构函数说明 . . . . .	177
8.12.1.1 Bms() . . . . .	177
8.12.1.2 ~Bms() . . . . .	177
8.12.2 成员函数说明 . . . . .	177
8.12.2.1 GetData() . . . . .	177
8.12.3 类成员变量说明 . . . . .	178
8.12.3.1 data_ . . . . .	178
8.13 cyberdog_visual_programming_terminal::Interface::Type::Class结构体 参考 . . . . .	178
8.13.1 构造及析构函数说明 . . . . .	178
8.13.1.1 Class() . . . . .	179
8.13.2 成员函数说明 . . . . .	179
8.13.2.1 Class_() . . . . .	179
8.13.3 类成员变量说明 . . . . .	179

---

8.13.3.1 show_info_ . . . . .	179
8.14 cyberdog_visual_programming_terminal::Interface::Cyberdog结构体 参考 . . . . .	179
8.14.1 构造及析构函数说明 . . . . .	180
8.14.1.1 Cyberdog() . . . . .	180
8.14.2 成员函数说明 . . . . .	180
8.14.2.1 Cyberdog_() . . . . .	180
8.14.3 类成员变量说明 . . . . .	181
8.14.3.1 audio_ . . . . .	181
8.14.3.2 bms_ . . . . .	181
8.14.3.3 follow_ . . . . .	181
8.14.3.4 gesture_ . . . . .	181
8.14.3.5 gps_ . . . . .	181
8.14.3.6 imu_ . . . . .	181
8.14.3.7 led_ . . . . .	182
8.14.3.8 lidar_ . . . . .	182
8.14.3.9 motion_ . . . . .	182
8.14.3.10 navigation_ . . . . .	182
8.14.3.11 network_ . . . . .	182
8.14.3.12 odometer_ . . . . .	182
8.14.3.13 personnel_ . . . . .	182
8.14.3.14 show_info_ . . . . .	183
8.14.3.15 skeleton_ . . . . .	183
8.14.3.16 task_ . . . . .	183
8.14.3.17 tof_ . . . . .	183
8.14.3.18 touch_ . . . . .	183
8.14.3.19 train_ . . . . .	183
8.14.3.20 ultrasonic_ . . . . .	183
8.15 cyberdog_visual_programming_abilityset::Cyberdog类 参考 . . . . .	184
8.15.1 构造及析构函数说明 . . . . .	185
8.15.1.1 Cyberdog() . . . . .	186
8.15.1.2 ~Cyberdog() . . . . .	186
8.15.2 成员函数说明 . . . . .	186
8.15.2.1 SetLog() . . . . .	186
8.15.2.2 Shutdown() . . . . .	187
8.15.3 类成员变量说明 . . . . .	187
8.15.3.1 audio_ . . . . .	187
8.15.3.2 bms_ . . . . .	187
8.15.3.3 follow_ . . . . .	187
8.15.3.4 gesture_ . . . . .	188
8.15.3.5 gps_ . . . . .	188
8.15.3.6 imu_ . . . . .	188
8.15.3.7 led_ . . . . .	188

---

8.15.3.8 lidar_ . . . . .	188
8.15.3.9 motion_ . . . . .	188
8.15.3.10 navigation_ . . . . .	188
8.15.3.11 network_ . . . . .	188
8.15.3.12 odometer_ . . . . .	189
8.15.3.13 personnel_ . . . . .	189
8.15.3.14 skeleton_ . . . . .	189
8.15.3.15 task_ . . . . .	189
8.15.3.16 tof_ . . . . .	189
8.15.3.17 touch_ . . . . .	189
8.15.3.18 train_ . . . . .	189
8.15.3.19 ultrasonic_ . . . . .	189
8.16 cyberdog_visual_programming_terminal::DebugAbilityset类 参考 . . . . .	190
8.16.1 构造及析构函数说明 . . . . .	190
8.16.1.1 DebugAbilityset() . . . . .	190
8.16.1.2 ~DebugAbilityset() . . . . .	190
8.16.2 成员函数说明 . . . . .	190
8.16.2.1 Init() . . . . .	191
8.16.2.2 Launch() . . . . .	191
8.16.2.3 Stop() . . . . .	191
8.17 cyberdog_visual_programming_terminal::DebugEngine类 参考 . . . . .	191
8.17.1 构造及析构函数说明 . . . . .	192
8.17.1.1 DebugEngine() . . . . .	192
8.17.1.2 ~DebugEngine() . . . . .	192
8.17.2 成员函数说明 . . . . .	192
8.17.2.1 Init() . . . . .	193
8.17.2.2 MockAppRequest() . . . . .	193
8.18 cyberdog_visual_programming_terminal::Debugger类 参考 . . . . .	193
8.18.1 构造及析构函数说明 . . . . .	194
8.18.1.1 Debugger() . . . . .	194
8.18.1.2 ~Debugger() . . . . .	194
8.18.2 类成员变量说明 . . . . .	195
8.18.2.1 abilityset_ . . . . .	195
8.18.2.2 engine_ . . . . .	195
8.19 cyberdog_visual_programming_terminal::Interface::Visual::Debugger结构体 参考 . . . . .	195
8.19.1 构造及析构函数说明 . . . . .	196
8.19.1.1 Debugger() . . . . .	196
8.19.2 成员函数说明 . . . . .	196
8.19.2.1 Debugger_() . . . . .	196
8.19.3 类成员变量说明 . . . . .	196
8.19.3.1 abilityset_ . . . . .	196
8.19.3.2 engine_ . . . . .	196

---

8.19.3.3 show_info_ . . . . .	197
8.20 cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine结构体 参考 . . . . .	197
8.20.1 构造及析构函数说明 . . . . .	197
8.20.1.1 Engine() . . . . .	197
8.20.2 成员函数说明 . . . . .	198
8.20.2.1 Engine_() . . . . .	198
8.20.3 类成员变量说明 . . . . .	198
8.20.3.1 show_info_ . . . . .	198
8.21 cyberdog_visual_programming_terminal::Interface::Type::Enum结构体 参考 . . . . .	198
8.21.1 构造及析构函数说明 . . . . .	199
8.21.1.1 Enum() . . . . .	199
8.21.2 成员函数说明 . . . . .	199
8.21.2.1 Enum_() . . . . .	199
8.21.3 类成员变量说明 . . . . .	199
8.21.3.1 show_info_ . . . . .	199
8.22 cyberdog_visual_programming_abilityset::Face类 参考 . . . . .	200
8.22.1 构造及析构函数说明 . . . . .	202
8.22.1.1 Face() . . . . .	202
8.22.1.2 ~Face() . . . . .	202
8.22.2 成员函数说明 . . . . .	202
8.22.2.1 CancelRecognize() . . . . .	202
8.22.2.2 Recognized() . . . . .	203
8.23 cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse类 参考 . . . . .	203
8.23.1 详细描述 . . . . .	204
8.23.2 构造及析构函数说明 . . . . .	204
8.23.2.1 FaceRecognizedSeviceResponse() . . . . .	205
8.23.2.2 ~FaceRecognizedSeviceResponse() . . . . .	205
8.23.3 类成员变量说明 . . . . .	205
8.23.3.1 dictionary . . . . .	205
8.23.3.2 list . . . . .	205
8.23.3.3 state . . . . .	205
8.24 cyberdog_visual_programming_abilityset::FaceSeviceResponse类 参考 . . . . .	206
8.24.1 详细描述 . . . . .	206
8.24.2 构造及析构函数说明 . . . . .	206
8.24.2.1 FaceSeviceResponse() . . . . .	207
8.24.2.2 ~FaceSeviceResponse() . . . . .	207
8.24.3 类成员变量说明 . . . . .	207
8.24.3.1 response . . . . .	207
8.24.3.2 state . . . . .	207
8.25 cyberdog_visual_programming_abilityset::Follow类 参考 . . . . .	207
8.25.1 构造及析构函数说明 . . . . .	210
8.25.1.1 Follow() . . . . .	210

---

---

8.25.1.2 ~Follow()	210
8.25.2 成员函数说明	210
8.25.2.1 AddPersonnel()	210
8.25.2.2 CancelFollow()	211
8.25.2.3 DeletePersonnel()	211
8.25.2.4 FollowPersonnel()	212
8.26 cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow结构体 参考	212
8.26.1 构造及析构函数说明	213
8.26.1.1 Follow()	213
8.26.2 成员函数说明	213
8.26.2.1 Follow_()	213
8.26.3 类成员变量说明	213
8.26.3.1 show_info_	213
8.27 cyberdog_visual_programming_engine::FrontendMessage类 参考	213
8.27.1 成员类型定义说明	214
8.27.1.1 CommonEnum	214
8.27.2 构造及析构函数说明	214
8.27.2.1 FrontendMessage()	214
8.27.2.2 ~FrontendMessage()	214
8.27.3 成员函数说明	214
8.27.3.1 getRobotendMsg()	215
8.27.4 类成员变量说明	215
8.27.4.1 describe_	215
8.27.4.2 frontend_	215
8.27.4.3 keys	215
8.27.4.4 state_	215
8.28 cyberdog_visual_programming_engine::Fsm类 参考	216
8.28.1 构造及析构函数说明	217
8.28.1.1 Fsm()	218
8.28.1.2 ~Fsm()	218
8.28.2 成员函数说明	218
8.28.2.1 Init()	218
8.28.2.2 OnActive()	219
8.28.2.3 OnDeActive()	219
8.28.2.4 OnError()	219
8.28.2.5 OnLowPower()	220
8.28.2.6 OnOTA()	220
8.28.2.7 OnProtected()	220
8.28.2.8 OnSelfCheck()	221
8.28.2.9 OnSetUp()	221
8.28.2.10 ONTearDown()	221
8.28.2.11 RespondToRequests()	222

---

8.29 cyberdog_visual_programming_abilityset::Gesture类 参考 . . . . .	222
8.29.1 构造及析构函数说明 . . . . .	225
8.29.1.1 Gesture() . . . . .	225
8.29.1.2 ~Gesture() . . . . .	225
8.29.2 成员函数说明 . . . . .	225
8.29.2.1 Recognized() . . . . .	225
8.29.2.2 RecognizedAnyGesture() . . . . .	226
8.29.2.3 RecognizedDesignatedGesture() . . . . .	226
8.29.2.4 TurnOffRecognition() . . . . .	227
8.29.2.5 TurnOnRecognition() . . . . .	227
8.30 cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture结构体 参考 . . . . .	228
8.30.1 构造及析构函数说明 . . . . .	228
8.30.1.1 Gesture() . . . . .	228
8.30.2 成员函数说明 . . . . .	228
8.30.2.1 Gesture_() . . . . .	229
8.30.3 类成员变量说明 . . . . .	229
8.30.3.1 show_info_ . . . . .	229
8.31 cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse类 参考 . . . . .	229
8.31.1 详细描述 . . . . .	230
8.31.2 构造及析构函数说明 . . . . .	231
8.31.2.1 GestureRecognizedMessageResponse() . . . . .	231
8.31.2.2 ~GestureRecognizedMessageResponse() . . . . .	231
8.31.3 类成员变量说明 . . . . .	231
8.31.3.1 data . . . . .	231
8.31.3.2 state . . . . .	231
8.32 cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse类 参考 . . . . .	232
8.32.1 详细描述 . . . . .	232
8.32.2 构造及析构函数说明 . . . . .	233
8.32.2.1 GestureRecognizedSeviceResponse() . . . . .	233
8.32.2.2 ~GestureRecognizedSeviceResponse() . . . . .	233
8.32.3 类成员变量说明 . . . . .	233
8.32.3.1 response . . . . .	233
8.32.3.2 state . . . . .	233
8.33 cyberdog_visual_programming_abilityset::Gps类 参考 . . . . .	234
8.33.1 构造及析构函数说明 . . . . .	236
8.33.1.1 Gps() . . . . .	236
8.33.1.2 ~Gps() . . . . .	236
8.33.2 成员函数说明 . . . . .	236
8.33.2.1 GetData() . . . . .	236
8.33.3 类成员变量说明 . . . . .	237
8.33.3.1 data_ . . . . .	237
8.34 cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps结构体 参考 . . . . .	237

---

---

8.34.1 构造及析构函数说明 . . . . .	237
8.34.1.1 Gps() . . . . .	238
8.34.2 成员函数说明 . . . . .	238
8.34.2.1 Gps_() . . . . .	238
8.34.3 类成员变量说明 . . . . .	238
8.34.3.1 show_info_ . . . . .	238
8.35 cyberdog_visual_programming_abilityset::Imu类 参考 . . . . .	238
8.35.1 构造及析构函数说明 . . . . .	241
8.35.1.1 Imu() . . . . .	241
8.35.1.2 ~Imu() . . . . .	241
8.35.2 成员函数说明 . . . . .	241
8.35.2.1 GetData() . . . . .	241
8.35.2.2 GetRPY() . . . . .	242
8.35.2.3 IfDeltaAngular() . . . . .	242
8.35.3 类成员变量说明 . . . . .	242
8.35.3.1 data_ . . . . .	242
8.36 cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu结构体 参考 . . . . .	242
8.36.1 构造及析构函数说明 . . . . .	243
8.36.1.1 Imu() . . . . .	243
8.36.2 成员函数说明 . . . . .	243
8.36.2.1 Imu_() . . . . .	243
8.36.3 类成员变量说明 . . . . .	243
8.36.3.1 show_info_ . . . . .	243
8.37 cyberdog_visual_programming_engine::Interact类 参考 . . . . .	244
8.37.1 构造及析构函数说明 . . . . .	244
8.37.1.1 Interact() . . . . .	244
8.37.1.2 ~Interact() . . . . .	245
8.37.2 成员函数说明 . . . . .	245
8.37.2.1 Init() . . . . .	245
8.38 cyberdog_visual_programming_terminal::Interface::Visual::Interface结构体 参考 . . . . .	245
8.38.1 构造及析构函数说明 . . . . .	246
8.38.1.1 Interface() . . . . .	246
8.38.2 成员函数说明 . . . . .	246
8.38.2.1 Interface_() . . . . .	246
8.38.3 类成员变量说明 . . . . .	246
8.38.3.1 show_info_ . . . . .	247
8.39 cyberdog_visual_programming_terminal::Interface类 参考 . . . . .	247
8.39.1 构造及析构函数说明 . . . . .	247
8.39.1.1 Interface() . . . . .	247
8.39.1.2 ~Interface() . . . . .	248
8.39.2 成员函数说明 . . . . .	248
8.39.2.1 Interface_() . . . . .	248

---

8.39.3 类成员变量说明 . . . . .	248
8.39.3.1 cyberdog_ . . . . .	248
8.39.3.2 type_ . . . . .	248
8.39.3.3 visual_ . . . . .	248
8.40 cyberdog_visual_programming_abilityset::Led类 参考 . . . . .	249
8.40.1 构造及析构函数说明 . . . . .	251
8.40.1.1 Led() . . . . .	251
8.40.1.2 ~Led() . . . . .	251
8.40.2 成员函数说明 . . . . .	251
8.40.2.1 Freed() . . . . .	251
8.40.2.2 Play() . . . . .	252
8.40.2.3 PlayRgb() . . . . .	252
8.41 cyberdog_visual_programming_terminal::Interface::Cyberdog::Led结构体 参考 . . . . .	253
8.41.1 构造及析构函数说明 . . . . .	253
8.41.1.1 Led() . . . . .	253
8.41.2 成员函数说明 . . . . .	253
8.41.2.1 Led_() . . . . .	253
8.41.3 类成员变量说明 . . . . .	254
8.41.3.1 show_info_ . . . . .	254
8.42 cyberdog_visual_programming_abilityset::LedSeviceResponse类 参考 . . . . .	254
8.42.1 详细描述 . . . . .	255
8.42.2 构造及析构函数说明 . . . . .	255
8.42.2.1 LedSeviceResponse() . . . . .	255
8.42.2.2 ~LedSeviceResponse() . . . . .	255
8.42.3 类成员变量说明 . . . . .	255
8.42.3.1 response . . . . .	255
8.42.3.2 state . . . . .	255
8.43 cyberdog_visual_programming_abilityset::Lidar类 参考 . . . . .	256
8.43.1 构造及析构函数说明 . . . . .	258
8.43.1.1 Lidar() . . . . .	258
8.43.1.2 ~Lidar() . . . . .	258
8.43.2 成员函数说明 . . . . .	258
8.43.2.1 GetData() . . . . .	258
8.43.3 类成员变量说明 . . . . .	259
8.43.3.1 data_ . . . . .	259
8.44 cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar结构体 参考 . . . . .	259
8.44.1 构造及析构函数说明 . . . . .	259
8.44.1.1 Lidar() . . . . .	260
8.44.2 成员函数说明 . . . . .	260
8.44.2.1 Lidar_() . . . . .	260
8.44.3 类成员变量说明 . . . . .	260
8.44.3.1 show_info_ . . . . .	260

---

---

8.45 cyberdog_visual_programming_engine::LinuxAt类 参考 . . . . .	260
8.45.1 构造及析构函数说明 . . . . .	261
8.45.1.1 LinuxAt() . . . . .	261
8.45.1.2 ~LinuxAt() . . . . .	261
8.45.2 成员函数说明 . . . . .	261
8.45.2.1 Cancellation() . . . . .	261
8.45.2.2 GetTaskAtMap() . . . . .	262
8.45.2.3 Init() . . . . .	262
8.45.2.4 Registration() . . . . .	262
8.46 cyberdog_visual_programming_engine::LinuxCron类 参考 . . . . .	263
8.46.1 构造及析构函数说明 . . . . .	263
8.46.1.1 LinuxCron() . . . . .	263
8.46.1.2 ~LinuxCron() . . . . .	263
8.46.2 成员函数说明 . . . . .	264
8.46.2.1 Cancellation() . . . . .	264
8.46.2.2 GetNowCrontab() . . . . .	264
8.46.2.3 Init() . . . . .	265
8.46.2.4 Registration() . . . . .	265
8.46.2.5 UpdateCrontabFile() . . . . .	265
8.47 cyberdog_visual_programming_abilityset::Log类 参考 . . . . .	266
8.47.1 成员枚举类型说明 . . . . .	267
8.47.1.1 Rank . . . . .	267
8.47.2 构造及析构函数说明 . . . . .	267
8.47.2.1 Log() [1/2] . . . . .	267
8.47.2.2 Log() [2/2] . . . . .	267
8.47.2.3 ~Log() . . . . .	268
8.47.3 成员函数说明 . . . . .	268
8.47.3.1 GetInstance() . . . . .	268
8.47.3.2 operator=() . . . . .	269
8.47.3.3 Record() . . . . .	269
8.47.3.4 SetData() . . . . .	270
8.47.3.5 SetMechanism() . . . . .	271
8.47.3.6 Trim() . . . . .	271
8.48 cyberdog_visual_programming_abilityset::MapPresetSeviceResponse类 参考 . . . . .	272
8.48.1 详细描述 . . . . .	273
8.48.2 构造及析构函数说明 . . . . .	273
8.48.2.1 MapPresetSeviceResponse() . . . . .	274
8.48.2.2 ~MapPresetSeviceResponse() . . . . .	274
8.48.3 类成员变量说明 . . . . .	274
8.48.3.1 dictionary . . . . .	274
8.48.3.2 is_outdoor . . . . .	274
8.48.3.3 list . . . . .	274

---

8.48.3.4 map_name . . . . .	274
8.48.3.5 state . . . . .	274
8.49 cyberdog_visual_programming_engine::Module类 参考 . . . . .	275
8.49.1 构造及析构函数说明 . . . . .	277
8.49.1.1 Module() . . . . .	277
8.49.1.2 ~Module() . . . . .	277
8.50 cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion结构体 参考 . . . . .	277
8.50.1 构造及析构函数说明 . . . . .	278
8.50.1.1 Motion() . . . . .	278
8.50.2 成员函数说明 . . . . .	278
8.50.2.1 Motion_() . . . . .	278
8.50.3 类成员变量说明 . . . . .	278
8.50.3.1 show_info_ . . . . .	278
8.51 cyberdog_visual_programming_abilityset::Motion类 参考 . . . . .	278
8.51.1 构造及析构函数说明 . . . . .	281
8.51.1.1 Motion() . . . . .	281
8.51.1.2 ~Motion() . . . . .	281
8.51.2 成员函数说明 . . . . .	281
8.51.2.1 AbsoluteForceControlAttitude() . . . . .	282
8.51.2.2 AbsolutePositionControlAttitude() . . . . .	282
8.51.2.3 AutomaticFrequencyConversionWalking() . . . . .	282
8.51.2.4 GoStraight() . . . . .	283
8.51.2.5 InitIndependent() . . . . .	283
8.51.2.6 JumpBackAndForth() . . . . .	284
8.51.2.7 LateralMovement() . . . . .	284
8.51.2.8 RelativelyForceControlAttitude() . . . . .	284
8.51.2.9 RelativelyPositionControlAttitude() . . . . .	285
8.51.2.10 Request() . . . . .	285
8.51.2.11 RunFastWalking() . . . . .	286
8.51.2.12 RunSequence() . . . . .	286
8.51.2.13 SmallJumpWalking() . . . . .	287
8.51.2.14 TrotWalking() . . . . .	287
8.51.2.15 Turn() . . . . .	288
8.51.2.16 WalkTheDog() . . . . .	288
8.51.3 类成员变量说明 . . . . .	288
8.51.3.1 motion_id_ . . . . .	288
8.51.3.2 params_ . . . . .	289
8.52 cyberdog_visual_programming_abilityset::MotionResultServiceResponse类 参考 . . . . .	289
8.52.1 详细描述 . . . . .	290
8.52.2 构造及析构函数说明 . . . . .	290
8.52.2.1 MotionResultServiceResponse() . . . . .	290
8.52.2.2 ~MotionResultServiceResponse() . . . . .	290

---

---

8.52.3 类成员变量说明 . . . . .	290
8.52.3.1 response . . . . .	290
8.52.3.2 state . . . . .	290
8.53 cyberdog_visual_programming_abilityset::MotionSequence类 参考 . . . . .	291
8.53.1 详细描述 . . . . .	291
8.53.2 构造及析构函数说明 . . . . .	291
8.53.2.1 MotionSequence() . . . . .	291
8.53.2.2 ~MotionSequence() . . . . .	292
8.53.3 类成员变量说明 . . . . .	292
8.53.3.1 describe . . . . .	292
8.53.3.2 gait_list . . . . .	292
8.53.3.3 name . . . . .	292
8.53.3.4 pace_list . . . . .	292
8.54 cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse类 参考 . . . . .	293
8.54.1 详细描述 . . . . .	293
8.54.2 构造及析构函数说明 . . . . .	294
8.54.2.1 MotionSequenceServiceResponse() . . . . .	294
8.54.2.2 ~MotionSequenceServiceResponse() . . . . .	294
8.54.3 类成员变量说明 . . . . .	294
8.54.3.1 response . . . . .	294
8.54.3.2 state . . . . .	294
8.55 cyberdog_visual_programming_abilityset::MotionServoCmdResponse类 参考 . . . . .	295
8.55.1 详细描述 . . . . .	295
8.55.2 构造及析构函数说明 . . . . .	295
8.55.2.1 MotionServoCmdResponse() . . . . .	296
8.55.2.2 ~MotionServoCmdResponse() . . . . .	296
8.55.3 类成员变量说明 . . . . .	296
8.55.3.1 response . . . . .	296
8.55.3.2 state . . . . .	296
8.56 cyberdog_visual_programming_abilityset::Navigation类 参考 . . . . .	296
8.56.1 构造及析构函数说明 . . . . .	299
8.56.1.1 Navigation() . . . . .	299
8.56.1.2 ~Navigation() . . . . .	299
8.56.2 成员函数说明 . . . . .	299
8.56.2.1 AddPreset() . . . . .	299
8.56.2.2 CancelNavigation() . . . . .	300
8.56.2.3 DeletePreset() . . . . .	300
8.56.2.4 GetPreset() . . . . .	300
8.56.2.5 InitDependent() . . . . .	301
8.56.2.6 NavigationToCoordinates() . . . . .	301
8.56.2.7 NavigationToPose() . . . . .	302
8.56.2.8 NavigationToPreset() . . . . .	302

---

8.56.2.9 ToPreset() . . . . .	303
8.56.2.10 TurnOffNavigation() . . . . .	303
8.56.2.11 TurnOnNavigation() . . . . .	304
8.57 cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation结构体 参考 . . . . .	304
8.57.1 构造及析构函数说明 . . . . .	305
8.57.1.1 Navigation() . . . . .	305
8.57.2 成员函数说明 . . . . .	305
8.57.2.1 Navigation_() . . . . .	305
8.57.3 类成员变量说明 . . . . .	305
8.57.3.1 show_info_ . . . . .	305
8.58 cyberdog_visual_programming_abilityset::NavigationActionResponse类 参考 . . . . .	306
8.58.1 详细描述 . . . . .	306
8.58.2 构造及析构函数说明 . . . . .	306
8.58.2.1 NavigationActionResponse() . . . . .	307
8.58.2.2 ~NavigationActionResponse() . . . . .	307
8.58.3 类成员变量说明 . . . . .	307
8.58.3.1 response . . . . .	307
8.58.3.2 state . . . . .	307
8.59 cyberdog_visual_programming_terminal::Interface::Cyberdog::Network结构体 参考 . . . . .	307
8.59.1 构造及析构函数说明 . . . . .	308
8.59.1.1 Network() . . . . .	308
8.59.2 成员函数说明 . . . . .	308
8.59.2.1 Network_() . . . . .	308
8.59.3 类成员变量说明 . . . . .	308
8.59.3.1 show_info_ . . . . .	308
8.60 cyberdog_visual_programming_abilityset::Network类 参考 . . . . .	309
8.60.1 构造及析构函数说明 . . . . .	311
8.60.1.1 Network() . . . . .	311
8.60.1.2 ~Network() . . . . .	311
8.60.2 类成员变量说明 . . . . .	311
8.60.2.1 data_ . . . . .	311
8.61 cyberdog_visual_programming_abilityset::Odometer类 参考 . . . . .	311
8.61.1 构造及析构函数说明 . . . . .	314
8.61.1.1 Odometer() . . . . .	314
8.61.1.2 ~Odometer() . . . . .	314
8.61.2 成员函数说明 . . . . .	314
8.61.2.1 GetData() . . . . .	314
8.61.2.2 IfCumulativeDistance() . . . . .	315
8.61.3 类成员变量说明 . . . . .	315
8.61.3.1 data_ . . . . .	315
8.62 cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer结构体 参考 . . . . .	315
8.62.1 构造及析构函数说明 . . . . .	316

---

---

8.62.1.1 Odometer()	316
8.62.2 成员函数说明	316
8.62.2.1 Odometer_()	316
8.62.3 类成员变量说明	316
8.62.3.1 show_info_	316
8.63 cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel结构体 参考	316
8.63.1 构造及析构函数说明	317
8.63.1.1 Personnel()	317
8.63.2 成员函数说明	317
8.63.2.1 Personnel_()	317
8.63.3 类成员变量说明	317
8.63.3.1 show_info_	317
8.64 cyberdog_visual_programming_abilityset::Personnel类 参考	318
8.64.1 构造及析构函数说明	320
8.64.1.1 Personnel()	320
8.64.1.2 ~Personnel()	320
8.64.2 成员函数说明	320
8.64.2.1 FaceRecognized()	320
8.64.2.2 GetData()	321
8.64.2.3 VoiceprintRecognized()	321
8.64.3 类成员变量说明	322
8.64.3.1 face_	322
8.64.3.2 voiceprint_	322
8.65 cyberdog_visual_programming_engine::PythonInterpreter类 参考	322
8.65.1 构造及析构函数说明	323
8.65.1.1 PythonInterpreter()	323
8.65.1.2 ~PythonInterpreter()	323
8.65.2 成员函数说明	323
8.65.2.1 GenerateDerivativeFile()	323
8.65.2.2 GetModuleHeader()	323
8.65.2.3 GetTaskHeader()	324
8.65.2.4 Init()	324
8.66 cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton结构体 参考	324
8.66.1 构造及析构函数说明	325
8.66.1.1 Skeleton()	325
8.66.2 成员函数说明	325
8.66.2.1 Skeleton_()	325
8.66.3 类成员变量说明	325
8.66.3.1 show_info_	325
8.67 cyberdog_visual_programming_abilityset::Skeleton类 参考	326
8.67.1 构造及析构函数说明	328
8.67.1.1 Skeleton()	328

8.67.1.2 ~Skeleton()	328
8.67.2 成员函数说明	328
8.67.2.1 BlockingRecognized()	328
8.67.2.2 InitDependent()	329
8.67.2.3 InstantRecognized()	329
8.67.2.4 SportsRecognition()	329
8.67.2.5 TurnOffRecognition()	330
8.67.2.6 TurnOnRecognition()	330
8.68 cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse类 参考	331
8.68.1 详细描述	332
8.68.2 构造及析构函数说明	332
8.68.2.1 SkeletonRecognizedMessageResponse()	332
8.68.2.2 ~SkeletonRecognizedMessageResponse()	332
8.68.3 类成员变量说明	332
8.68.3.1 response	332
8.68.3.2 state	332
8.69 cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse类 参考	333
8.69.1 详细描述	333
8.69.2 构造及析构函数说明	334
8.69.2.1 SkeletonRecognizedSeviceResponse()	334
8.69.2.2 ~SkeletonRecognizedSeviceResponse()	334
8.69.3 类成员变量说明	334
8.69.3.1 response	334
8.69.3.2 state	334
8.70 cyberdog_visual_programming_engine::Slam类 参考	334
8.70.1 构造及析构函数说明	335
8.70.1.1 Slam()	335
8.70.1.2 ~Slam()	335
8.70.2 成员函数说明	335
8.70.2.1 Init()	335
8.70.2.2 RespondToRequests()	335
8.71 cyberdog_visual_programming_abilityset::State类 参考	336
8.71.1 详细描述	336
8.71.2 构造及析构函数说明	336
8.71.2.1 State()	336
8.71.2.2 ~State()	337
8.71.3 类成员变量说明	337
8.71.3.1 code	337
8.71.3.2 describe	337
8.72 cyberdog_visual_programming_engine::Task类 参考	337
8.72.1 构造及析构函数说明	340
8.72.1.1 Task()	340

---

8.72.1.2 ~Task()	340
8.73 cyberdog_visual_programming_terminal::Interface::Cyberdog::Task结构体 参考	340
8.73.1 构造及析构函数说明	341
8.73.1.1 Task()	341
8.73.2 成员函数说明	341
8.73.2.1 Task_()	341
8.73.3 类成员变量说明	341
8.73.3.1 show_info_	341
8.74 cyberdog_visual_programming_abilityset::Task类 参考	341
8.74.1 构造及析构函数说明	344
8.74.1.1 Task()	344
8.74.1.2 ~Task()	344
8.74.2 成员函数说明	344
8.74.2.1 Block()	344
8.74.2.2 BreakpointBlock()	345
8.74.2.3 GetProcessor()	345
8.74.2.4 InitDependent()	345
8.74.2.5 Start()	346
8.74.2.6 Stop()	346
8.75 cyberdog_visual_programming_terminal::Interface::Tof结构体 参考	347
8.75.1 构造及析构函数说明	347
8.75.1.1 Tof()	347
8.75.2 成员函数说明	347
8.75.2.1 Tof_()	347
8.75.3 类成员变量说明	348
8.75.3.1 show_info_	348
8.76 cyberdog_visual_programming_abilityset::Tof类 参考	348
8.76.1 构造及析构函数说明	350
8.76.1.1 Tof()	351
8.76.1.2 ~Tof()	351
8.76.2 成员函数说明	351
8.76.2.1 GetData()	351
8.76.3 类成员变量说明	351
8.76.3.1 data_	351
8.76.3.2 obstacle_	352
8.77 cyberdog_visual_programming_abilityset::TofPayload类 参考	352
8.77.1 详细描述	352
8.77.2 构造及析构函数说明	352
8.77.2.1 TofPayload()	353
8.77.2.2 ~TofPayload()	353
8.77.3 类成员变量说明	353
8.77.3.1 head	353

8.77.3.2 rear . . . . .	353
8.78 cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch结构体 参考 . . . . .	353
8.78.1 构造及析构函数说明 . . . . .	354
8.78.1.1 Touch() . . . . .	354
8.78.2 成员函数说明 . . . . .	354
8.78.2.1 Touch_() . . . . .	354
8.78.3 类成员变量说明 . . . . .	354
8.78.3.1 show_info_ . . . . .	354
8.79 cyberdog_visual_programming_abilityset::Touch类 参考 . . . . .	355
8.79.1 构造及析构函数说明 . . . . .	357
8.79.1.1 Touch() . . . . .	357
8.79.1.2 ~Touch() . . . . .	357
8.79.2 成员函数说明 . . . . .	357
8.79.2.1 GetData() . . . . .	357
8.79.3 类成员变量说明 . . . . .	358
8.79.3.1 data_ . . . . .	358
8.80 cyberdog_visual_programming_abilityset::Train类 参考 . . . . .	358
8.80.1 构造及析构函数说明 . . . . .	361
8.80.1.1 Train() . . . . .	361
8.80.1.2 ~Train() . . . . .	361
8.80.2 成员函数说明 . . . . .	361
8.80.2.1 GetTrainingWordsSet() . . . . .	361
8.80.2.2 TrainingWordsRecognized() . . . . .	362
8.81 cyberdog_visual_programming_terminal::Interface::Cyberdog::Train结构体 参考 . . . . .	362
8.81.1 构造及析构函数说明 . . . . .	363
8.81.1.1 Train() . . . . .	363
8.81.2 成员函数说明 . . . . .	363
8.81.2.1 Train_() . . . . .	363
8.81.3 类成员变量说明 . . . . .	363
8.81.3.1 show_info_ . . . . .	363
8.82 cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse类 参考 . . . . .	364
8.82.1 详细描述 . . . . .	364
8.82.2 构造及析构函数说明 . . . . .	365
8.82.2.1 TrainingWordsRecognizedMessageResponse() . . . . .	365
8.82.2.2 ~TrainingWordsRecognizedMessageResponse() . . . . .	365
8.82.3 类成员变量说明 . . . . .	365
8.82.3.1 response . . . . .	365
8.82.3.2 state . . . . .	365
8.83 cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse类 参考 . . . . .	366
8.83.1 详细描述 . . . . .	366
8.83.2 构造及析构函数说明 . . . . .	367
8.83.2.1 TrainingWordsRecognizedSeviceResponse() . . . . .	367

---

8.83.2.2 ~TrainingWordsRecognizedSeviceResponse()	367
8.83.3 类成员变量说明	367
8.83.3.1 dictionary	367
8.83.3.2 response	367
8.83.3.3 state	367
8.84 cyberdog_visual_programming_terminal::Interface::Type结构体 参考	368
8.84.1 构造及析构函数说明	368
8.84.1.1 Type()	369
8.84.2 成员函数说明	369
8.84.2.1 Type_()	369
8.84.3 类成员变量说明	369
8.84.3.1 class_	369
8.84.3.2 enum_	369
8.84.3.3 show_info_	369
8.85 cyberdog_visual_programming_abilityset::Ultrasonic类 参考	370
8.85.1 构造及析构函数说明	372
8.85.1.1 Ultrasonic()	372
8.85.1.2 ~Ultrasonic()	372
8.85.2 成员函数说明	372
8.85.2.1 GetData()	372
8.85.3 类成员变量说明	373
8.85.3.1 data_	373
8.86 cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic结构体 参考	373
8.86.1 构造及析构函数说明	373
8.86.1.1 Ultrasonic()	374
8.86.2 成员函数说明	374
8.86.2.1 Ultrasonic_()	374
8.86.3 类成员变量说明	374
8.86.3.1 show_info_	374
8.87 cyberdog_visual_programming_terminal::Interface::Visual结构体 参考	374
8.87.1 构造及析构函数说明	375
8.87.1.1 Visual()	375
8.87.2 成员函数说明	375
8.87.2.1 Visual_()	375
8.87.3 类成员变量说明	375
8.87.3.1 debugger_	375
8.87.3.2 interface_	376
8.87.3.3 show_info_	376
8.88 cyberdog_visual_programming_terminal::Visual类 参考	376
8.88.1 构造及析构函数说明	376
8.88.1.1 Visual()	377
8.88.1.2 ~Visual()	377

8.88.2 类成员变量说明 . . . . .	377
8.88.2.1 cyberdog_ptr . . . . .	377
8.88.2.2 debugger . . . . .	377
8.88.2.3 interface . . . . .	377
8.89 cyberdog_visual_programming_abilityset::Voiceprint类 参考 . . . . .	378
8.89.1 构造及析构函数说明 . . . . .	380
8.89.1.1 Voiceprint() . . . . .	380
8.89.1.2 ~Voiceprint() . . . . .	380
8.89.2 成员函数说明 . . . . .	380
8.89.2.1 Recognized() . . . . .	380
8.90 cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse类 参考 . . . . .	381
8.90.1 详细描述 . . . . .	381
8.90.2 构造及析构函数说明 . . . . .	382
8.90.2.1 VoiceprintRecognizedResponse() . . . . .	382
8.90.2.2 ~VoiceprintRecognizedResponse() . . . . .	382
8.90.3 类成员变量说明 . . . . .	382
8.90.3.1 data . . . . .	382
8.90.3.2 list . . . . .	382
8.90.3.3 state . . . . .	382
<b>9 文件说明 . . . . .</b>	<b>383</b>
9.1 cyberdog_vp/cyberdog_vp/doc/doxygen/index.md 文件参考 . . . . .	383
9.2 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/audio.hpp 文件参考 . . . . .	383
9.2.1 详细描述 . . . . .	384
9.3 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/base.hpp 文件参考 . . . . .	384
9.3.1 详细描述 . . . . .	385
9.4 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/base.hpp 文件参考 . . . . .	386
9.4.1 详细描述 . . . . .	387
9.5 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/bms.hpp 文件参考 . . . . .	387
9.5.1 详细描述 . . . . .	388
9.6 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/common.hpp 文件参考 . . . . .	389
9.6.1 详细描述 . . . . .	395
9.6.2 类说明 . . . . .	396
9.6.2.1 struct cyberdog_visual_programming_abilityset::DefaultAndMaximum . . . . .	396
9.6.2.2 struct cyberdog_visual_programming_abilityset::RPY . . . . .	396
9.6.2.3 struct cyberdog_visual_programming_abilityset::ObstacleMeta . . . . .	397
9.6.2.4 struct cyberdog_visual_programming_abilityset::TofObstacle . . . . .	398
9.6.2.5 struct cyberdog_visual_programming_abilityset::GestureData . . . . .	398
9.6.3 宏定义说明 . . . . .	399
9.6.3.1 FORMAT . . . . .	400
9.6.3.2 ONE_FORMAT . . . . .	400
9.6.3.3 STREAM . . . . .	400

---

9.7 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/common.hpp 文件参考 . . . . .	400
9.7.1 详细描述 . . . . .	403
9.7.2 类说明 . . . . .	403
9.7.2.1 struct cyberdog_visual_programming_engine::BackendMsg . . . . .	403
9.8 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/common.hpp 文件参考 . . . . .	404
9.8.1 详细描述 . . . . .	406
9.9 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/cyberdog.hpp 文件参考 . . . . .	406
9.9.1 详细描述 . . . . .	407
9.10 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/face.hpp 文件参考 . . . . .	408
9.10.1 详细描述 . . . . .	409
9.11 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/follow.hpp 文件参考 . . . . .	410
9.11.1 详细描述 . . . . .	410
9.12 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/gesture.hpp 文件参考 . . . . .	411
9.12.1 详细描述 . . . . .	412
9.13 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/gps.hpp 文件参考 . . . . .	413
9.13.1 详细描述 . . . . .	414
9.14 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/imu.hpp 文件参考 . . . . .	415
9.14.1 详细描述 . . . . .	415
9.15 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/led.hpp 文件参考 . . . . .	416
9.15.1 详细描述 . . . . .	417
9.16 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/lidar.hpp 文件参考 . . . . .	418
9.16.1 详细描述 . . . . .	418
9.17 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/log.hpp 文件参考 . . . . .	419
9.17.1 详细描述 . . . . .	420
9.18 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/motion.hpp 文件参考 . . . . .	420
9.18.1 详细描述 . . . . .	421
9.19 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/navigation.hpp 文件参考 . . . . .	422
9.19.1 详细描述 . . . . .	423
9.20 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/network.hpp 文件参考 . . . . .	424
9.20.1 详细描述 . . . . .	424
9.21 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/odometer.hpp 文件参考 . . . . .	425
9.21.1 详细描述 . . . . .	426
9.22 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/personnel.hpp 文件参考 . . . . .	427
9.22.1 详细描述 . . . . .	428
9.23 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/skeleton.hpp 文件参考 . . . . .	429
9.23.1 详细描述 . . . . .	430
9.24 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/task.hpp 文件参考 . . . . .	430
9.24.1 详细描述 . . . . .	431
9.25 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/task.hpp 文件参考 . . . . .	432
9.25.1 详细描述 . . . . .	433
9.26 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/tof.hpp 文件参考 . . . . .	434
9.26.1 详细描述 . . . . .	434

9.27 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/touch.hpp 文件参考 . . . . .	435
9.27.1 详细描述 . . . . .	436
9.28 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/train.hpp 文件参考 . . . . .	437
9.28.1 详细描述 . . . . .	437
9.29 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ultrasonic.hpp 文件参考 . . . . .	438
9.29.1 详细描述 . . . . .	439
9.30 cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/voiceprint.hpp 文件参考 . . . . .	440
9.31 cyberdog_vp/cyberdog_vp_abilityset/src/api/audio.cpp 文件参考 . . . . .	441
9.32 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/audio.cpp 文件参考 . . . . .	441
9.33 cyberdog_vp/cyberdog_vp_abilityset/src/api/base.cpp 文件参考 . . . . .	441
9.34 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/base.cpp 文件参考 . . . . .	442
9.35 cyberdog_vp/cyberdog_vp_engine/src/base.cpp 文件参考 . . . . .	442
9.36 cyberdog_vp/cyberdog_vp_abilityset/src/api/bms.cpp 文件参考 . . . . .	442
9.37 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/bms.cpp 文件参考 . . . . .	443
9.38 cyberdog_vp/cyberdog_vp_abilityset/src/api/common.cpp 文件参考 . . . . .	443
9.39 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/common.cpp 文件参考 . . . . .	443
9.40 cyberdog_vp/cyberdog_vp_engine/src/common.cpp 文件参考 . . . . .	444
9.41 cyberdog_vp/cyberdog_vp_terminal/src/sdk/common.cpp 文件参考 . . . . .	445
9.42 cyberdog_vp/cyberdog_vp_abilityset/src/api/cyberdog.cpp 文件参考 . . . . .	445
9.43 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/cyberdog.cpp 文件参考 . . . . .	446
9.44 cyberdog_vp/cyberdog_vp_abilityset/src/api/follow.cpp 文件参考 . . . . .	446
9.45 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/follow.cpp 文件参考 . . . . .	447
9.46 cyberdog_vp/cyberdog_vp_abilityset/src/api/gesture.cpp 文件参考 . . . . .	447
9.47 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/gesture.cpp 文件参考 . . . . .	447
9.48 cyberdog_vp/cyberdog_vp_abilityset/src/api/gps.cpp 文件参考 . . . . .	448
9.49 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/gps.cpp 文件参考 . . . . .	448
9.50 cyberdog_vp/cyberdog_vp_abilityset/src/api imu.cpp 文件参考 . . . . .	449
9.51 cyberdog_vp/cyberdog_vp_abilityset/src/sdk imu.cpp 文件参考 . . . . .	449
9.52 cyberdog_vp/cyberdog_vp_abilityset/src/api led.cpp 文件参考 . . . . .	449
9.53 cyberdog_vp/cyberdog_vp_abilityset/src/sdk led.cpp 文件参考 . . . . .	450
9.54 cyberdog_vp/cyberdog_vp_abilityset/src/api lidar.cpp 文件参考 . . . . .	450
9.55 cyberdog_vp/cyberdog_vp_abilityset/src/sdk lidar.cpp 文件参考 . . . . .	451
9.56 cyberdog_vp/cyberdog_vp_abilityset/src/api main.cpp 文件参考 . . . . .	451
9.57 cyberdog_vp/cyberdog_vp_terminal/src/api main.cpp 文件参考 . . . . .	452
9.58 cyberdog_vp/cyberdog_vp_terminal/src/node/main.cpp 文件参考 . . . . .	452
9.58.1 函数说明 . . . . .	453
9.58.1.1 main() . . . . .	453
9.59 cyberdog_vp/cyberdog_vp_abilityset/src/api motion.cpp 文件参考 . . . . .	453
9.60 cyberdog_vp/cyberdog_vp_abilityset/src/sdk motion.cpp 文件参考 . . . . .	453
9.61 cyberdog_vp/cyberdog_vp_abilityset/src/api navigation.cpp 文件参考 . . . . .	454
9.62 cyberdog_vp/cyberdog_vp_abilityset/src/sdk navigation.cpp 文件参考 . . . . .	454
9.63 cyberdog_vp/cyberdog_vp_abilityset/src/api network.cpp 文件参考 . . . . .	454

---

9.64 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/network.cpp 文件参考 . . . . .	455
9.65 cyberdog_vp/cyberdog_vp_abilityset/src/api/odometer.cpp 文件参考 . . . . .	455
9.66 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/odometer.cpp 文件参考 . . . . .	456
9.67 cyberdog_vp/cyberdog_vp_abilityset/src/api/personnel.cpp 文件参考 . . . . .	456
9.68 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/personnel.cpp 文件参考 . . . . .	457
9.69 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_builtin_interfaces.cpp 文件参考 . . . . .	457
9.70 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_geometry_msgs.cpp 文件参考 . . . . .	458
9.71 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_nav_msgs.cpp 文件参考 . . . . .	458
9.72 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_protocol.cpp 文件参考 . . . . .	459
9.73 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_sensor_msgs.cpp 文件参考 . . . . .	460
9.74 cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_std_msgs.cpp 文件参考 . . . . .	460
9.75 cyberdog_vp/cyberdog_vp_abilityset/src/api/skeleton.cpp 文件参考 . . . . .	461
9.76 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/skeleton.cpp 文件参考 . . . . .	461
9.77 cyberdog_vp/cyberdog_vp_abilityset/src/api/task.cpp 文件参考 . . . . .	462
9.78 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/task.cpp 文件参考 . . . . .	462
9.79 cyberdog_vp/cyberdog_vp_engine/src/task.cpp 文件参考 . . . . .	463
9.80 cyberdog_vp/cyberdog_vp_abilityset/src/api/tof.cpp 文件参考 . . . . .	463
9.81 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/tof.cpp 文件参考 . . . . .	463
9.82 cyberdog_vp/cyberdog_vp_abilityset/src/api/touch.cpp 文件参考 . . . . .	464
9.83 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/touch.cpp 文件参考 . . . . .	464
9.84 cyberdog_vp/cyberdog_vp_abilityset/src/api/train.cpp 文件参考 . . . . .	465
9.85 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/train.cpp 文件参考 . . . . .	465
9.86 cyberdog_vp/cyberdog_vp_abilityset/src/api/ultrasonic.cpp 文件参考 . . . . .	466
9.87 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ultrasonic.cpp 文件参考 . . . . .	466
9.88 cyberdog_vp/cyberdog_vp_abilityset/src/api/vpapy.hpp 文件参考 . . . . .	467
9.89 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/face.cpp 文件参考 . . . . .	469
9.90 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/log.cpp 文件参考 . . . . .	469
9.91 cyberdog_vp/cyberdog_vp_abilityset/src/sdk/voiceprint.cpp 文件参考 . . . . .	470
9.92 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/artificial_intelligence.hpp 文件参考 . . . . .	470
9.92.1 详细描述 . . . . .	471
9.93 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/backend_message.hpp 文件参考 . . . . .	473
9.93.1 详细描述 . . . . .	473
9.94 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/frontend_message.hpp 文件参考 . . . . .	474
9.94.1 详细描述 . . . . .	475
9.95 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/fsm.hpp 文件参考 . . . . .	476
9.95.1 详细描述 . . . . .	476
9.96 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/interact.hpp 文件参考 . . . . .	477
9.96.1 详细描述 . . . . .	478
9.97 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/linux_at.hpp 文件参考 . . . . .	479
9.97.1 详细描述 . . . . .	479
9.98 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/linux_cron.hpp 文件参考 . . . . .	480
9.98.1 详细描述 . . . . .	481

---

9.99 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/module.hpp 文件参考 . . . . .	482
9.99.1 详细描述 . . . . .	482
9.100 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/python_interpreter.hpp 文件参考 . . . . .	483
9.100.1 详细描述 . . . . .	484
9.101 cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/slam.hpp 文件参考 . . . . .	485
9.102 cyberdog_vp/cyberdog_vp_engine/src/artificial_intelligence.cpp 文件参考 . . . . .	485
9.103 cyberdog_vp/cyberdog_vp_engine/src/backend_message.cpp 文件参考 . . . . .	486
9.104 cyberdog_vp/cyberdog_vp_engine/src/engine.cpp 文件参考 . . . . .	486
9.104.1 函数说明 . . . . .	486
9.104.1.1 main() . . . . .	486
9.105 cyberdog_vp/cyberdog_vp_engine/src/frontend_message.cpp 文件参考 . . . . .	487
9.106 cyberdog_vp/cyberdog_vp_engine/src/fsm.cpp 文件参考 . . . . .	487
9.107 cyberdog_vp/cyberdog_vp_engine/src/interact.cpp 文件参考 . . . . .	487
9.108 cyberdog_vp/cyberdog_vp_engine/src/linux_at.cpp 文件参考 . . . . .	488
9.109 cyberdog_vp/cyberdog_vp_engine/src/linux_cron.cpp 文件参考 . . . . .	488
9.110 cyberdog_vp/cyberdog_vp_engine/src/module.cpp 文件参考 . . . . .	488
9.111 cyberdog_vp/cyberdog_vp_engine/src/python_interpreter.cpp 文件参考 . . . . .	489
9.112 cyberdog_vp/cyberdog_vp_engine/src/slam.cpp 文件参考 . . . . .	489
9.113 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/debug_abilityset.hpp 文件参考 . . . . .	490
9.113.1 详细描述 . . . . .	491
9.114 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/debug_engine.hpp 文件参考 . . . . .	491
9.114.1 详细描述 . . . . .	492
9.115 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/debugger.hpp 文件参考 . . . . .	493
9.115.1 详细描述 . . . . .	494
9.116 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/interface.hpp 文件参考 . . . . .	495
9.116.1 详细描述 . . . . .	496
9.117 cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/visual.hpp 文件参考 . . . . .	497
9.117.1 详细描述 . . . . .	498
9.118 cyberdog_vp/cyberdog_vp_terminal/src/api/debug_abilityset.cpp 文件参考 . . . . .	498
9.119 cyberdog_vp/cyberdog_vp_terminal/src/sdk/debug_abilityset.cpp 文件参考 . . . . .	499
9.120 cyberdog_vp/cyberdog_vp_terminal/src/api/debug_engine.cpp 文件参考 . . . . .	499
9.121 cyberdog_vp/cyberdog_vp_terminal/src/sdk/debug_engine.cpp 文件参考 . . . . .	500
9.122 cyberdog_vp/cyberdog_vp_terminal/src/api/debugger.cpp 文件参考 . . . . .	500
9.123 cyberdog_vp/cyberdog_vp_terminal/src/sdk/debugger.cpp 文件参考 . . . . .	501
9.124 cyberdog_vp/cyberdog_vp_terminal/src/api/interface.cpp 文件参考 . . . . .	501
9.125 cyberdog_vp/cyberdog_vp_terminal/src/sdk/interface.cpp 文件参考 . . . . .	502
9.126 cyberdog_vp/cyberdog_vp_terminal/src/api/visual.cpp 文件参考 . . . . .	502
9.127 cyberdog_vp/cyberdog_vp_terminal/src/sdk/visual.cpp 文件参考 . . . . .	503
9.128 cyberdog_vp/cyberdog_vp_terminal/src/api/vptpy.hpp 文件参考 . . . . .	503



# Chapter 1

## cyberdog\_vp

### 1.1 许可证 (License)

Copyright (c) 2023 Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.  
Licensed under the Apache License, Version 2.0 (the 'License');  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at  
<http://www.apache.org/licenses/LICENSE-2.0>  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an 'AS IS' BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

### 1.2 cyberdog\_vp 设计文档 - 中文版

#### 1.2.1 1. 修订

项目	软件版本	协议版本	修订日期	修订人员	备注
铁蛋能力集SDK	V1.1.0.0	V1.0.0.0	2023-02-06	尚子涵	无
可视化编程引擎	V1.1.0.0	V1.0.0.0	2023-02-06	尚子涵	无
可视化编程终端	V1.1.0.0	V1.0.0.0	2023-02-06	尚子涵	无

#### 1.2.2 2. 概述

可视化编程，亦即可视化程序设计：以“所见即所得”的编程思想为原则，力图实现编程工作的可视化，即随时可以看到结果，程序与结果的调整同步。可视化编程是与传统的编程方式相比而言的，这里的“可视”，指的是无须编程，仅通过直观的操作方式即可完成界面的设计工作，在此通过移动设备进行编程，机器人负责执行程序。

#### 1.2.3 3. 设计

##### 1.2.3.1 3.1. 操作流程设计

###### 1.2.3.1.1 列表界面设计



如上图所示，进入图像化编程界面的时候，最初是没有任务以及模块的，此时需要点击新建按钮进行新建。

#### 1.2.3.1.2 3.1.2 编辑界面设计



如上图所示，当点击新建按钮进行新建任务或模块的时候，会进入到一个新的页面，该页面内会默认存在一个“开始运行程序”的起始块，后续编程需要连接在该块下方。

#### 1.2.3.1.3 3.1.3 原子功能块界面设计



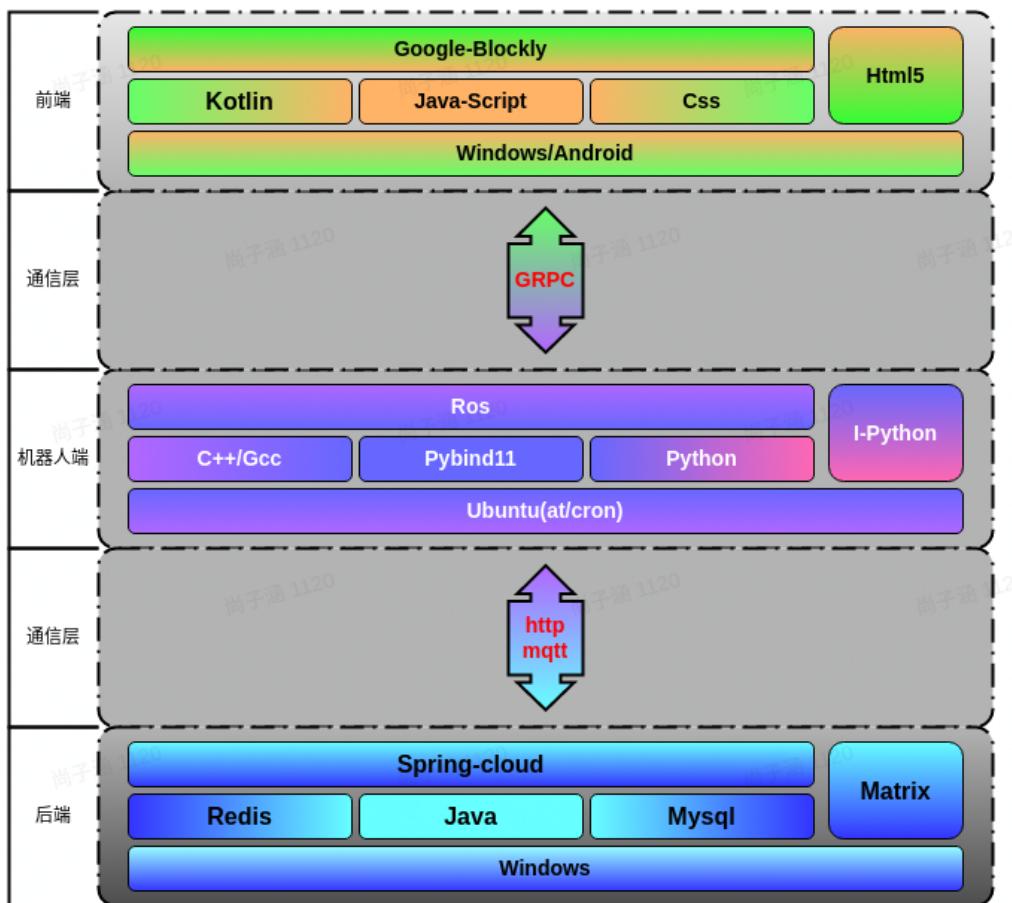
如上图所示，在任务或模块编辑界面内，左侧会呈现出原子块的分类，当需要某类原子块的时候，点击分类就可以在里面选取所需的原子块并拖拽到右侧的空白画布上进行编程了，当编程结束就可以点击保存按钮进行保存或者点击调试按钮进行调试了。

#### 1.2.3.1.4 3.1.4 列表界面展示设计



如上图所示，当任务或模块编辑完后，如果保存编程内容，就可以在可视化编程列表界面看到我们保存的任务或模块了。在此可以开启某个想要运行的定时任务。

### 1.2.3.2 3.2 技术架构

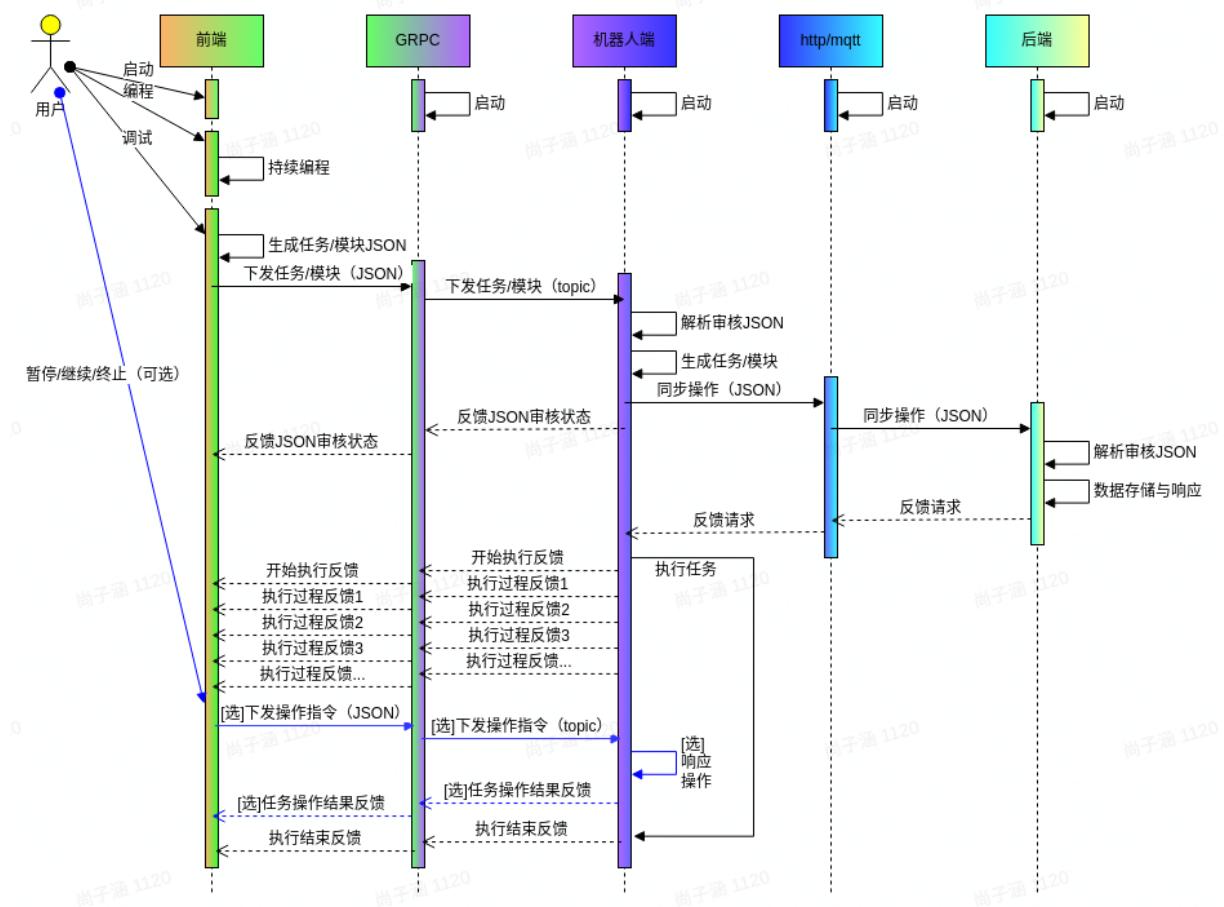


如上图所示，可视化编程由三大模块构成，其技术架构如下：

1. 前端：主要技术有kotlin、JS、Css、H5和谷歌Blockly。
2. 机器人端：主要技术有C++、Pybind11、Python、I-Python和ROS。
3. 后端：主要技术有Java、Redis、mysql、matrix和Spring-cloud。

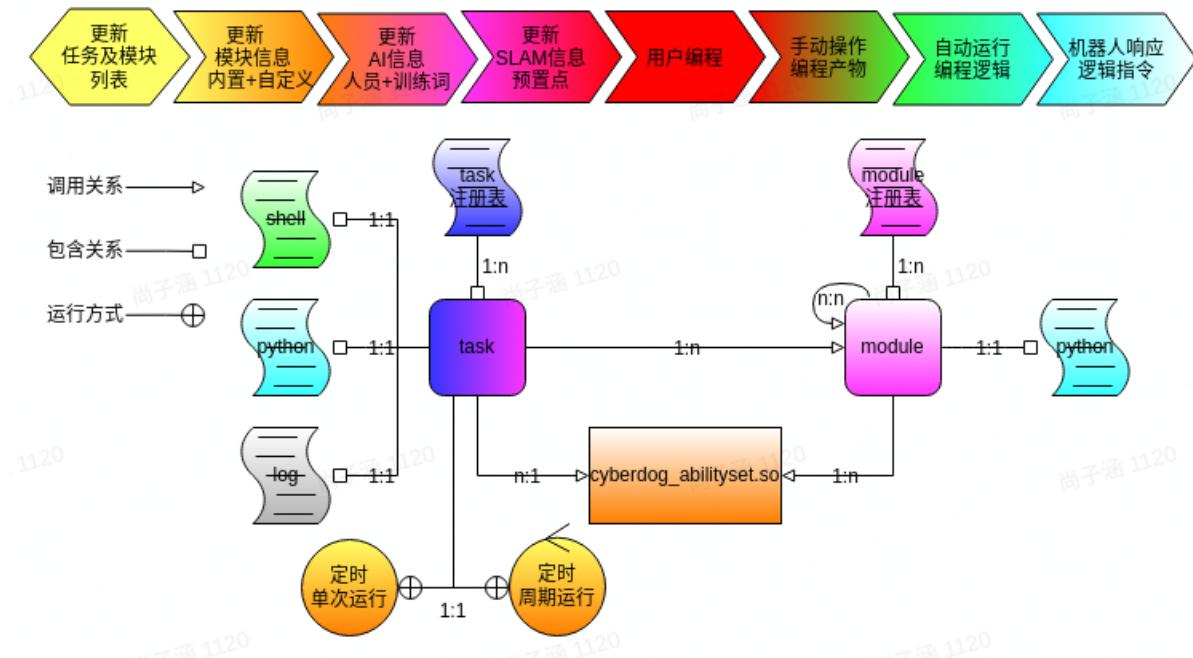
前端和机器人端通过grpc通信；机器人端和后端通过http或mqtt通信。

### 1.2.3.3 3.3 交互时序



如上图所示，是可视化编程三大模块之间最简单的交互说明时序图。

### 1.2.3.4 3.4 文件架构



如上图所示，是可视化编程所涉及的文件架构：

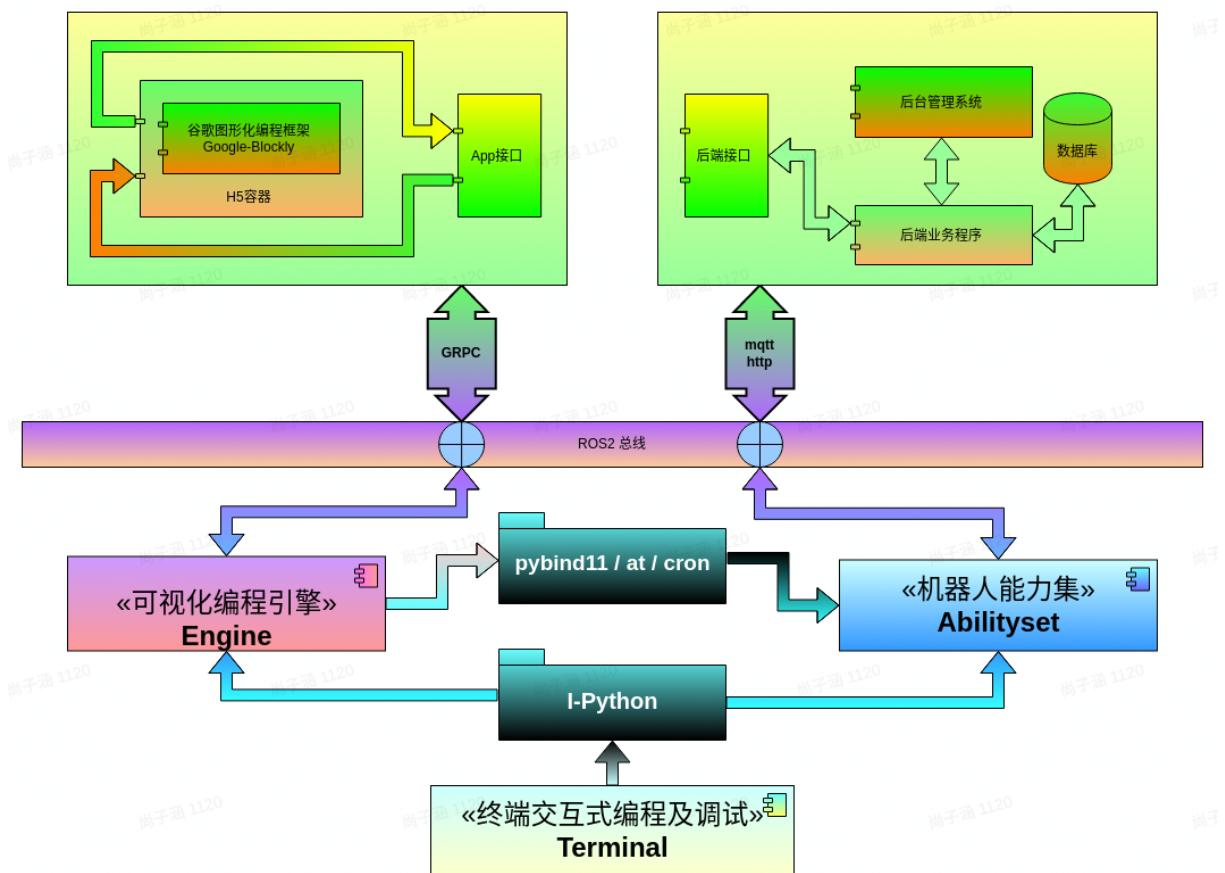
### 1. 模块：

- 1个模块注册表包含多个模块；
- 1个模块对应1个python文件；
- 1个模块可以调用多个其他模块；
- 1个模块可以被多个其他模块或任务调用；

### 2. 任务：

- 1个任务注册表包含多个任务；
- 1个任务对应1个python文件，1个shell文件和1个日志文件；
- 1个任务可以调用多个模块；
- 1个任务执行方式有两种：
  - 定时单次运行；
  - 定时周期执行

### 1.2.3.5 3.5 模块架构

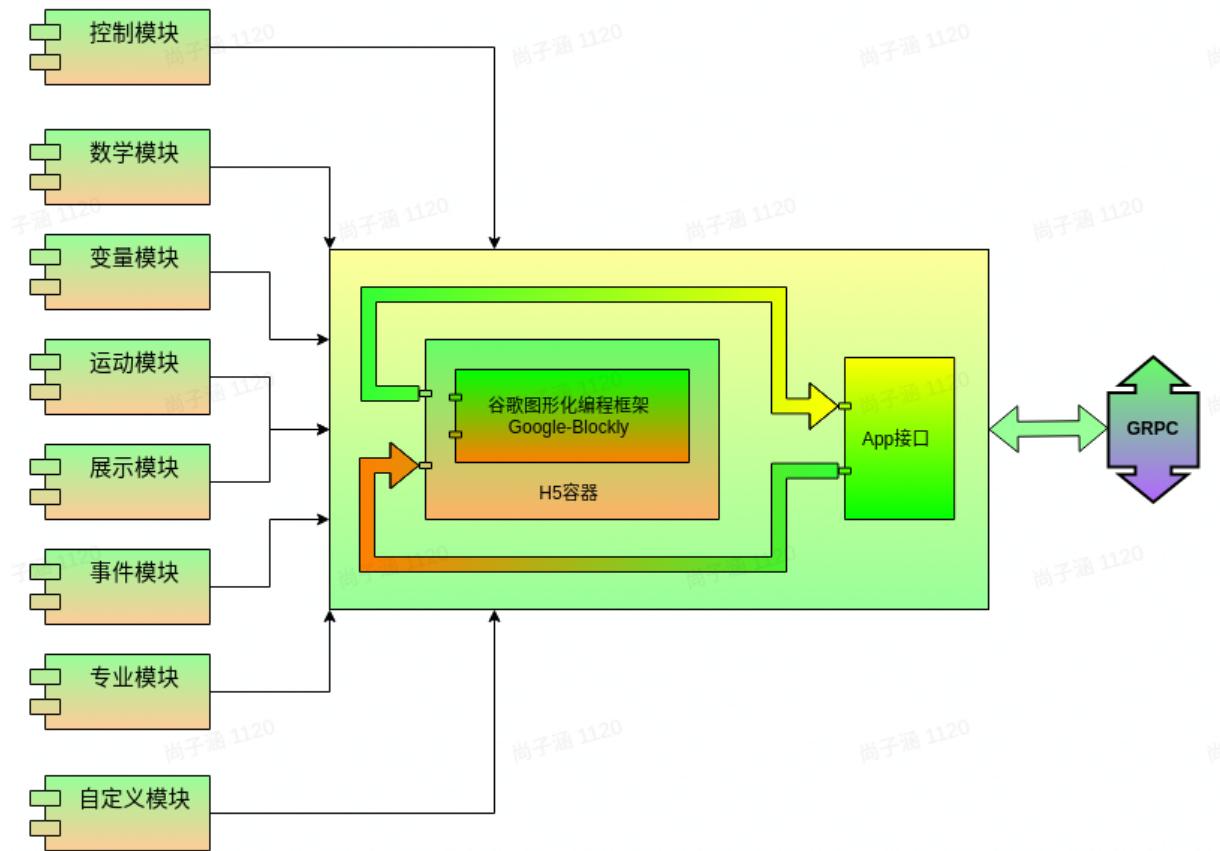


如上图所示，是可视化编程所涉及的模块架构：

1. 前端：通过H5容器集成骨骼图形化编程框架到APP。
2. 后端：由web端后台管理系统、后台业务程序和数据库组成。
3. 机器人端：由可视化编程引擎、机器人能力集和终端交互式编程及调试模块组成。

基于I-Python实现交互式编程，基于pybind11实现C++和python的交叉编译及C++到python的接口映射。

### 1.2.3.5.1 3.5.1 前端模块架构



如上图所示，可视化编程前端模块架构：

#### 1. 编程基础类别：

- 控制类：基础控制语法；
- 数学类：四则运算；
- 变量类：基础变量；

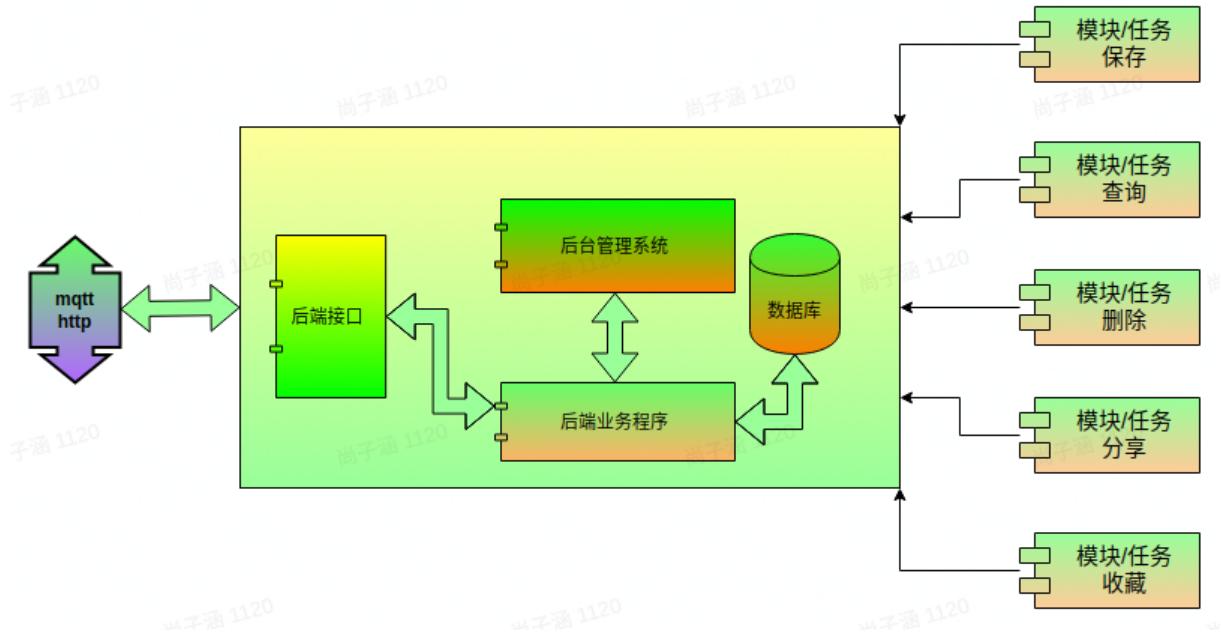
#### 2. 机器人能力类别：

- 运动类：基础运动；
- 展示类：语音交互；
- 事件类：AI识别；
- 专业类：专业运动接口；

#### 3. 编程依赖类别：

- 自定义模块类。

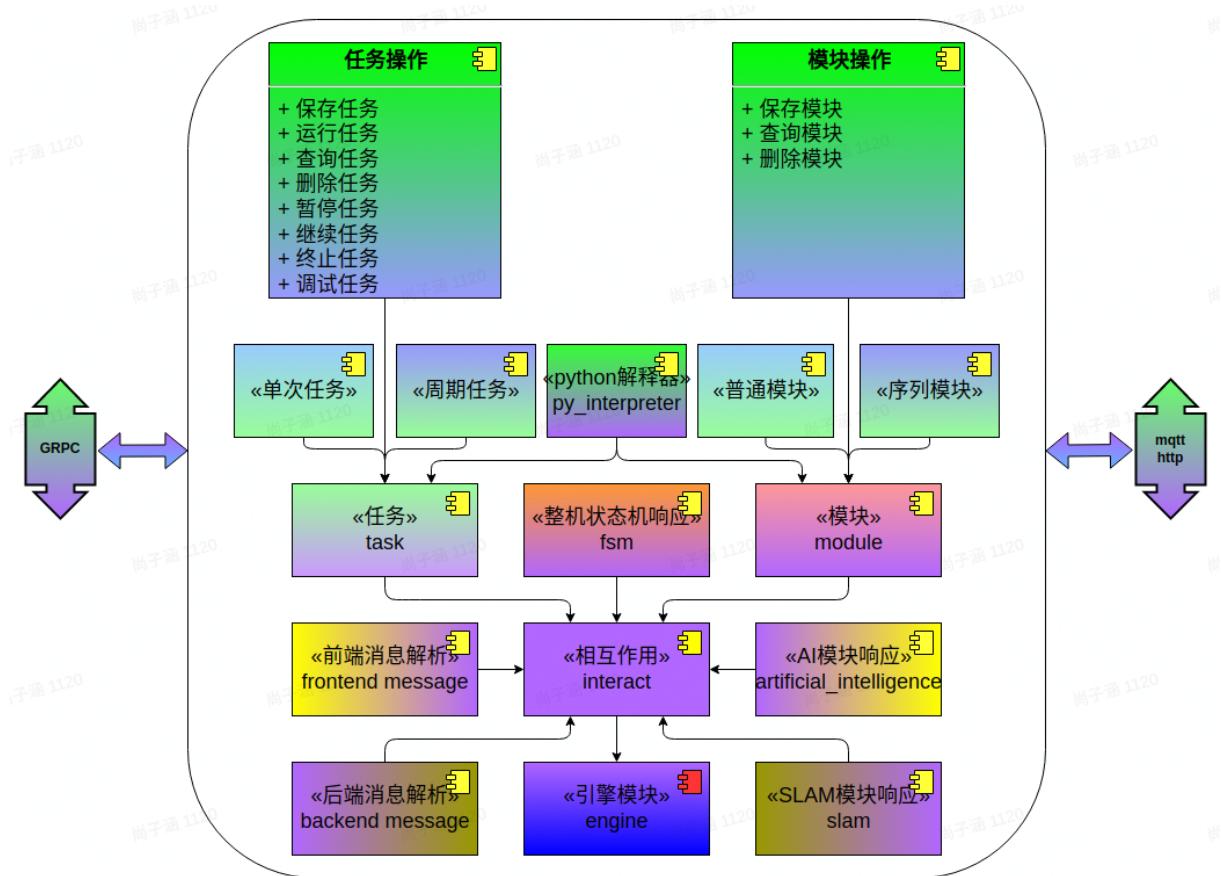
### 1.2.3.5.2 3.5.2 后端模块架构



如上图所示，可视化编程后端模块架构：

1. 后端提供一个web端的后台管理系统；
2. 后端业务程序API提供5种能力：
  - 保存模块或任务；
  - 查询模块或任务；
  - 删除模块或任务；
  - 分享模块或任务；
  - 收藏模块或任务。

### 1.2.3.5.3 3.5.3 机器人端-引擎模块架构



如上图所示，可视化编程引擎模块架构：

### 1. 核心功能：

- 解析及响应前端消息；
- 请求及处理后端消息；
- 构造及注册模块或任务；
- 控制任务运行。

### 2. 任务操作及模块操作参见任务及模块状态流转。

#### 1.2.3.5.4 3.5.3.1 机器人端-模块操作及状态流转设计

A	B	C	D
1	状态流转关系表		
当前状态	当前操作请求		
	查询(inquiry)	保存(save)	删除(delete)
空 (null)	空状态	错误状态 正常状态	非法操作
错误(error)	错误状态	错误状态 正常状态	空状态
正常(normal)	正常状态	错误状态 正常状态	空状态 正常状态

如上表所示，三种模块状态分别在三种模块操作下的流转关系尽收眼底，现对三种模块操作及三种模块状态进行说明如下：

- 三种模块操作

1. 保存任务：构建当前id的模块，若模块id已存在则覆盖，并审核任务语法是否合规，反馈操作结果。
2. 查询任务：查询当前id对应的模块，反馈操作结果。
3. 删除任务：审核当前id对应的模块是否可以删除，若可以则删除当前id对应的模块，反馈操作结果。

- 三种模块状态

1. 空状态：该状态是指没有当前模块的任何状态，本质为当前模块不存在，也就是说任何未记录的模块的状态均为空状态。
2. 错误状态：该状态是指当前模块不符合语法规则，也就是说当前模块无法运行只能再次编辑。
3. 正常状态：该状态是指当前模块符合语法规则，也就是说当前模块可以调用。

### 1.2.3.5.5 3.5.3.2 机器人端-任务操作及状态流转设计

	A	B	C	D	E	F	G	H	I	J	K
1											
状态流转关系表											
2	当前状态	查询(inquiry)	保存(save)	删除(delete)	调试(debug)	运行(run)	暂停(suspend)	继续(recover)	终止(shutdown)	启动(start)	停止(stop)
3	空(null)	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态
4	错误(error)	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态	错误状态
5	等待运行(wait_run)	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态	等待运行状态
6	运行运行(run_run)	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态	运行等待状态
7	暂停(suspend)	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态	暂停状态
8	继续(recover)	继续状态	继续状态	继续状态	继续状态	继续状态	继续状态	继续状态	继续状态	继续状态	继续状态
9	终止(shutdown)	终止状态	终止状态	终止状态	终止状态	终止状态	终止状态	终止状态	终止状态	终止状态	终止状态
10											

如上表所示，七种任务状态分别在八种任务操作下的流转关系尽收眼底，现对八种任务操作及七种任务状态进行说明如下：

- 八种任务操作

1. 保存任务：构建当前id的任务，若任务id已存在则覆盖，并审核任务语法是否合规，反馈操作结果。
2. 运行任务：运行当前id对应的任务，若任务id对应的任务不存在或语法状态错误则不执行，反馈操作结果。
3. 查询任务：查询当前id对应的任务，反馈操作结果。
4. 删除任务：删除当前id对应的任务，反馈操作结果。
5. 暂停任务：暂停当前id对应的任务，反馈操作结果。
6. 继续任务：继续当前id对应的任务，反馈操作结果。
7. 终止任务：终止当前id对应的任务，反馈操作结果。
8. 调试任务：以当前调试id为基础，保存、审核及运行当前帧携带的逻辑，反馈操作结果。

- 七种任务状态

1. 空状态：该状态是指没有当前任务的任何状态，本质为当前任务不存在，也就是说任何未记录的任务的状态均为空状态。
2. 错误状态：该状态是指当前任务不符合语法规则，本质为当前任务不合规，也就是说当前任务无法运行只能再次编辑。
3. 等待运行状态：该状态是指当前任务可以运行但尚未加入任务注册表中，本质为当前任务语法规则，也就是说当前任务处于等待用户确认运行状态。

4. 运行等待状态：该状态是指当前任务已加入任务注册表中，但尚未满足运行条件，本质为当前任务语法规则，且正在等待运行条件满足后即刻运行的状态。
  5. 运行状态：该状态是指当前任务已满足执行条件，正在执行内部逻辑的状态。
  6. 暂停状态：该状态是指当前任务处于暂停执行的状态，此时任务进程任在，可以是断点暂停或用户手动暂停正在执行的任务。
  7. 终止状态：该状态是指当前任务被终止执行，此时不存在任务进程，可以是正常执行结束或被迫终止任务。
- 注意：当前的任务状态是基于任务内编程逻辑考虑，不做任务执行约束条件考虑（由调用方考虑）：
  - 当某条任务内编程逻辑审核通过，该任务状态即为等待运行（wait\_run）状态，若该任务执行约束条件为定时单次执行时，需要调用方根据定时约束判断当前任务的约束时间是否已过期，当未过期时再开启任务。当然，也可以不用管，直接下发开启请求，但是会被视为非法请求，操作会失败，对用户操作体验不友好。

#### 1.2.3.5.6 3.5.3.3 机器人端-状态机各模式下操作约束设计

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	当前模式	用户对任务、模块、AI能力及SLAM能力的查询操作	用户对任务或模块的操作	用户对任务的操作	用户对任务的操作	用户对任务的操作	用户对任务的操作	用户对任务的操作	用户对任务的操作	用户对任务的操作	任务自动触发	
3	模式名称	模式标识	查询(inquiry)	保存(save)	删除(delete)	调试(debug)	运行(run)	暂停(suspend)	继续(recover)	终止(shutdown)	启动(start)	停止(stop)
4	未初始化	Uninitialized	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作
5	资源加载模式	SetUp	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作
6	资源释放模式	TearDown	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作
7	自检模式	SekfCheck	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
8	活动模式	Active	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
9	静默模式	DeActive	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
10	低电量模式	Protected	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
11	低功耗模式	LowPower	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
12	远程升级模式	OTA	合法操作	非法操作	非法操作	非法操作	合法操作	合法操作	合法操作	合法操作	合法操作	合法操作
13	错误模式	Error	合法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作	非法操作

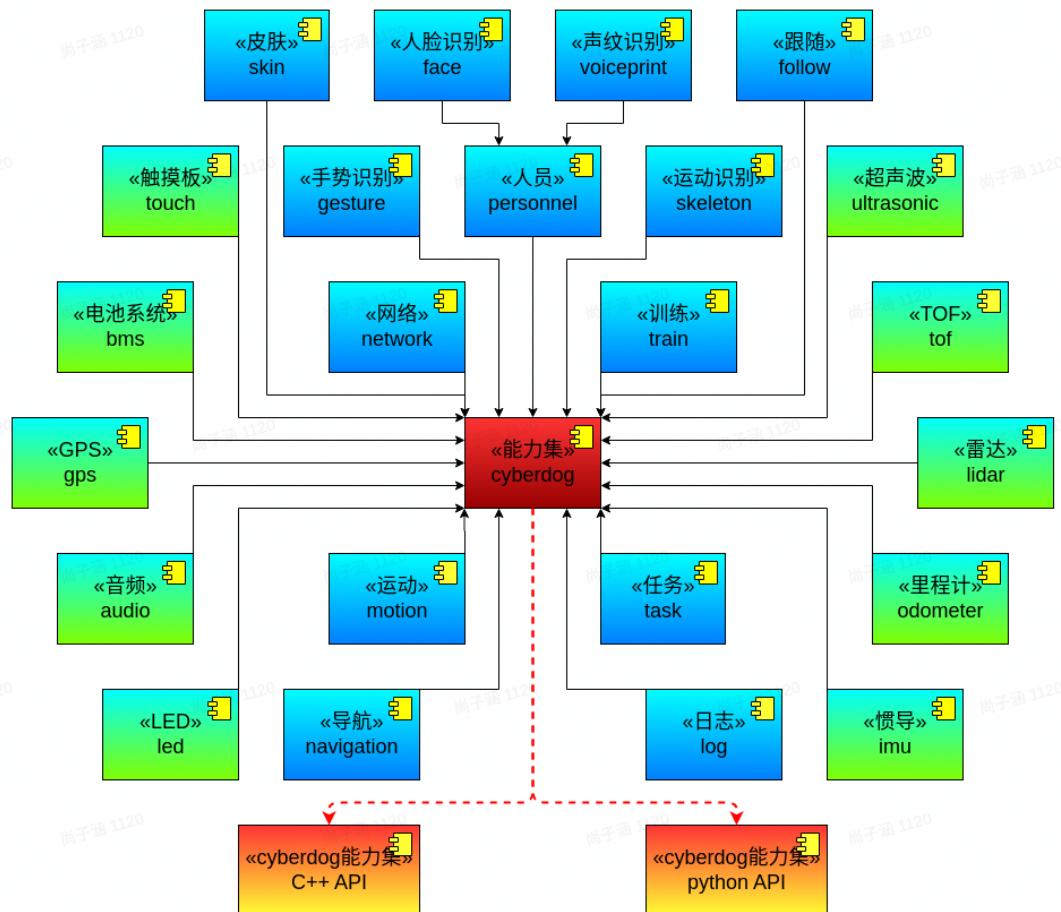
如上表所示，十种机器人模式下用户对任务或模块的八种操作，用户对AI能力的一种操作以及任务自行触发的两种操作约束条件尽收眼底，各模式下的约束指标主要考虑如下：

1. Uninitialized、SetUp、TearDown三种模式下，机器人各功能模块均处于无法正常工作的状态，故而限制所有操作；
2. SekfCheck、OTA以及Error模式下，机器人各功能模块均处于封闭状态，故而限制所有可能导致图形化编程新增进程的操作，只允许现有进程自动停止（stop）或用户手动暂停（suspend）或终止（shutdown）现有进程，允许查询（inquiry）是想为用户提供展示任务、模块及AI信息，为用户的暂停（suspend）或终止提供基础，同时也支持二次编辑任务或模块的功能，提升用户和机器人互动过程中的异步体验（不会因自检阻塞用户编程），该思想同样适用于Active、DeActive、Protected、LowPower、OTA以及Error模式；
3. Active模式下，机器人各功能模块均处于能够正常工作的阶段，故而开放所有操作；
4. Protected模式下，机器人大部分功能模块均处于能够正常工作的阶段，故而开放所有操作，该模式下受限功能如下：
  - 运动模块：除站立、趴下外的所有结果指令；
  - LED模块：BMS会抢占LED设备。
5. DeActive以及LowPower模式下，机器人各功能模块均处于休眠状态，查询（inquiry）、停止（stop）及暂停（suspend）或终止（shutdown）约束考虑和SekfCheck模式一样，对于保存（save）和删除（delete）操作的考虑和查询一样，也是想为用户提供展示任务、模块及AI信息，为用户的暂停（suspend）或终止提供基础，同时也支持二次编辑任务或模块的功能，提升用户和机器人互动过程中的异步体验（不会因自检阻塞用户编程）。

注意：

- 低电量模式：
  - 进入：
    1. 电量低于20, 自动进入;
  - 退出：
    1. 电量大于等于20, 自动退出;
- 低功耗模式
  - 进入：
    1. 电量小于5时自动进入低功耗;
    2. 趴下超过30s, 进入低功耗;
    3. 唤醒退出低功耗模式后, 若30s内未进行运动控制, 则再次进入低功耗模式;
  - 退出：
    1. 语音唤醒: “铁蛋、铁蛋”;
    2. app端点击退出低功耗;
    3. 双击狗头退出低功耗;

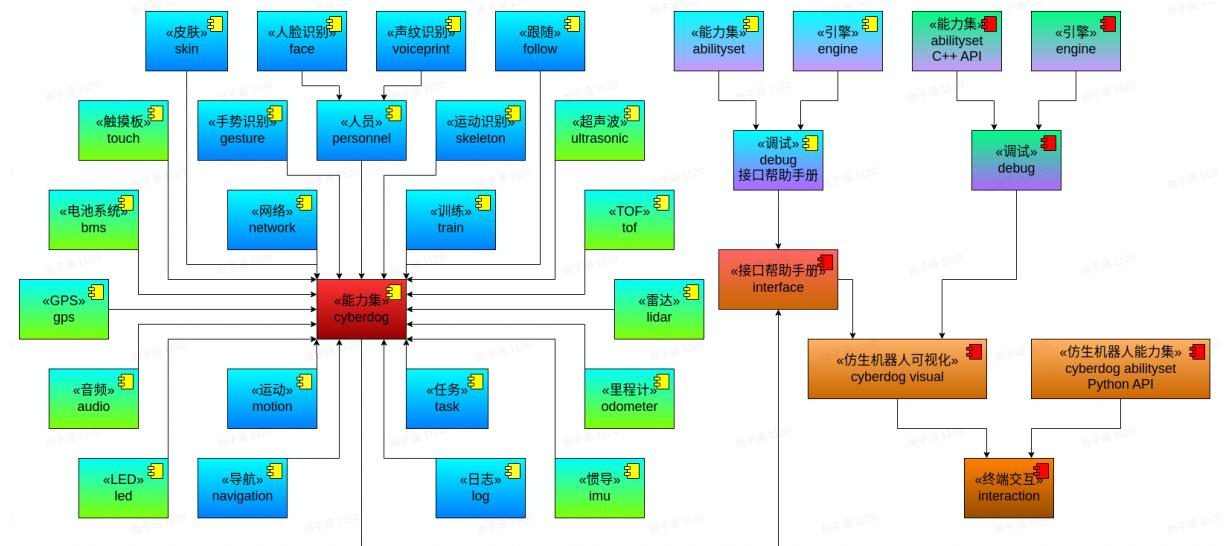
#### 1.2.3.5.7 3.5.4 机器人端-铁蛋能力集模块架构



如上图所示，可视化编程铁蛋能力集模块架构：

- 10种设备（传感器）：触摸板、电池、超声波、雷达、里程计、语音、led、IMU、TOF、GPS。
- 13种功能：运动、导航、任务控制、日志记录、网络状态、训练词、手势识别、人员信息、运动识别（基于骨骼点）、人脸识别、声纹识别、皮肤控制及跟随功能。其中，皮肤控制和跟随功能预留，等待适配业务场景。
- 2种API：基于同一个机器人能力集实现动态库，同时开放C++ API和Python API。

### 1.2.3.5.8 3.5.5 机器人端-终端交互式编程与调试模块架构



如上图所示，可视化编程铁蛋能力集模块架构：

- 2种能力：
  1. 终端下交互式编程：基于Python API实现终端编程，用于高级玩家，同时也是研发过程中脱离APP进行Python API的调试工具。
  2. 终端下调试工具：主要调试图形化编程引擎和机器人能力集C++ API，同时提供了终端下的API帮助文档功能。

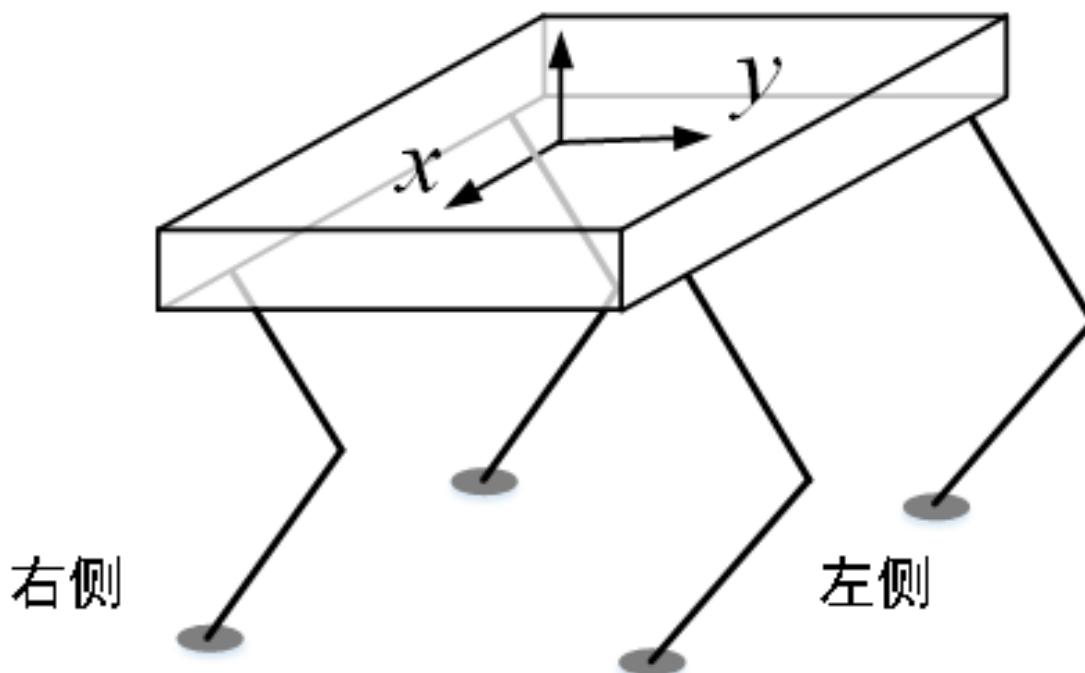
## 1.2.4 4 附录

### 1.2.4.1 4.1 自定义动作说明

MIT开源代码 **Cheetah Software**，包括其他开源软件，对于四足机器人的步态定义都是针对周期步态，通过定义周期，占空比和相位差来完全确定一个步态。步态可以理解为机器人的每条腿支撑和摆动相位随时间变化的关系，周期步态是指每条腿的支撑相和摆动相是周期变化的，比如四足机器人对角步态走路就是典型的周期步态。定义的步态配合相应参数，包括躯干的轨迹，落足点的位置，抬脚的高度等，就形成了不同的动作，简单来说就是“步态定义+步态参数=动作”。

周期步态可以实现周期的动作，比如走路、奔跑等，但是舞蹈动作往往没有一个严格的周期，所以就需要一种定义非周期步态并编辑步态参数的方法。而且，所有开源软件都是在代码中定义的步态，这样修改性就比较差，每次修改都需要编译代码，所以我们的目的是开发一种基于图形化的非周期步态定义和编辑方法，不仅可以提高开发效率，也方便用户去修改或新建自己的步态，生成自己的舞蹈动作，降低用户的开发门槛，提高机器人的可玩性，让每个用户都可以开发自己喜欢的步态。

机器人坐标系如图 1所示，四条腿分别以FL (Front-left) , FR (Front-right) , RL (Rear-left) , RR (Rear-right) 来命名。



步态表示每条腿支撑和摆动相位随时间变化的关系，为了方便修改，我们将步态定义内容以文件的形式存储。

每一个步态块表示一个步态单元，每个步态单元内包括支撑状态和持续时间。对于支撑状态1表示支撑，0表示摆动，四个数字分别代表FR, FL, RR, RL四条腿；持续时间表示支撑状态持续N个单位时间，单位时间是30毫秒。对应的时间为 $T_i=N_i \times 30\text{ms}$  ( $i=1\dots 8$ )，步态约束格式参见可视化编程-接口文档中的MotionSequenceGait类型。

步伐参数文件用于描述步态运行中的所有参数，每一个步伐块代表一个参数单元，包括躯干的速度、位置和姿态，摆动腿的落足点，以及摆动腿的抬腿高度，另外还需要指明当前地面与足底的摩擦系数以免打滑，最后就是这个步态的持续时间，步伐约束格式参见可视化编程-接口文档中的MotionSequence↔Pace类型。

需要注意的是，步态定义和步态参数的持续时间是相关的，但并不是一一对应的。一般只有在支撑相位变化时才需要规定落足点。

完成步态定义和步态参数后，就开始了动作的执行。这里需要将定义的步态和相应的参数转化为每一个时刻计算机器人所有的期望状态。

对于躯干，期望的状态包括位移、姿态、线速度和角速度，在步态参数文件中，如果某个方向指定了期望速度，那么这个方向的期望位移就根据速度的积分获得。以Moonwalk第一个参数块为例，在x方向指定了速度为 $vx$ ，持续时间为 $T_1$ ，那么x方向期望的位移就 $vx \times T_1$ ，body\_pos\_des中x方向的期望位移0.0就不会被使用。如果速度为0.0，那么就与以body\_pos\_des中定义的位移为准，期望速度就为位移除以时间，其他的方向与x方向逻辑相同。在实际使用中，平移运动一般通过速度定义，因为用户比较关心机器人往那个方向运动，对于具体的运动距离并没有特别要求；旋转运动一般通过定义姿态，因为用户更关心旋转的角度而不是速度。

对于摆动腿，需要计算它的摆动轨迹，我们使用三阶贝塞尔曲线，只需要指定曲线初始的位置速度和终点的位置速度就可以计算出每个时刻的位置。这里说的x, y, z方向都有各自的贝塞尔轨迹，描述的是对应方向的位移与时间的关系。摆动腿的起点就是支撑状态的终点，不需要用户指定，用户只需要指定每条腿的摆动终点即可，也就是落足点。其中x, y方向只有一个终点，计算一条贝塞尔曲线即可，z方向由于有抬腿高度的需求，所以包括抬起和落下，需要两条贝塞尔曲线。由于摆动腿抬起和落地时刻相对于地面的速度都是零，所以通过给定的落足点和抬脚高度，就可以计算出摆动腿的轨迹了。

有了躯干的期望状态和摆动腿的期望状态，又已知支撑腿相对地面不能运动，这样就可以完全确定此时刻机器人的期望状态。利用MPC+WBC的开源控制算法或者其他运动控制算法都可以实现当前状态下动作的执行。每一时刻都会检测动作执行是否完成，完成的标志就是执行时间到达用户定义的总时间，如果执行完成，机器人就进入站立状态等待下一个指令；否则就时刻增加 $dt$ ，继续执行动作。

## Chapter 2

# Bug 列表

文件 [artificial.intelligence.hpp](#)

模块操作尚待调试

模块操作尚待调试

文件 [common.hpp](#)

确保通用文档的正确性。

文件 [debug.abilityset.hpp](#)

确保各个接口的可用及稳定性。

文件 [debug.engine.hpp](#)

确保各个接口的可用及稳定性。

文件 [debugger.hpp](#)

确保各个接口的可用及稳定性。

文件 [follow.hpp](#)

确保各个接口的可用及稳定性。

文件 [fsm.hpp](#)

模块操作尚待调试

文件 [imu.hpp](#)

确保各个接口的可用及稳定性。

文件 [interface.hpp](#)

确保接口文档的正确性。

文件 [module.hpp](#)

模块操作尚待调试

文件 [motion.hpp](#)

确保各个接口的可用及稳定性。

文件 [navigation.hpp](#)

确保各个接口的可用及稳定性。

文件 [network.hpp](#)

确保各个接口的可用及稳定性。

文件 [odometer.hpp](#)

确保各个接口的可用及稳定性。

**文件 task.hpp**

任务操作尚待调试  
任务依赖模块的筛查尚待调试  
依赖模块的任务的编程尚待调试

## Chapter 3

# 命名空间索引

### 3.1 命名空间列表

这里列出了所有命名空间定义,附带简要说明:

<a href="#">cyberdog_visual_programming_abilityset</a> . . . . .	29
<a href="#">cyberdog_visual_programming_abilityset.py</a> . . . . .	62
<a href="#">cyberdog_visual_programming_engine</a> . . . . .	98
<a href="#">cyberdog_visual_programming_terminal</a> . . . . .	107
<a href="#">cyberdog_visual_programming_terminal.py</a> . . . . .	110



## Chapter 4

# 继承关系索引

### 4.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset . . . . .	125
cyberdog_visual_programming_engine::ArtificialIntelligence . . . . .	126
cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio . . . . .	128
cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse . . . . .	135
cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse . . . . .	137
cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse . . . . .	139
cyberdog_visual_programming_engine::BackendMessage . . . . .	141
cyberdog_visual_programming_engine::BackendMsg . . . . .	98
cyberdog_visual_programming_abilityset::Base . . . . .	145
cyberdog_visual_programming_abilityset::Audio . . . . .	129
cyberdog_visual_programming_abilityset::Bms . . . . .	174
cyberdog_visual_programming_abilityset::Cyberdog . . . . .	184
cyberdog_visual_programming_abilityset::Face . . . . .	200
cyberdog_visual_programming_abilityset::Follow . . . . .	207
cyberdog_visual_programming_abilityset::Gesture . . . . .	222
cyberdog_visual_programming_abilityset::Gps . . . . .	234
cyberdog_visual_programming_abilityset::Imu . . . . .	238
cyberdog_visual_programming_abilityset::Led . . . . .	249
cyberdog_visual_programming_abilityset::Lidar . . . . .	256
cyberdog_visual_programming_abilityset::Motion . . . . .	278
cyberdog_visual_programming_abilityset::Navigation . . . . .	296
cyberdog_visual_programming_abilityset::Network . . . . .	309
cyberdog_visual_programming_abilityset::Odometer . . . . .	311
cyberdog_visual_programming_abilityset::Personnel . . . . .	318
cyberdog_visual_programming_abilityset::Skeleton . . . . .	326
cyberdog_visual_programming_abilityset::Task . . . . .	341
cyberdog_visual_programming_abilityset::Tof . . . . .	348
cyberdog_visual_programming_abilityset::Touch . . . . .	355
cyberdog_visual_programming_abilityset::Train . . . . .	358
cyberdog_visual_programming_abilityset::Ultrasonic . . . . .	370
cyberdog_visual_programming_abilityset::Voiceprint . . . . .	378
cyberdog_visual_programming_engine::Base . . . . .	163
cyberdog_visual_programming_engine::Module . . . . .	275
cyberdog_visual_programming_engine::Task . . . . .	337
cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms . . . . .	173

cyberdog_visual_programming_terminal::Interface::Type::Class . . . . .	178
cyberdog_visual_programming_terminal::Interface::Cyberdog . . . . .	179
cyberdog_visual_programming_terminal::DebugAbilityset . . . . .	190
cyberdog_visual_programming_terminal::DebugEngine . . . . .	191
cyberdog_visual_programming_terminal::Debugger . . . . .	193
cyberdog_visual_programming_terminal::Interface::Visual::Debugger . . . . .	195
cyberdog_visual_programming_abilityset::DefaultAndMaximum . . . . .	29
cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine . . . . .	197
cyberdog_visual_programming_terminal::Interface::Type::Enum . . . . .	198
cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse . . . . .	203
cyberdog_visual_programming_abilityset::FaceSeviceResponse . . . . .	206
cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow . . . . .	212
cyberdog_visual_programming_engine::FrontendMessage . . . . .	213
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture . . . . .	228
cyberdog_visual_programming_abilityset::GestureData . . . . .	29
cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse . . . . .	229
cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse . . . . .	232
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps . . . . .	237
cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu . . . . .	242
cyberdog_visual_programming_engine::Interact . . . . .	244
cyberdog_visual_programming_terminal::Interface::Visual::Interface . . . . .	245
cyberdog_visual_programming_terminal::Interface . . . . .	247
cyberdog_visual_programming_terminal::Interface::Cyberdog::Led . . . . .	253
cyberdog_visual_programming_abilityset::LedSeviceResponse . . . . .	254
cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar . . . . .	259
cyberdog_visual_programming_engine::LinuxAt . . . . .	260
cyberdog_visual_programming_engine::LinuxCron . . . . .	263
cyberdog_visual_programming_abilityset::Log . . . . .	266
MachineActuator	
cyberdog_visual_programming_engine::Fsm . . . . .	216
cyberdog_visual_programming_abilityset::MapPresetSeviceResponse . . . . .	272
cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion . . . . .	277
cyberdog_visual_programming_abilityset::MotionResultServiceResponse . . . . .	289
cyberdog_visual_programming_abilityset::MotionSequence . . . . .	291
cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse . . . . .	293
cyberdog_visual_programming_abilityset::MotionServoCmdResponse . . . . .	295
cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation . . . . .	304
cyberdog_visual_programming_abilityset::NavigationActionResponse . . . . .	306
cyberdog_visual_programming_terminal::Interface::Cyberdog::Network . . . . .	307
cyberdog_visual_programming_abilityset::ObstacleMeta . . . . .	29
cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer . . . . .	315
cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel . . . . .	316
cyberdog_visual_programming_engine::PythonInterpreter . . . . .	322
cyberdog_visual_programming_abilityset::RPY . . . . .	29
cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton . . . . .	324
cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse . . . . .	331
cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse . . . . .	333
cyberdog_visual_programming_engine::Slam . . . . .	334
cyberdog_visual_programming_abilityset::State . . . . .	336
cyberdog_visual_programming_terminal::Interface::Cyberdog::Task . . . . .	340
cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof . . . . .	347
cyberdog_visual_programming_abilityset::TofObstacle . . . . .	29
cyberdog_visual_programming_abilityset::TofPayload . . . . .	352
cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch . . . . .	353
cyberdog_visual_programming_terminal::Interface::Cyberdog::Train . . . . .	362
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse . . . . .	364
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse . . . . .	366
cyberdog_visual_programming_terminal::Interface::Type . . . . .	368

cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic . . . . .	373
cyberdog_visual_programming_terminal::Interface::Visual . . . . .	374
cyberdog_visual_programming_terminal::Visual . . . . .	376
cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse . . . . .	381



# Chapter 5

## 类索引

### 5.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明：

cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset	125
cyberdog_visual_programming_engine::ArtificialIntelligence	126
cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio	128
cyberdog_visual_programming_abilityset::Audio	129
cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse	135
cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse	137
cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse	139
cyberdog_visual_programming_engine::BackendMessage	141
cyberdog_visual_programming_abilityset::Base	145
cyberdog_visual_programming_engine::Base	163
cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms	173
cyberdog_visual_programming_abilityset::Bms	174
cyberdog_visual_programming_terminal::Interface::Type::Class	178
cyberdog_visual_programming_terminal::Interface::Cyberdog	179
cyberdog_visual_programming_abilityset::Cyberdog	184
cyberdog_visual_programming_terminal::DebugAbilityset	190
cyberdog_visual_programming_terminal::DebugEngine	191
cyberdog_visual_programming_terminal::Debugger	193
cyberdog_visual_programming_terminal::Interface::Visual::Debugger	195
cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine	197
cyberdog_visual_programming_terminal::Interface::Type::Enum	198
cyberdog_visual_programming_abilityset::Face	200
cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse	203
cyberdog_visual_programming_abilityset::FaceSeviceResponse	206
cyberdog_visual_programming_abilityset::Follow	207
cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow	212
cyberdog_visual_programming_engine::FrontendMessage	213
cyberdog_visual_programming_engine::Fsm	216
cyberdog_visual_programming_abilityset::Gesture	222
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture	228
cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse	229
cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse	232
cyberdog_visual_programming_abilityset::Gps	234
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps	237
cyberdog_visual_programming_abilityset::Imu	238

cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu . . . . .	242
cyberdog_visual_programming_engine::Interact . . . . .	244
cyberdog_visual_programming_terminal::Interface::Visual::Interface . . . . .	245
cyberdog_visual_programming_terminal::Interface . . . . .	247
cyberdog_visual_programming_abilityset::Led . . . . .	249
cyberdog_visual_programming_terminal::Interface::Cyberdog::Led . . . . .	253
cyberdog_visual_programming_abilityset::LedSeviceResponse . . . . .	254
cyberdog_visual_programming_abilityset::Lidar . . . . .	256
cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar . . . . .	259
cyberdog_visual_programming_engine::LinuxAt . . . . .	260
cyberdog_visual_programming_engine::LinuxCron . . . . .	263
cyberdog_visual_programming_abilityset::Log . . . . .	266
cyberdog_visual_programming_abilityset::MapPresetSeviceResponse . . . . .	272
cyberdog_visual_programming_engine::Module . . . . .	275
cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion . . . . .	277
cyberdog_visual_programming_abilityset::Motion . . . . .	278
cyberdog_visual_programming_abilityset::MotionResultServiceResponse . . . . .	289
cyberdog_visual_programming_abilityset::MotionSequence . . . . .	291
cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse . . . . .	293
cyberdog_visual_programming_abilityset::MotionServoCmdResponse . . . . .	295
cyberdog_visual_programming_abilityset::Navigation . . . . .	296
cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation . . . . .	304
cyberdog_visual_programming_abilityset::NavigationActionResponse . . . . .	306
cyberdog_visual_programming_terminal::Interface::Cyberdog::Network . . . . .	307
cyberdog_visual_programming_abilityset::Network . . . . .	309
cyberdog_visual_programming_abilityset::Odometer . . . . .	311
cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer . . . . .	315
cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel . . . . .	316
cyberdog_visual_programming_abilityset::Personnel . . . . .	318
cyberdog_visual_programming_engine::PythonInterpreter . . . . .	322
cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton . . . . .	324
cyberdog_visual_programming_abilityset::Skeleton . . . . .	326
cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse . . . . .	331
cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse . . . . .	333
cyberdog_visual_programming_engine::Slam . . . . .	334
cyberdog_visual_programming_abilityset::State . . . . .	336
cyberdog_visual_programming_engine::Task . . . . .	337
cyberdog_visual_programming_terminal::Interface::Cyberdog::Task . . . . .	340
cyberdog_visual_programming_abilityset::Task . . . . .	341
cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof . . . . .	347
cyberdog_visual_programming_abilityset::Tof . . . . .	348
cyberdog_visual_programming_abilityset::TofPayload . . . . .	352
cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch . . . . .	353
cyberdog_visual_programming_abilityset::Touch . . . . .	355
cyberdog_visual_programming_abilityset::Train . . . . .	358
cyberdog_visual_programming_terminal::Interface::Cyberdog::Train . . . . .	362
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse . . . . .	364
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse . . . . .	366
cyberdog_visual_programming_terminal::Interface::Type . . . . .	368
cyberdog_visual_programming_abilityset::Ultrasonic . . . . .	370
cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic . . . . .	373
cyberdog_visual_programming_terminal::Interface::Visual . . . . .	374
cyberdog_visual_programming_terminal::Visual . . . . .	376
cyberdog_visual_programming_abilityset::Voiceprint . . . . .	378
cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse . . . . .	381

# Chapter 6

## 文件索引

### 6.1 文件列表

这里列出了所有文件，并附带简要说明：

cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">audio.hpp</a>	383
语音模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">base.hpp</a>	384
可视化编程SDK的基础模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">bms.hpp</a>	387
电池管理系统模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">common.hpp</a>	389
可视化编程参数模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">cyberdog.hpp</a>	406
铁蛋模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">face.hpp</a>	408
人脸识别模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">follow.hpp</a>	410
跟随模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">gesture.hpp</a>	411
手势识别模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">gps.hpp</a>	413
GPS模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">imu.hpp</a>	415
Imu模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">led.hpp</a>	416
Led模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">lidar.hpp</a>	418
雷达模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">log.hpp</a>	419
可视化编程SDK的日志模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">motion.hpp</a>	420
运动模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">navigation.hpp</a>	422
导航模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">network.hpp</a>	424
网络模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">odometer.hpp</a>	425
里程计模块。	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">personnel.hpp</a>	427
人员识别模块。	

cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">skeleton.hpp</a>	429
骨骼（点）识别模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">task.hpp</a>	430
任务模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">tof.hpp</a>	434
激光测距模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">touch.hpp</a>	435
触摸板模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">train.hpp</a>	437
训练模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">ultrasonic.hpp</a>	438
超声波模块。 . . . . .	
cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/ <a href="#">voiceprint.hpp</a>	440
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">audio.cpp</a>	441
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">base.cpp</a>	441
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">bms.cpp</a>	442
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">common.cpp</a>	443
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">cyberdog.cpp</a>	445
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">follow.cpp</a>	446
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">gesture.cpp</a>	447
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">gps.cpp</a>	448
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">imu.cpp</a>	449
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">led.cpp</a>	449
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">lidar.cpp</a>	450
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">main.cpp</a>	451
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">motion.cpp</a>	453
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">navigation.cpp</a>	454
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">network.cpp</a>	454
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">odometer.cpp</a>	455
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">personnel.cpp</a>	456
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_builtin_interfaces.cpp</a>	457
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_geometry_msgs.cpp</a>	458
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_nav_msgs.cpp</a>	458
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_protocol.cpp</a>	459
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_sensor_msgs.cpp</a>	460
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ros_std_msgs.cpp</a>	460
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">skeleton.cpp</a>	461
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">task.cpp</a>	462
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">tof.cpp</a>	463
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">touch.cpp</a>	464
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">train.cpp</a>	465
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">ultrasonic.cpp</a>	466
cyberdog_vp/cyberdog_vp_abilityset/src/api/ <a href="#">vpavy.hpp</a>	467
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">audio.cpp</a>	441
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">base.cpp</a>	442
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">bms.cpp</a>	443
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">common.cpp</a>	443
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">cyberdog.cpp</a>	446
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">face.cpp</a>	469
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">follow.cpp</a>	447
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">gesture.cpp</a>	447
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">gps.cpp</a>	448
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">imu.cpp</a>	449
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">led.cpp</a>	450
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">lidar.cpp</a>	451
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">log.cpp</a>	469
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">motion.cpp</a>	453
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">navigation.cpp</a>	454

cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">network.cpp</a>	455
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">odometer.cpp</a>	456
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">personnel.cpp</a>	457
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">skeleton.cpp</a>	461
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">task.cpp</a>	462
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">tof.cpp</a>	463
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">touch.cpp</a>	464
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">train.cpp</a>	465
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">ultrasonic.cpp</a>	466
cyberdog_vp/cyberdog_vp_abilityset/src/sdk/ <a href="#">voiceprint.cpp</a>	470
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">artificial_intelligence.hpp</a> AI模块。	470
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">backend_message.hpp</a> 消息模块。	473
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">base.hpp</a> 可视化编程引擎的基础模块。	386
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">common.hpp</a> 可视化编程引擎的共享模块。	400
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">frontend_message.hpp</a> 消息模块。	474
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">fsm.hpp</a> 状态机模块。	476
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">interact.hpp</a> 可视化编程引擎相互作用模块。	477
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">linux_at.hpp</a> 单次任务模块。	479
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">linux_cron.hpp</a> 周期任务模块。	480
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">module.hpp</a> 模块编程模块。	482
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">python_interpreter.hpp</a> Python 解释器模块。	483
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">slam.hpp</a>	485
cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/ <a href="#">task.hpp</a> 任务编程模块。	432
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">artificial_intelligence.cpp</a>	485
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">backend_message.cpp</a>	486
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">base.cpp</a>	442
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">common.cpp</a>	444
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">engine.cpp</a>	486
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">frontend_message.cpp</a>	487
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">fsm.cpp</a>	487
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">interact.cpp</a>	487
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">linux_at.cpp</a>	488
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">linux_cron.cpp</a>	488
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">module.cpp</a>	488
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">python_interpreter.cpp</a>	489
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">slam.cpp</a>	489
cyberdog_vp/cyberdog_vp_engine/src/ <a href="#">task.cpp</a>	463
cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">common.hpp</a> 通用模块。	404
cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">debug_abilityset.hpp</a> 能力集调试器模块。	490
cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">debug_engine.hpp</a> 引擎调试器模块。	491
cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">debugger.hpp</a> 调试器模块。	493

cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">interface.hpp</a>	495
接口模块。	495
cyberdog_vp/cyberdog_vp_terminal/include/cyberdog_vp_terminal/ <a href="#">visual.hpp</a>	497
可视化模块。	497
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">debug_abilityset.cpp</a>	498
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">debug_engine.cpp</a>	499
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">debugger.cpp</a>	500
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">interface.cpp</a>	501
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">main.cpp</a>	452
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">visual.cpp</a>	502
cyberdog_vp/cyberdog_vp_terminal/src/api/ <a href="#">vptpy.hpp</a>	503
cyberdog_vp/cyberdog_vp_terminal/src/node/ <a href="#">main.cpp</a>	452
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">common.cpp</a>	445
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">debug_abilityset.cpp</a>	499
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">debug_engine.cpp</a>	500
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">debugger.cpp</a>	501
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">interface.cpp</a>	502
cyberdog_vp/cyberdog_vp_terminal/src/sdk/ <a href="#">visual.cpp</a>	503

# Chapter 7

## 命名空间文档

### 7.1 `cyberdog_visual_programming_abilityset` 命名空间参考

#### 类

- class `Audio`
- class `AudioGetVolumeSeviceResponse`
- class `AudioPlaySeviceResponse`
- class `AudioSetVolumeSeviceResponse`
- class `Base`
- class `Bms`
- class `Cyberdog`
- struct `DefaultAndMaximum`
- class `Face`
- class `FaceRecognizedSeviceResponse`
- class `FaceSeviceResponse`
- class `Follow`
- class `Gesture`
- struct `GestureData`
- class `GestureRecognizedMessageResponse`
- class `GestureRecognizedSeviceResponse`
- class `Gps`
- class `Imu`
- class `Led`
- class `LedSeviceResponse`
- class `Lidar`
- class `Log`
- class `MapPresetSeviceResponse`
- class `Motion`
- struct **`MotionParams`**
- class `MotionResultServiceResponse`
- class `MotionSequence`
- class `MotionSequenceServiceResponse`
- class `MotionServoCmdResponse`
- class `Navigation`
- class `NavigationActionResponse`
- class `Network`
- struct `ObstacleMeta`

- class [Odometer](#)
- class [Personnel](#)
- struct [RPY](#)
- class [Skeleton](#)
- class [SkeletonRecognizedMessageResponse](#)
- class [SkeletonRecognizedSeviceResponse](#)
- class [State](#)
- class [Task](#)
- class [Tof](#)
- struct [TofObstacle](#)
- class [TofPayload](#)
- class [Touch](#)
- class [Train](#)
- class [TrainingWordsRecognizedMessageResponse](#)
- class [TrainingWordsRecognizedSeviceResponse](#)
- class [Ultrasonic](#)
- class [Voiceprint](#)
- class [VoiceprintRecognizedResponse](#)

## 类型定义

- using [CyberdogJson](#) = cyberdog::common::CyberdogJson
- using [CyberdogToml](#) = cyberdog::common::CyberdogToml
- using [MsgHeader](#) = std\_msgs::msg::Header
- using [MsgString](#) = std\_msgs::msg::String
- using [MsgBool](#) = std\_msgs::msg::Bool
- using [MsgLaserScan](#) = sensor\_msgs::msg::LaserScan
- using [MsgImu](#) = sensor\_msgs::msg::Imu
- using [MsgRange](#) = sensor\_msgs::msg::Range
- using [MsgOdometry](#) = nav\_msgs::msg::Odometry
- using [MsgPose](#) = geometry\_msgs::msg::Pose
- using [MsgPoseStamped](#) = geometry\_msgs::msg::PoseStamped
- using [MsgPoint](#) = geometry\_msgs::msg::Point
- using [MsgMotionID](#) = protocol::msg::MotionID
- using [MsgMotionServoCmd](#) = protocol::msg::MotionServoCmd
- using [MsgMotionStatus](#) = protocol::msg::MotionStatus
- using [MsgSingleTofPayload](#) = protocol::msg::SingleTofPayload
- using [MsgHeadTofPayload](#) = protocol::msg::HeadTofPayload
- using [MsgRearTofPayload](#) = protocol::msg::RearTofPayload
- using [MsgGpsPayload](#) = protocol::msg::GpsPayload
- using [MsgAudioPlayExtend](#) = protocol::msg::AudioPlayExtend
- using [MsgAudioPlay](#) = protocol::msg::AudioPlay
- using [MsgConnectorStatus](#) = protocol::msg::ConnectorStatus
- using [MsgMotionServoResponse](#) = protocol::msg::MotionServoResponse
- using [MsgMotionSequenceGait](#) = protocol::msg::MotionSequenceGait
- using [MsgMotionSequencePace](#) = protocol::msg::MotionSequencePace
- using [MsgTouchStatus](#) = protocol::msg::TouchStatus
- using [MsgBmsStatus](#) = protocol::msg::BmsStatus
- using [MsgPersonnel](#) = protocol::msg::UserInformation
- using [MsgFaceRes](#) = protocol::msg::FaceRecognitionResult
- using [MsgGesture](#) = protocol::msg::GestureActionResult
- using [MsgSport](#) = protocol::msg::SportCountsResult
- using [MsgPreset](#) = protocol::msg::Label
- using [MsgTrainingWords](#) = protocol::msg::TrainPlan

- using `MsgAlgoStatus` = protocol::msg::AlgoTaskStatus
- using `MsgVisualProgrammingOperate` = protocol::msg::VisualProgrammingOperate
- using `SrvVisualProgrammingOperate` = protocol::srv::VisualProgrammingOperate
- using `SrvTrainingWords` = protocol::srv::TrainPlanAll
- using `SrvPersonnel` = protocol::srv::AllUserSearch
- using `SrvFaceRec` = protocol::srv::FaceRec
- using `SrvGesture` = protocol::srv::GestureActionControl
- using `SrvSport` = protocol::srv::SportManager
- using `SrvMotionSequenceShow` = protocol::srv::MotionSequenceShow
- using `SrvMotionResultCmd` = protocol::srv::MotionResultCmd
- using `SrvLedExecute` = protocol::srv::LedExecute
- using `SrvAudioTextPlay` = protocol::srv::AudioTextPlay
- using `SrvAudioGetVolume` = protocol::srv::AudioVolumeGet
- using `SrvAudioSetVolume` = protocol::srv::AudioVolumeSet
- using `SrvGetPreset` = protocol::srv::GetMapLabel
- using `SrvCancelNavigation` = protocol::srv::StopAlgoTask
- using `SrvSetBool` = std::srvs::srv::SetBool
- using `ActNavigation` = protocol::action::Navigation

## 枚举

- enum `Processor` { `task` = 0 }
- enum `MotionId` {
   
`emergency_stop` = 0, `get_down` = 101, `resume_standing` = 111, `servo_standing` = 112,
 `back_flip` = 121, `front_flip` = 122, `bow` = 123, `roll_left` = 124,
 `walk_the_dog` = 125, `jump_stair` = 126, `right_somersault` = 127, `left_somersault` = 128,
 `run_and_jump_front_flip` = 129, `jump3d_left90deg` = 130, `jump3d_right90deg` = 131, `jump3d_forward60cm` = 132,
 `jump3d_forward30cm` = 133, `jump3d_left20cm` = 134, `jump3d_right20cm` = 135, `jump3d_up30cm` = 136,
 `jump3d_down_stair` = 137, `roll_right` = 138, `dance_collection` = 140, `hold_left_hand` = 141,
 `hold_right_hand` = 142, `sit_down` = 143, `butt_circle` = 144, `head_circle` = 145,
 `stretch_the_body` = 146, `shake_ass_left` = 148, `shake_ass_right` = 149, `shake_ass_from_side_to_side` = 150,
 `ballet` = 151, `space_walk` = 152, `front_leg_jumping` = 153, `hind_leg_jumping` = 154,
 `lift_the_left_leg_and_nod` = 155, `lift_the_right_leg_and_nod` = 156, `left_front_right_back_legs_apart` = 157,
 `right_front_left_back_legs_apart` = 158,
 `walk_nodding` = 159, `walking_with_divergence_and_adduction_alternately`, `nodding_in_place` = 161,
 `front_legs_jump_back_and_forth` = 162,
 `hind_legs_jump_back_and_forth` = 163, `alternately_front_leg_lift` = 164, `alternately_hind_leg_lift` = 165,
 `jump_collection` = 166,
 `stretching_left_and_right` = 167, `jump_forward_and_backward` = 168, `step_left_and_right` = 169, `right_leg_back_and_forth_stepping` = 170,
 `left_leg_back_and_forth_stepping` = 171, `squat_down_on_all_fours` = 173, `push_ups` = 174, `bow_to_each_other` = 175,
 `absolute_force_control_attitude` = 201, `relatively_force_control_attitude` = 202, `absolute_position_control_attitude` = 211, `relatively_position_control_attitude` = 212,
 `relatively_position_control_attitude_insert_frame_1`, `relatively_position_control_attitude_insert_frame_2`, `jump_back_and_forth` = 301, `small_jump_walking` = 302,
 `trot_walking` = 303, `automatic_frequency_conversion_walking` = 304, `run_fast_walking` = 305, `sequence_custom` = 400,
 `illegal_motion_id` = 999
 }
- enum `StateCode` {
   
`invalid` = -1, `success` = 0, `error_base` = 5800, `fail` = 5801,
 `uninitialized` = 5802, `fsm_does_not_allow` = 5803, `module_status_error` = 5804, `network_error` = 5805,
 `no_operation_authority` = 5806, `timeout` = 5807, `command_does_not_support` = 5808, `self_test_failed` = 5809,
 `parameter_is_invalid` = 5810, `status_is_busy` = 5811, `hardware_error` = 5812, `command_waiting_execute` = 5821,
 }

```

spin_future_interrupted = 5831, spin_future_timeout = 5832, no_data_update = 5841, service_client_interrupted
= 5851,
service_appear_timeout = 5852, service_request_interrupted = 5853, service_request_rejected = 5854,
service_request_timeout = 5855,
action_request_timeout = 5861, action_request_rejected = 5862, action_result_timeout = 5863 }
• enum RPYType { ROLL = 0, PITCH, YAW }
• enum LedConstraint {
    target_head, target_tail, target_mini, effect_line_on,
    effect_line_blink, effect_line_blink_fast, effect_line_breath, effect_line_breath_fast,
    effect_line_one_by_one, effect_line_one_by_one_fast, effect_line_back_and_forth, effect_line_trailing_race,
    system_effect_line_off, system_effect_line_red_on, system_effect_line_red_blink, system_effect_line_red_blink_fast,
    system_effect_line_red_breath, system_effect_line_red_breath_fast, system_effect_line_red_one_by_one,
    system_effect_line_red_one_by_one_fast,
    system_effect_line_blue_on, system_effect_line_blue_blink, system_effect_line_blue_blink_fast, system_effect_line_blue_breath,
    system_effect_line_blue_breath_fast, system_effect_line_blue_one_by_one, system_effect_line_blue_one_by_one_fast,
    system_effect_line_yellow_on,
    system_effect_line_yellow_blink, system_effect_line_yellow_blink_fast, system_effect_line_yellow_breath,
    system_effect_line_yellow_breath_fast,
    system_effect_line_yellow_one_by_one, system_effect_line_yellow_one_by_one_fast, effect_mini_circular_breath,
    effect_mini_circular_ring,
    system_effect_mini_off, system_effect_mini_rectangle_color, system_effect_mini_centre_color, system_effect_mini_three_circular,
    system_effect_mini_one_by_one }
• enum GestureType {
    no_gesture = 0, pulling_hand_or_two_fingers_in = 1, pushing_hand_or_two_fingers_away = 2, sliding_hand_or_two_fingers_up
    = 3,
    sliding_hand_or_two_fingers_down = 4, sliding_hand_or_two_fingers_left = 5, sliding_hand_or_two_fingers_right =
    6, stop_sign = 7,
    thumb_up = 8, zooming_in_with_hand_or_two_fingers = 9, zooming_out_with_hand_or_two_fingers = 10,
    thumb_down = 11,
    id_upper_limit = 12 }
• enum SkeletonType {
    SPORT_SQUAT = 1, SPORT_HIGHKNEES, SPORT_SITUP, SPORT_PRESSUP,
    SPORT_PLANK, SPORT_JUMPJACK }
• enum TimeMode { Ms1970 = 0, _Y_M_D_H_M_S, STANDARD, DETAILED }

```

## 函数

- template<typename ... Args>
 std::string Format (const char \*format, Args... args)
- geometry\_msgs::msg::Quaternion RPY2Orientation (const double, const double, const double)
- geometry\_msgs::msg::Vector3 Orientation2RPY (const double, const double, const double, const double)
- std::string GetTime (int nowModo=0)
- double Angle2Radian (const double \_degree)
- double Radian2Angle (const double \_rad)
- uint64\_t GetTimeNs ()
- bool JudgeToml (const std::string &)
- bool GetWorkspace (std::string &)
- bool Timeout (const uint64\_t &\_old\_ns, uint64\_t .timeout\_ms=3000)
- std::string int2binary (const int)
- std::string covariance36 (const std::array< double, 36 > &, const std::string, const int line\_size=6)
- std::string covariance9 (const std::array< double, 9 > &, const std::string, const int line\_size=3)
- std::string sequenceGaitVector (const std::vector< MsgMotionSequenceGait > &, const std::string)
- std::string sequencePaceVector (const std::vector< MsgMotionSequencePace > &, const std::string)
- std::string msgFaceResVector (const std::vector< MsgFaceRes > &, const std::string)
- std::string msgPersonnelVector (const std::vector< MsgPersonnel > &, const std::string)

- std::string `msgFaceResMap` (const std::map< std::string, `MsgFaceRes` > &, const std::string)
- std::string `msgPresetVector` (const std::vector< `MsgPreset` > &, const std::string)
- std::string `msgPresetMap` (const std::map< std::string, `MsgPreset` > &, const std::string)
- std::string `msgTrainingWordsVector` (const std::vector< `MsgTrainingWords` > &, const std::string)
- std::string `msgTrainingWordsMap` (const std::map< std::string, `MsgTrainingWords` > &, const std::string)
- std::string `stringVector` (const std::vector< std::string > &)
- std::string `intVectorToString` (const std::vector< int > &)

## 变量

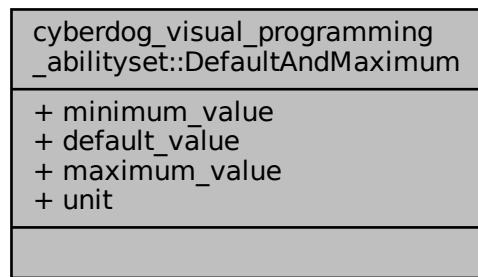
- std::mutex `bms_data_cvm_`
- std::condition\_variable `bms_data_cv_`
- std::mutex `gesture_recognition_state_cvm_`
- std::condition\_variable `gesture_recognition_state_cv_`
- std::mutex `gps_data_cvm_`
- std::condition\_variable `gps_data_cv_`
- std::mutex `imu_data_cvm_`
- std::condition\_variable `imu_data_cv_`
- std::mutex `lidar_data_cvm_`
- std::condition\_variable `lidar_data_cv_`
- std::mutex `compensation_frame_mtx_`
- std::mutex `odometer_data_cvm_`
- std::condition\_variable `odometer_data_cv_`
- std::mutex `skeleton_recognition_state_cvm_`
- std::condition\_variable `skeleton_recognition_state_cv_`
- std::mutex `task_state_cvm_`
- std::condition\_variable `task_state_cv_`
- std::mutex `tof_data_cvm_`
- std::condition\_variable `tof_data_cv_`
- std::mutex `touch_data_cvm_`
- std::condition\_variable `touch_data_cv_`
- std::mutex `training_words_state_cvm_`
- std::condition\_variable `training_words_state_cv_`
- std::mutex `ultrasonic_data_cvm_`
- std::condition\_variable `ultrasonic_data_cv_`

### 7.1.1 类说明

#### 7.1.1.1 struct `cyberdog_visual_programming_abilityset::DefaultAndMaximum`

参数约束:默认值和最大值

cyberdog\_visual\_programming\_abilityset::DefaultAndMaximum 的协作图:



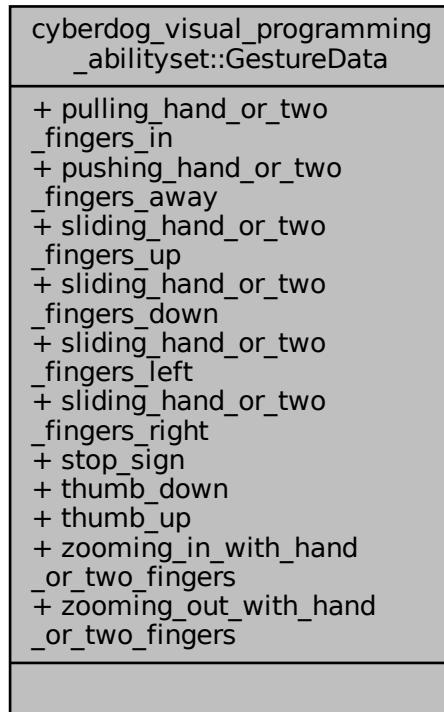
### 类成员

double	default.value	默认值
double	maximum.value	最大值
double	minimum.value	最小值
string	unit	单位

### 7.1.1.2 struct cyberdog\_visual\_programming\_abilityset::GestureData

### 手势数据

cyberdog\_visual\_programming\_abilityset::GestureData 的协作图:



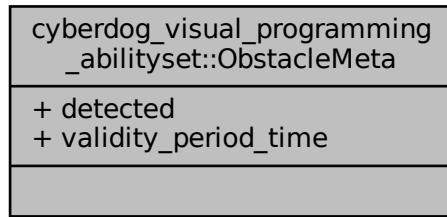
### 类成员

bool	pulling_hand_or_two_fingers_in	手掌拉近
bool	pushing_hand_or_two_fingers_away	手掌推开
bool	sliding_hand_or_two_fingers_down	手向下压
bool	sliding_hand_or_two_fingers_left	手向左推
bool	sliding_hand_or_two_fingers_right	手向右推
bool	sliding_hand_or_two_fingers_up	手向上抬
bool	stop_sign	停止手势
bool	thumb_down	大拇指朝下
bool	thumb_up	大拇指朝上
bool	zooming_in_with_hand_or_two_fingers	张开手掌或手指
bool	zooming_out_with_hand_or_two_fingers	闭合手掌或手指

### 7.1.1.3 struct cyberdog\_visual\_programming\_abilityset::ObstacleMeta

#### 障碍物

cyberdog\_visual\_programming\_abilityset::ObstacleMeta 的协作图:



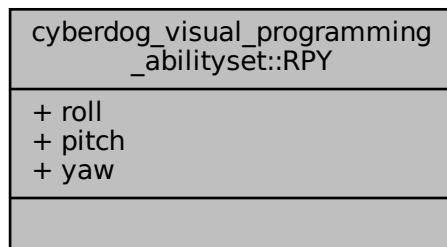
类成员

bool	detected	检测到
time_point< system_clock >	validity_period_time	有效期时间

#### 7.1.1.4 struct cyberdog\_visual\_programming\_abilityset::RPY

欧拉角约束调用

cyberdog\_visual\_programming\_abilityset::RPY 的协作图:



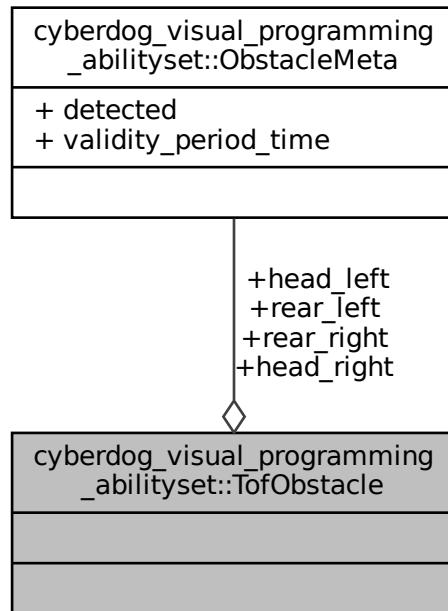
类成员

double	pitch	俯仰
double	roll	横滚
double	yaw	偏航

### 7.1.1.5 struct cyberdog\_visual\_programming\_abilityset::TofObstacle

障碍物

cyberdog\_visual\_programming\_abilityset::TofObstacle 的协作图:



类成员

<code>ObstacleMeta</code>	<code>head_left</code>	头部左侧
<code>ObstacleMeta</code>	<code>head_right</code>	头部右侧
<code>ObstacleMeta</code>	<code>rear_left</code>	尾部左侧
<code>ObstacleMeta</code>	<code>rear_right</code>	尾部右侧

## 7.1.2 类型定义说明

### 7.1.2.1 ActNavigation

```
using cyberdog_visual_programming_abilityset::ActNavigation = typedef protocol::action::Navigation
```

导航动作

### 7.1.2.2 CyberdogJson

```
using cyberdog_visual_programming_abilityset::CyberdogJson = typedef cyberdog::common::Cyberdog<→
Json
```

JSON 解析及构建模块类型

### 7.1.2.3 CyberdogToml

```
using cyberdog_visual_programming_abilityset::CyberdogToml = typedef cyberdog::common::Cyberdog<→
Toml
```

Toml 解析及构建模块类型

### 7.1.2.4 MsgAlgoStatus

```
using cyberdog_visual_programming_abilityset::MsgAlgoStatus = typedef protocol::msg::AlgoTask<→
Status
```

算法状态

### 7.1.2.5 MsgAudioPlay

```
using cyberdog_visual_programming_abilityset::MsgAudioPlay = typedef protocol::msg::AudioPlay
语音消息:离线
```

### 7.1.2.6 MsgAudioPlayExtend

```
using cyberdog_visual_programming_abilityset::MsgAudioPlayExtend = typedef protocol::msg::←
AudioPlayExtend
```

语音消息:在线

### 7.1.2.7 MsgBmsStatus

```
using cyberdog_visual_programming_abilityset::MsgBmsStatus = typedef protocol::msg::BmsStatus
```

电池系统

### 7.1.2.8 MsgBool

```
using cyberdog_visual_programming_abilityset::MsgBool = typedef std_msgs::msg::Bool
```

Voiceprint

### 7.1.2.9 MsgConnectorStatus

```
using cyberdog_visual_programming_abilityset::MsgConnectorStatus = typedef protocol::msg::→
ConnectorStatus
```

连接状态

### 7.1.2.10 MsgFaceRes

```
using cyberdog_visual_programming_abilityset::MsgFaceRes = typedef protocol::msg::FaceRecognition→
Result
```

人脸识别反馈

### 7.1.2.11 MsgGesture

```
using cyberdog_visual_programming_abilityset::MsgGesture = typedef protocol::msg::Gesture→
ActionResult
```

手势信息

### 7.1.2.12 MsgGpsPayload

```
using cyberdog_visual_programming_abilityset::MsgGpsPayload = typedef protocol::msg::GpsPayload
GPS数据
```

### 7.1.2.13 MsgHeader

```
using cyberdog_visual_programming_abilityset::MsgHeader = typedef std_msgs::msg::Header
消息头
```

### 7.1.2.14 MsgHeadTofPayload

```
using cyberdog_visual_programming_abilityset::MsgHeadTofPayload = typedef protocol::msg::Head→
TofPayload
```

头部tof数据

### 7.1.2.15 MsgImu

```
using cyberdog_visual_programming_abilityset::MsgImu = typedef sensor_msgs::msg::Imu
IMU
```

### 7.1.2.16 MsgLaserScan

```
using cyberdog_visual_programming_abilityset::MsgLaserScan = typedef sensor_msgs::msg::LaserScan  
激光数据
```

### 7.1.2.17 MsgMotionID

```
using cyberdog_visual_programming_abilityset::MsgMotionID = typedef protocol::msg::MotionID  
运动 id
```

### 7.1.2.18 MsgMotionSequenceGait

```
using cyberdog_visual_programming_abilityset::MsgMotionSequenceGait = typedef protocol::msg::MotionSequenceGait
```

运动序列步态

### 7.1.2.19 MsgMotionSequencePace

```
using cyberdog_visual_programming_abilityset::MsgMotionSequencePace = typedef protocol::msg::MotionSequencePace
```

运动序列步伐

### 7.1.2.20 MsgMotionServoCmd

```
using cyberdog_visual_programming_abilityset::MsgMotionServoCmd = typedef protocol::msg::MotionServoCmd
```

伺服指令

### 7.1.2.21 MsgMotionServoResponse

```
using cyberdog_visual_programming_abilityset::MsgMotionServoResponse = typedef protocol::msg::MotionServoResponse
```

伺服指令

### 7.1.2.22 MsgMotionStatus

```
using cyberdog_visual_programming_abilityset::MsgMotionStatus = typedef protocol::msg::MotionStatus
```

运动状态

### 7.1.2.23 MsgOdometry

```
using cyberdog_visual_programming_abilityset::MsgOdometry = typedef nav_msgs::msg::Odometry
```

里程计

### 7.1.2.24 MsgPersonnel

```
using cyberdog_visual_programming_abilityset::MsgPersonnel = typedef protocol::msg::UserInformation
```

人员信息

### 7.1.2.25 MsgPoint

```
using cyberdog_visual_programming_abilityset::MsgPoint = typedef geometry_msgs::msg::Point
```

位置

### 7.1.2.26 MsgPose

```
using cyberdog_visual_programming_abilityset::MsgPose = typedef geometry_msgs::msg::Pose
```

位姿

### 7.1.2.27 MsgPoseStamped

```
using cyberdog_visual_programming_abilityset::MsgPoseStamped = typedef geometry_msgs::msg::PoseStamped
```

位姿

### 7.1.2.28 MsgPreset

```
using cyberdog_visual_programming_abilityset::MsgPreset = typedef protocol::msg::Label
```

预置点 消息

### 7.1.2.29 MsgRange

```
using cyberdog_visual_programming_abilityset::MsgRange = typedef sensor_msgs::msg::Range
```

Range

### 7.1.2.30 MsgRearTofPayload

```
using cyberdog_visual_programming_abilityset::MsgRearTofPayload = typedef protocol::msg::Rear<→  
TofPayload
```

尾部tof数据

### 7.1.2.31 MsgSingleTofPayload

```
using cyberdog_visual_programming_abilityset::MsgSingleTofPayload = typedef protocol::msg::←  
SingleTofPayload
```

单个tof数据

### 7.1.2.32 MsgSport

```
using cyberdog_visual_programming_abilityset::MsgSport = typedef protocol::msg::SportCounts<→  
Result
```

骨骼点消息

### 7.1.2.33 MsgString

```
using cyberdog_visual_programming_abilityset::MsgString = typedef std_msgs::msg::String
```

GRPC Voiceprint

### 7.1.2.34 MsgTouchStatus

```
using cyberdog_visual_programming_abilityset::MsgTouchStatus = typedef protocol::msg::Touch<→  
Status
```

触摸板

### 7.1.2.35 MsgTrainingWords

```
using cyberdog_visual_programming_abilityset::MsgTrainingWords = typedef protocol::msg::Train<→  
Plan
```

训练词

### 7.1.2.36 MsgVisualProgrammingOperate

```
using cyberdog_visual_programming_abilityset::MsgVisualProgrammingOperate = typedef protocol←  
::msg::VisualProgrammingOperate
```

任务操作消息

### 7.1.2.37 SrvAudioGetVolume

```
using cyberdog_visual_programming_abilityset::SrvAudioGetVolume = typedef protocol::srv::Audio<→
VolumeGet
```

audio 音量获取

### 7.1.2.38 SrvAudioSetVolume

```
using cyberdog_visual_programming_abilityset::SrvAudioSetVolume = typedef protocol::srv::Audio<→
VolumeSet
```

audio 音量设置

### 7.1.2.39 SrvAudioTextPlay

```
using cyberdog_visual_programming_abilityset::SrvAudioTextPlay = typedef protocol::srv::Audio<→
TextPlay
```

audio 播放

### 7.1.2.40 SrvCancelNavigation

```
using cyberdog_visual_programming_abilityset::SrvCancelNavigation = typedef protocol::srv::←
StopAlgoTask
```

取消导航

### 7.1.2.41 SrvFaceRec

```
using cyberdog_visual_programming_abilityset::SrvFaceRec = typedef protocol::srv::FaceRec
```

人脸识别服务

### 7.1.2.42 SrvGesture

```
using cyberdog_visual_programming_abilityset::SrvGesture = typedef protocol::srv::Gesture<→
ActionControl
```

手势服务

### 7.1.2.43 SrvGetPreset

```
using cyberdog_visual_programming_abilityset::SrvGetPreset = typedef protocol::srv::GetMapLabel
```

audio 获取预置点

#### 7.1.2.44 SrvLedExecute

```
using cyberdog_visual_programming_abilityset::SrvLedExecute = typedef protocol::srv::LedExecute  
led
```

#### 7.1.2.45 SrvMotionResultCmd

```
using cyberdog_visual_programming_abilityset::SrvMotionResultCmd = typedef protocol::srv::←  
MotionResultCmd
```

结果指令

#### 7.1.2.46 SrvMotionSequenceShow

```
using cyberdog_visual_programming_abilityset::SrvMotionSequenceShow = typedef protocol::srv::←  
MotionSequenceShow
```

序列动作

#### 7.1.2.47 SrvPersonnel

```
using cyberdog_visual_programming_abilityset::SrvPersonnel = typedef protocol::srv::AllUser←  
Search
```

人员服务

#### 7.1.2.48 SrvSetBool

```
using cyberdog_visual_programming_abilityset::SrvSetBool = typedef std_srvs::srv::SetBool  
bool类型
```

#### 7.1.2.49 SrvSport

```
using cyberdog_visual_programming_abilityset::SrvSport = typedef protocol::srv::SportManager  
骨骼点服务
```

#### 7.1.2.50 SrvTrainingWords

```
using cyberdog_visual_programming_abilityset::SrvTrainingWords = typedef protocol::srv::Train←  
PlanAll
```

训练词服务

### 7.1.2.51 SrvVisualProgrammingOperate

```
using cyberdog_visual_programming_abilityset::SrvVisualProgrammingOperate = typedef protocol<~
::srv::VisualProgrammingOperate
```

任务操作服务

## 7.1.3 枚举类型说明

### 7.1.3.1 GestureType

```
enum cyberdog_visual_programming_abilityset::GestureType
```

手势种类

枚举值

no_gesture	无手势
pulling_hand_or_two_fingers_in	手掌拉近
pushing_hand_or_two_fingers_away	手掌推开
sliding_hand_or_two_fingers_up	手向上抬
sliding_hand_or_two_fingers_down	手向下压
sliding_hand_or_two_fingers_left	手向左推
sliding_hand_or_two_fingers_right	手向右推
stop_sign	停止手势
thumb_up	大拇指朝上
zooming_in_with_hand_or_two_fingers	张开手掌或手指
zooming_out_with_hand_or_two_fingers	闭合手掌或手指
thumb_down	大拇指朝下
id_upper_limit	手势id上限

### 7.1.3.2 LedConstraint

```
enum cyberdog_visual_programming_abilityset::LedConstraint
```

LED约束

枚举值

target_head	头灯
target_tail	尾灯

## 枚举值

target_mini	眼灯
effect_line_on	[灯带]常亮
effect_line_blink	[灯带]闪烁
effect_line_blink_fast	[灯带]快速闪烁
effect_line_breath	[灯带]呼吸
effect_line_breath_fast	[灯带]快速呼吸
effect_line_one_by_one	[灯带]逐个点亮
effect_line_one_by_one_fast	[灯带]快速逐个点亮
effect_line_back_and_forth	[灯带]往返逐个点亮
effect_line_trailing_race	[灯带]拖尾流跑马
system_effect_line_off	[灯带]常灭
system_effect_line_red_on	[灯带]红灯常亮
system_effect_line_red_blink	[灯带]红灯闪烁
system_effect_line_red_blink_fast	[灯带]红灯快速闪烁
system_effect_line_red_breath	[灯带]红灯呼吸
system_effect_line_red_breath_fast	[灯带]红灯快速呼吸
system_effect_line_red_one_by_one	[灯带]红灯逐个点亮
system_effect_line_red_one_by_one_fast	[灯带]红灯快速逐个点亮
system_effect_line_blue_on	[灯带]红灯常亮
system_effect_line_blue_blink	[灯带]红灯闪烁
system_effect_line_blue_blink_fast	[灯带]红灯快速闪烁
system_effect_line_blue_breath	[灯带]红灯呼吸
system_effect_line_blue_breath_fast	[灯带]红灯快速呼吸
system_effect_line_blue_one_by_one	[灯带]红灯逐个点亮
system_effect_line_blue_one_by_one_fast	[灯带]红灯快速逐个点亮
system_effect_line_yellow_on	[灯带]红灯常亮
system_effect_line_yellow_blink	[灯带]红灯闪烁
system_effect_line_yellow_blink_fast	[灯带]红灯快速闪烁
system_effect_line_yellow_breath	[灯带]红灯呼吸
system_effect_line_yellow_breath_fast	[灯带]红灯快速呼吸
system_effect_line_yellow_one_by_one	[灯带]红灯逐个点亮
system_effect_line_yellow_one_by_one_fast	[灯带]红灯快速逐个点亮
effect_mini_circular_breath	[眼灯]圆形缩放
effect_mini_circular_ring	[眼灯]画圆环
system_effect_mini_off	[眼灯]常灭
system_effect_mini_rectangle_color	[眼灯]方块变色
system_effect_mini_centre_color	[眼灯]中间彩带
system_effect_mini_three_circular	[眼灯]三圆呼吸
system_effect_mini_one_by_one	[眼灯]彩带逐个点亮

### 7.1.3.3 MotionId

```
enum cyberdog_visual_programming_abilityset::MotionId
```

运动ID 0.急停 1.趴下 2.位控姿态 3.力控姿态 4.运动中

枚举值

emergency_stop	急停
get_down	[1]趴下
resume_standing	[2]恢复站立
servo_standing	[3]伺服站立(伺服指令END帧后状态)
back_flip	[2]后空翻
front_flip	[2]前空翻
bow	[2]作揖
roll_left	[2]向左侧躺后恢复
walk_the_dog	[4]遛狗
jump_stair	[2]跳上台阶
right_somersault	[2]右侧空翻
left_somersault	[2]左侧空翻
run_and_jump_front_flip	[2]跑跳前空翻
jump3d_left90deg	[2]3D跳:左转90度
jump3d_right90deg	[2]3D跳:右转90度
jump3d_forward60cm	[2]3D跳:前跳60cm
jump3d_forward30cm	[2]3D跳:前跳30cm
jump3d_left20cm	[2]3D跳:左跳20cm
jump3d_right20cm	[2]3D跳:右跳20cm
jump3d_up30cm	[2]3D跳:向上30cm
jump3d_down_stair	[2]3D跳:跳下台阶
roll_right	[2]向右侧躺后恢复
dance_collection	[2]舞蹈集合
hold_left_hand	[2]握手
hold_right_hand	[2]握右手
sit_down	[2]坐下
butt_circle	[2]屁股画圆
head_circle	[2]头画圆
stretch_the_body	[2]伸展身体
shake_ass_left	[2]向左摇晃屁股
shake_ass_right	[2]向右摇晃屁股
shake_ass_from_side_to_side	[2]左右摇晃屁股
ballet	[3]芭蕾舞
space_walk	[3]太空步
front_leg_jumping	[3]前腿开合跳
hind_leg_jumping	[3]后腿开合跳
lift_the_left_leg_and_nod	[3]左腿抬起并点头

枚举值

lift_the_right_leg_and_nod	[3]右腿抬起并点头
left_front_right_back_legs_apart	[3]左前右后岔开腿
right_front_left_back_legs_apart	[3]右前左后岔开腿
walk_nodding	[3]走路点头
walking_with_divergence_and_adduction_alternately	[3]岔开内收交替走路
nodding_in_place	[3]原地踏步点头
front_legs_jump_back_and_forth	[3]前腿前后跳
hind_legs_jump_back_and_forth	[3]后腿前后跳
alternately_front_leg_lift	[3]前腿交替抬起
alternately_hind_leg_lift	[3]后腿交替抬起
jump_collection	[3]跳跃合集
stretching_left_and_right	[3]左右伸腿踏步
jump_forward_and_backward	[3]前后摆腿跳跃
step_left_and_right	[3]左右摆腿踏步
right_leg_back_and_forth_stepping	[3]右腿前后踏步
left_leg_back_and_forth_stepping	[3]左腿前后踏步
squat_down_on_all_fours	[2]四足蹲起
push_ups	[2]俯卧撑
bow_to_each_other	[2]作揖比心
absolute_force_control_attitude	[3]绝对力控姿态 (绝对姿态)
relatively_force_control_attitude	[3]相对力控姿态
absolute_position_control_attitude	[2]绝对位控姿态 (过渡站立)
relatively_position_control_attitude	[2]相对位控姿态
relatively_position_control_attitude_insert_frame_1	[2]相对位姿插帧1(握手)
relatively_position_control_attitude_insert_frame_2	[2]相对位姿插帧2(任务空时)
jump_back_and_forth	[3]前后跳
small_jump_walking	[3]小跳行走
trot_walking	[3]慢速 (小跑) 行走
automatic_frequency_conversion_walking	[3]自动变频行走
run_fast_walking	[3]快跑行走
sequence_custom	[3]序列动作
illegal_motion_id	非法运动ID

#### 7.1.3.4 Processor

```
enum cyberdog_visual_programming_abilityset::Processor
```

处理器类型

枚举值

task	任务
------	----

### 7.1.3.5 RPYType

enum `cyberdog_visual_programming_abilityset::RPYType`

通用状态描述

欧拉角约束调用 角度类型 合法值

枚举值

ROLL	横滚
PITCH	俯仰
YAW	偏航

### 7.1.3.6 SkeletonType

enum `cyberdog_visual_programming_abilityset::SkeletonType`

骨骼点识别类型 合法值

枚举值

SPORT_SQUAT	深蹲
SPORT_HIGHKNEES	高抬腿
SPORT_SITUP	仰卧起坐
SPORT_PRESSUP	俯卧撑
SPORT_PLANK	平板支撑
SPORT_JUMPJACK	开合跳

### 7.1.3.7 StateCode

enum `cyberdog_visual_programming_abilityset::StateCode`

通用状态约束:{内部错误码 = 错误码 - 基础码} 错误基础码: 5800 内部错误码: 01~20:全局; 21~30:模块;  
31~40:ROS; 41~50:Topic; 51~60:Service; 61~70:Action;

枚举值

invalid	无效
success	成功
error_base	错误基础码
fail	[ 全局码 ]失败
uninitialized	[ 全局码 ]未初始化
fsm_does_not_allow	[ 全局码 ]状态机不允许
module_status_error	[ 全局码 ]模块状态错误
network_error	[ 全局码 ]网络错误
no_operation_authority	[ 全局码 ]无操作权限
timeout	[ 全局码 ]超时
command_does_not_support	[ 全局码 ]指令不支持
self_test_failed	[ 全局码 ]自检失败
parameter_is_invalid	[ 全局码 ]参数不合法
status_is_busy	[ 全局码 ]状态忙碌
hardware_error	[ 全局码 ]硬件错误
command_waiting_execute	[ 模块码 ]命令等待执行
spin_future_interrupted	[ ROS ]请求服务中断
spin_future_timeout	[ ROS ]请求服务超时/延迟
no_data_update	[ Topic ]无数据更新
service_client_interrupted	[Service]客户端在请求服务出现时被打断
service_appear_timeout	[Service]等待服务出现（启动）超时
service_request_interrupted	[Service]请求服务中断
service_request_rejected	[Service]请求服务被拒绝
service_request_timeout	[Service]请求服务超时/延迟
action_request_timeout	[Action]请求动作超时/延迟
action_request_rejected	[Action]请求动作被拒绝
action_result_timeout	[Action]等待动作结果超时/延迟

### 7.1.3.8 TimeMode

```
enum cyberdog_visual_programming_abilityset::TimeMode
```

约束获取时间接口入参的合法值

枚举值

Ms1970	1970年1月1日到现在的时间(毫秒)
_Y_M_D_H_M↔_S	"*Y*M*D*H*M*S"
STANDARD	"*Y.*M.*D-*H:*M:*S"
DETAILED	"*Y.*M.*D-*H:*M:*S-US"

## 7.1.4 函数说明

### 7.1.4.1 Angle2Radian()

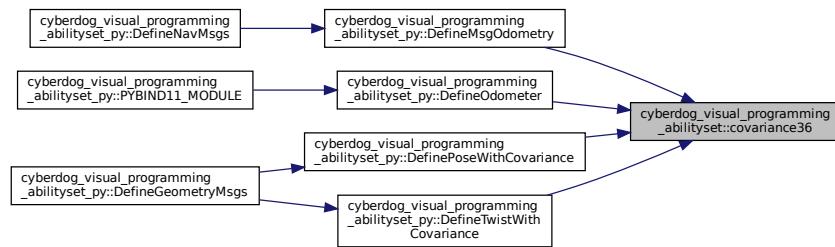
```
double cyberdog_visual_programming_abilityset::Angle2Radian (
    const double _degree )
```

角度转换为弧度

### 7.1.4.2 covariance36()

```
std::string cyberdog_visual_programming_abilityset::covariance36 (
    const std::array< double, 36 > & _ary,
    const std::string _head,
    const int line_size = 6 )
```

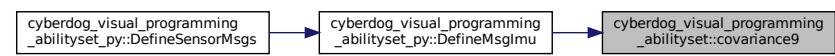
数组转字符串 这是这个函数的调用关系图:



### 7.1.4.3 covariance9()

```
std::string cyberdog_visual_programming_abilityset::covariance9 (
    const std::array< double, 9 > & _ary,
    const std::string _head,
    const int line_size = 3 )
```

数组转字符串 这是这个函数的调用关系图:



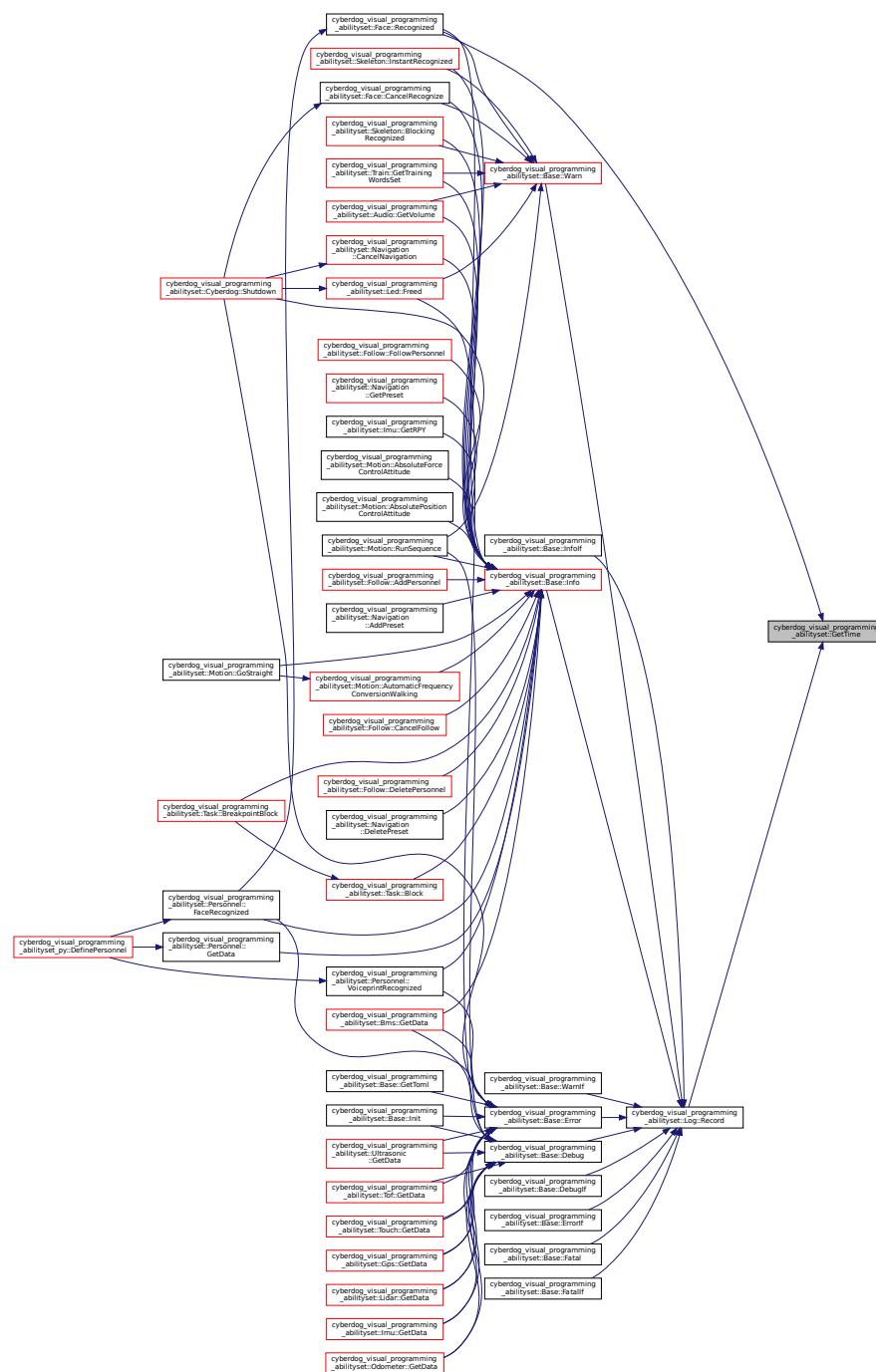
#### 7.1.4.4 Format()

```
template<typename ... Args>
std::string cyberdog_visual_programming_abilityset::Format (
    const char * format,
    Args... args )
```

#### 7.1.4.5 GetTime()

```
std::string cyberdog_visual_programming_abilityset::GetTime (
    int nowModo = 0 )
```

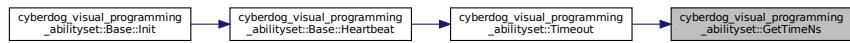
获取时间戳 这是这个函数的调用关系图:



#### 7.1.4.6 GetTimeNs()

```
uint64_t cyberdog_visual_programming_abilityset::GetTimeNs ( )
```

获取时间 这是这个函数的调用关系图:



#### 7.1.4.7 GetWorkspace()

```
bool cyberdog_visual_programming_abilityset::GetWorkspace (
    std::string & _workspace )
```

获取工作空间 函数调用图:



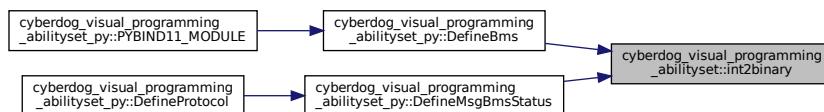
这是这个函数的调用关系图:



#### 7.1.4.8 int2binary()

```
std::string cyberdog_visual_programming_abilityset::int2binary (
    const int _int )
```

int 转 2进制 这是这个函数的调用关系图:



#### 7.1.4.9 intVectorToString()

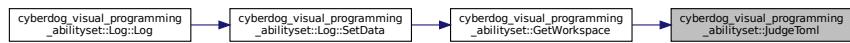
```
std::string cyberdog_visual_programming_abilityset::intVectorToString (
    const std::vector< int > & _vct )
```

数组 转 字符串

#### 7.1.4.10 JudgeToml()

```
bool cyberdog_visual_programming_abilityset::JudgeToml (
    const std::string & _toml_file )
```

判断 toml 这是这个函数的调用关系图:



#### 7.1.4.11 msgFaceResMap()

```
std::string cyberdog_visual_programming_abilityset::msgFaceResMap (
    const std::map< std::string, MsgFaceRes > & _map,
    const std::string _spacing )
```

数组 转 字符串

#### 7.1.4.12 msgFaceResVector()

```
std::string cyberdog_visual_programming_abilityset::msgFaceResVector (
    const std::vector< MsgFaceRes > & _vct,
    const std::string _spacing )
```

数组 转 字符串

#### 7.1.4.13 msgPersonnelVector()

```
std::string cyberdog_visual_programming_abilityset::msgPersonnelVector (
    const std::vector< MsgPersonnel > & _vct,
    const std::string _spacing )
```

数组 转 字符串

#### 7.1.4.14 msgPresetMap()

```
std::string cyberdog_visual_programming_abilityset::msgPresetMap (
    const std::map< std::string, MsgPreset > & _map,
    const std::string _spacing )
```

数组转字符串

#### 7.1.4.15 msgPresetVector()

```
std::string cyberdog_visual_programming_abilityset::msgPresetVector (
    const std::vector< MsgPreset > & _vct,
    const std::string _spacing )
```

数组转字符串

#### 7.1.4.16 msgTrainingWordsMap()

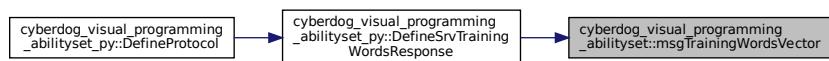
```
std::string cyberdog_visual_programming_abilityset::msgTrainingWordsMap (
    const std::map< std::string, MsgTrainingWords > & _map,
    const std::string _spacing )
```

数组转字符串

#### 7.1.4.17 msgTrainingWordsVector()

```
std::string cyberdog_visual_programming_abilityset::msgTrainingWordsVector (
    const std::vector< MsgTrainingWords > & _vct,
    const std::string _spacing )
```

数组转字符串 这是这个函数的调用关系图:



#### 7.1.4.18 Qrientation2RPY()

```
geometry_msgs::msg::Vector3 cyberdog_visual_programming_abilityset::Qrientation2RPY (
    const double _x,
    const double _y,
    const double _z,
    const double _w )
```

四元数转欧拉角

### 7.1.4.19 Radian2Angle()

```
double cyberdog_visual_programming_abilityset::Radian2Angle (
    const double _rad )
```

弧度转换为角度

### 7.1.4.20 RPY2Qrientation()

```
geometry_msgs::msg::Quaternion cyberdog_visual_programming_abilityset::RPY2Qrientation (
    const double _r,
    const double _p,
    const double _y )
```

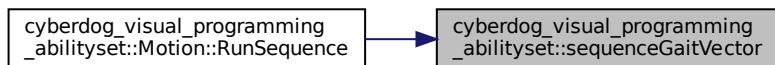
欧拉角转四元数 这是这个函数的调用关系图:



### 7.1.4.21 sequenceGaitVector()

```
std::string cyberdog_visual_programming_abilityset::sequenceGaitVector (
    const std::vector< MsgMotionSequenceGait > & _vct,
    const std::string _spacing )
```

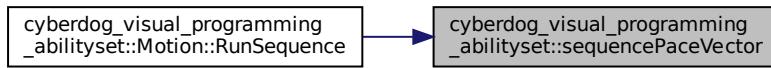
数组 转 字符串 这是这个函数的调用关系图:



### 7.1.4.22 sequencePaceVector()

```
std::string cyberdog_visual_programming_abilityset::sequencePaceVector (
    const std::vector< MsgMotionSequencePace > & _vct,
    const std::string _spacing )
```

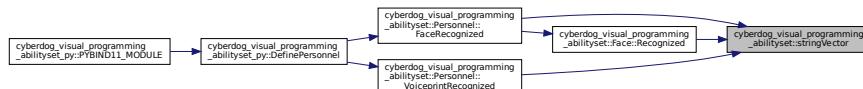
数组转字符串 这是这个函数的调用关系图:



### 7.1.4.23 stringVector()

```
std::string cyberdog_visual_programming_abilityset::stringVector (
    const std::vector< std::string > & _vct )
```

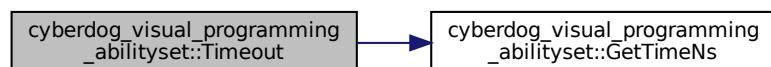
数组转字符串 这是这个函数的调用关系图:



### 7.1.4.24 Timeout()

```
bool cyberdog_visual_programming_abilityset::Timeout (
    const uint64_t & _old_ns,
    uint64_t _timeout_ms = 3000 )
```

判断超时 函数调用图:



这是这个函数的调用关系图:



## 7.1.5 变量说明

### 7.1.5.1 bms\_data\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::bms_data_cv_
```

### 7.1.5.2 bms\_data\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::bms_data_cvm_
```

### 7.1.5.3 compensation\_frame\_mtx\_

```
std::mutex cyberdog_visual_programming_abilityset::compensation_frame_mtx_
```

补偿帧互斥量

### 7.1.5.4 gesture\_recognition\_state\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::gesture_recognition_state_cv_
```

### 7.1.5.5 gesture\_recognition\_state\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::gesture_recognition_state_cvm_
```

### 7.1.5.6 gps\_data\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::gps_data_cv_
```

### 7.1.5.7 **gps\_data\_cvm\_**

```
std::mutex cyberdog_visual_programming_abilityset::gps_data_cvm_
```

### 7.1.5.8 **imu\_data\_cv\_**

```
std::condition_variable cyberdog_visual_programming_abilityset::imu_data_cv_
```

### 7.1.5.9 **imu\_data\_cvm\_**

```
std::mutex cyberdog_visual_programming_abilityset::imu_data_cvm_
```

### 7.1.5.10 **lidar\_data\_cv\_**

```
std::condition_variable cyberdog_visual_programming_abilityset::lidar_data_cv_
```

### 7.1.5.11 **lidar\_data\_cvm\_**

```
std::mutex cyberdog_visual_programming_abilityset::lidar_data_cvm_
```

### 7.1.5.12 **odometer\_data\_cv\_**

```
std::condition_variable cyberdog_visual_programming_abilityset::odometer_data_cv_
```

### 7.1.5.13 **odometer\_data\_cvm\_**

```
std::mutex cyberdog_visual_programming_abilityset::odometer_data_cvm_
```

### 7.1.5.14 **skeleton\_recognition\_state\_cv\_**

```
std::condition_variable cyberdog_visual_programming_abilityset::skeleton_recognition_state_cv_
```

### 7.1.5.15 skeleton\_recognition\_state\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::skeleton_recognition_state_cvm_
```

### 7.1.5.16 task\_state\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::task_state_cv_
```

### 7.1.5.17 task\_state\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::task_state_cvm_
```

### 7.1.5.18 tof\_data\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::tوف_data_cv_
```

### 7.1.5.19 tof\_data\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::tوف_data_cvm_
```

### 7.1.5.20 touch\_data\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::touch_data_cv_
```

### 7.1.5.21 touch\_data\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::touch_data_cvm_
```

### 7.1.5.22 training\_words\_state\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::training_words_state_cv_
```

### 7.1.5.23 training\_words\_state\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::training_words_state_cvm_
```

### 7.1.5.24 ultrasonic\_data\_cv\_

```
std::condition_variable cyberdog_visual_programming_abilityset::ultrasonic_data_cv_
```

### 7.1.5.25 ultrasonic\_data\_cvm\_

```
std::mutex cyberdog_visual_programming_abilityset::ultrasonic_data_cvm_
```

## 7.2 cyberdog\_visual\_programming\_abilityset\_py 命名空间参考

### 函数

- void `DefineAudio` (py::object m)
- void `DefineBase` (py::object m)
- void `DefineBms` (py::object m)
- void `DefineCyberdog` (py::object m)
- void `DefineFollow` (py::object m)
- void `DefineGesture` (py::object m)
- void `DefineGps` (py::object m)
- void `Definelmu` (py::object m)
- void `DefineLed` (py::object m)
- void `DefineLidar` (py::object m)
- `PYBIND11_MODULE` (abilityset, m)
- void `DefineMotion` (py::object m)
- void `DefineNavigation` (py::object m)
- void `DefineNetwork` (py::object m)
- void `DefineOdometer` (py::object m)
- void `DefinePersonnel` (py::object m)
- void `DefineBuiltInInterfaces` (py::object m)
- void `DefineTimer` (py::object m)
- void `DefineGeometryMsgs` (py::object m)
- void `DefinePoint` (py::object m)
- void `DefineQuaternion` (py::object m)
- void `DefinePose` (py::object m)
- void `DefineVector3` (py::object m)
- void `DefineTwist` (py::object m)
- void `DefinePoseWithCovariance` (py::object m)
- void `DefineTwistWithCovariance` (py::object m)
- void `DefineNavMsgs` (py::object m)
- void `DefineMsgOdometry` (py::object m)
- void `DefineProtocol` (py::object m)

- void `DefineMsgBmsStatus` (py::object m)
- void `DefineMsgTouchStatus` (py::object m)
- void `DefineMsgConnectorStatus` (py::object m)
- void `DefineMsgMotionSequenceGait` (py::object m)
- void `DefineMsgMotionSequencePace` (py::object m)
- void `DefineMsgGpsPayload` (py::object m)
- void `DefineMsgSingleTofPayload` (py::object m)
- void `DefineMsgHeadTofPayload` (py::object m)
- void `DefineMsgRearTofPayload` (py::object m)
- void `DefineMsgFaceRes` (py::object m)
- void `DefineMsgPreset` (py::object m)
- void `DefineMsgSport` (py::object m)
- void `DefineMsgTrainingWords` (py::object m)
- void `DefineSrvFaceRecResponse` (py::object m)
- void `DefineSrvSportResponse` (py::object m)
- void `DefineSrvTrainingWordsResponse` (py::object m)
- void `DefineSrvAudioTextPlayResponse` (py::object m)
- void `DefineSrvLedExecuteResponse` (py::object m)
- void `DefineSrvMotionResultCmdResponse` (py::object m)
- void `DefineSrvMotionSequenceShowResponse` (py::object m)
- void `DefineActNavigationResult` (py::object m)
- void `DefineSensorMsgs` (py::object m)
- void `DefineMsgRange` (py::object m)
- void `DefineMsgLaserScan` (py::object m)
- void `DefineMsgImu` (py::object m)
- void `DefineStdMsgs` (py::object m)
- void `DefineHeader` (py::object m)
- void `DefineSkeleton` (py::object m)
- void `DefineTask` (py::object m)
- void `DefineTof` (py::object m)
- void `DefineTouch` (py::object m)
- void `DefineTrain` (py::object m)
- void `DefineUltrasonic` (py::object m)
- void `DefineCommonType` (py::object)
- void `DefineState` (py::object)
- void `DefineAudioPlaySeviceResponse` (py::object)
- void `DefineAudioGetVolumeSeviceResponse` (py::object)
- void `DefineAudioSetVolumeSeviceResponse` (py::object)
- void `DefineLedConstraint` (py::object)
- void `DefineLedSeviceResponse` (py::object)
- void `DefineMotionResultServiceResponse` (py::object)
- void `DefineMotionSequenceServiceResponse` (py::object)
- void `DefineMotionServoCmdResponse` (py::object)
- void `DefineMotionParams` (py::object)
- void `DefineMotionId` (py::object)
- void `DefineMsgMotionSequenceGaitList` (py::object)
- void `DefineMsgMotionSequencePaceList` (py::object)
- void `DefineMotionSequence` (py::object)
- void `DefineObstacleMeta` (py::object)
- void `DefineTofObstacle` (py::object)
- void `DefineTofPayload` (py::object)
- void `DefineMsgPersonnel` (py::object)
- void `DefineMsgPersonnelList` (py::object)
- void `DefineSrvPersonnelResponse` (py::object)
- void `DefineFaceSeviceResponse` (py::object)

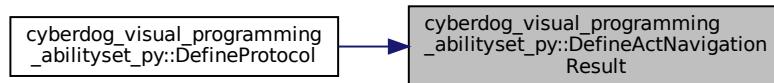
- void `DefineFaceRecognizedSeviceResponse` (py::object)
- void `DefineVoiceprintRecognizedResponse` (py::object)
- void `DefineGestureData` (py::object)
- void `DefineGestureType` (py::object)
- void `DefineGestureRecognizedSeviceResponse` (py::object)
- void `DefineGestureRecognizedMessageResponse` (py::object)
- void `DefineSkeletonType` (py::object)
- void `DefineSkeletonRecognizedSeviceResponse` (py::object)
- void `DefineSkeletonRecognizedMessageResponse` (py::object)
- void `DefineTrainingWordsRecognizedSeviceResponse` (py::object)
- void `DefineTrainingWordsRecognizedMessageResponse` (py::object)
- void `DefineMapPresetSeviceResponse` (py::object)
- void `DefineNavigationActionResponse` (py::object)

## 7.2.1 函数说明

### 7.2.1.1 DefineActNavigationResult()

```
void cyberdog_visual_programming_abilityset_py::DefineActNavigationResult (
    py::object    )
```

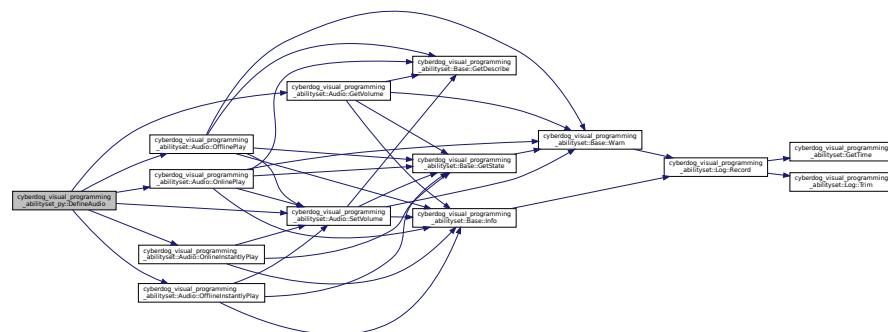
定义 `ActNavigation::Result` 这是这个函数的调用关系图:



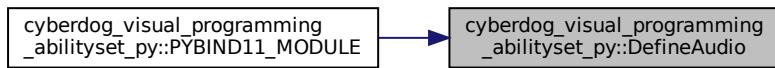
### 7.2.1.2 DefineAudio()

```
void cyberdog_visual_programming_abilityset_py::DefineAudio (
    py::object    )
```

定义 `Audio` 函数调用图:



这是这个函数的调用关系图:



#### 7.2.1.3 DefineAudioGetVolumeSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineAudioGetVolumeSeviceResponse (py::object )
```

定义 AudioGetVolumeSeviceResponse

#### 7.2.1.4 DefineAudioPlaySeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineAudioPlaySeviceResponse (py::object )
```

定义 AudioPlaySeviceResponse

#### 7.2.1.5 DefineAudioSetVolumeSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineAudioSetVolumeSeviceResponse (py::object )
```

定义 AudioSetVolumeSeviceResponse

#### 7.2.1.6 DefineBase()

```
void cyberdog_visual_programming_abilityset_py::DefineBase (py::object )
```

定义 Base 函数调用图:



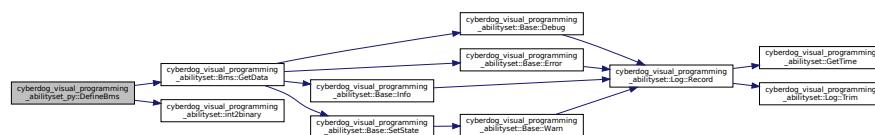
这是这个函数的调用关系图:



### 7.2.1.7 DefineBms()

```
void cyberdog_visual_programming_abilityset_py::DefineBms (
    py::object
)
```

定义 Bms 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.8 DefineBuiltInInterfaces()

```
void cyberdog_visual_programming_abilityset_py::DefineBuiltInInterfaces (
    py::object
)
```

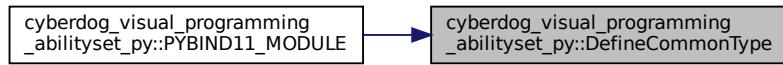
定义 builtin\_interfaces 函数调用图:



### 7.2.1.9 DefineCommonType()

```
void cyberdog_visual_programming_abilityset_py::DefineCommonType (
    py::object )
```

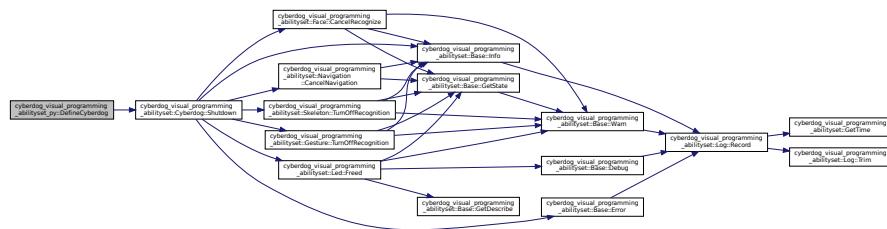
定义 通用类型 这是这个函数的调用关系图:



### 7.2.1.10 DefineCyberdog()

```
void cyberdog_visual_programming_abilityset_py::DefineCyberdog (
    py::object )
```

定义 Cyberdog 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.11 DefineFaceRecognizedSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineFaceRecognizedSeviceResponse (
    py::object )
```

定义 FaceRecognizedSeviceResponse

### 7.2.1.12 DefineFaceSeviceResponse()

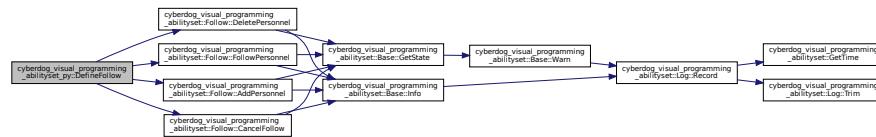
```
void cyberdog_visual_programming_abilityset_py::DefineFaceSeviceResponse (
    py::object    )
```

定义 FaceSeviceResponse

### 7.2.1.13 DefineFollow()

```
void cyberdog_visual_programming_abilityset_py::DefineFollow (
    py::object    )
```

定义 Follow 函数调用图:



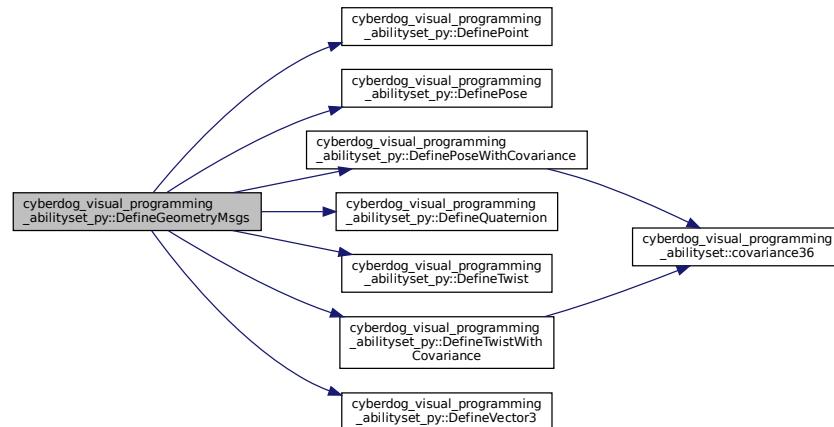
这是这个函数的调用关系图:



### 7.2.1.14 DefineGeometryMsgs()

```
void cyberdog_visual_programming_abilityset_py::DefineGeometryMsgs (
    py::object    )
```

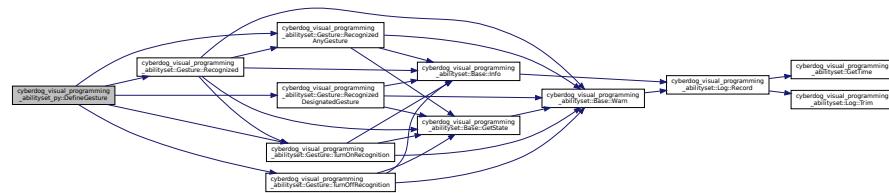
定义 geometry\_msgs 函数调用图:



### 7.2.1.15 DefineGesture()

```
void cyberdog_visual_programming_abilityset_py::DefineGesture (
    py::object )
```

定义 Gesture 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.16 DefineGestureData()

```
void cyberdog_visual_programming_abilityset_py::DefineGestureData (
    py::object )
```

定义 GestureData

### 7.2.1.17 DefineGestureRecognizedMessageResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineGestureRecognizedMessageResponse (
    py::object )
```

定义 GestureRecognizedMessageResponse

### 7.2.1.18 DefineGestureRecognizedSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineGestureRecognizedSeviceResponse (
    py::object )
```

定义 GestureRecognizedSeviceResponse

### 7.2.1.19 DefineGestureType()

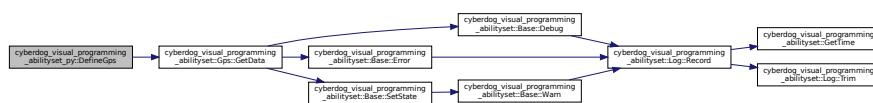
```
void cyberdog_visual_programming_abilityset_py::DefineGestureType (
    py::object )
```

定义 GestureType

### 7.2.1.20 DefineGps()

```
void cyberdog_visual_programming_abilityset_py::DefineGps (
    py::object )
```

定义 Gps 函数调用图:



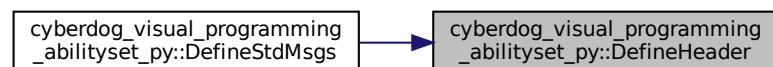
这是这个函数的调用关系图:



### 7.2.1.21 DefineHeader()

```
void cyberdog_visual_programming_abilityset_py::DefineHeader (
    py::object )
```

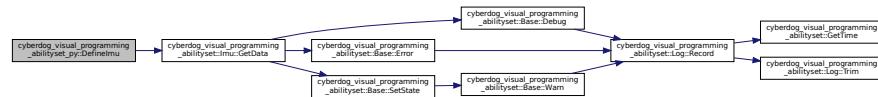
定义 Header 这是这个函数的调用关系图:



### 7.2.1.22 DefineImu()

```
void cyberdog_visual_programming_abilityset_py::DefineImu (
    py::object )
```

定义 Imu 函数调用图:



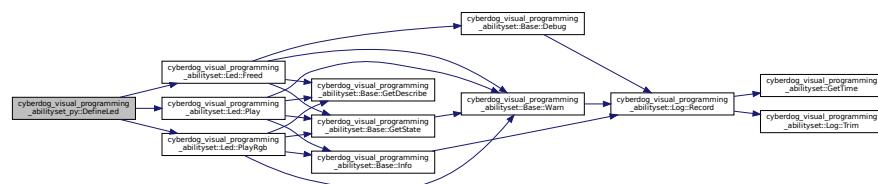
这是这个函数的调用关系图:



### 7.2.1.23 DefineLed()

```
void cyberdog_visual_programming_abilityset_py::DefineLed (
    py::object )
```

定义 Led 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.24 DefineLedConstraint()

```
void cyberdog_visual_programming_abilityset_py::DefineLedConstraint (
    py::object    )
```

定义 LedConstraint

### 7.2.1.25 DefineLedSeviceResponse()

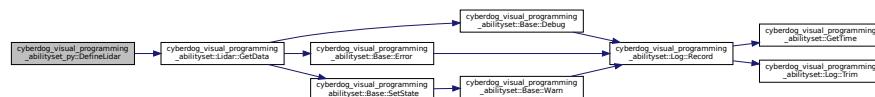
```
void cyberdog_visual_programming_abilityset_py::DefineLedSeviceResponse (
    py::object    )
```

定义 LedSeviceResponse

### 7.2.1.26 DefineLidar()

```
void cyberdog_visual_programming_abilityset_py::DefineLidar (
    py::object    )
```

定义 Lidar 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.27 DefineMapPresetSeviceResponse()

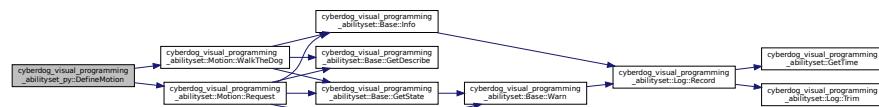
```
void cyberdog_visual_programming_abilityset_py::DefineMapPresetSeviceResponse (
    py::object    )
```

定义 MapPresetSeviceResponse

### 7.2.1.28 DefineMotion()

```
void cyberdog_visual_programming_abilityset_py::DefineMotion (
    py::object    )
```

定义 Motion 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.29 DefineMotionId()

```
void cyberdog_visual_programming_abilityset_py::DefineMotionId (
    py::object    )
```

定义 MotionId

### 7.2.1.30 DefineMotionParams()

```
void cyberdog_visual_programming_abilityset_py::DefineMotionParams (
    py::object    )
```

定义 MotionParams

### 7.2.1.31 DefineMotionResultServiceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineMotionResultServiceResponse (
    py::object    )
```

定义 MotionResultServiceResponse

### 7.2.1.32 DefineMotionSequence()

```
void cyberdog_visual_programming_abilityset_py::DefineMotionSequence (
    py::object    )
```

定义 ObstacleMeta

### 7.2.1.33 DefineMotionSequenceServiceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineMotionSequenceServiceResponse (
    py::object    )
```

定义 MotionSequenceServiceResponse

### 7.2.1.34 DefineMotionServoCmdResponse()

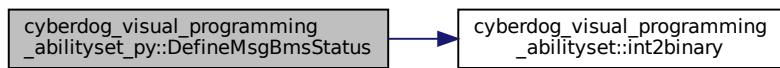
```
void cyberdog_visual_programming_abilityset_py::DefineMotionServoCmdResponse (
    py::object    )
```

定义 MotionServoCmdResponse

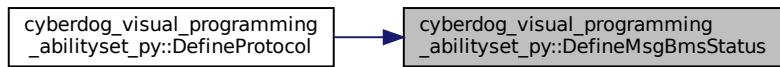
### 7.2.1.35 DefineMsgBmsStatus()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgBmsStatus (
    py::object    )
```

定义 MsgBmsStatus 函数调用图:



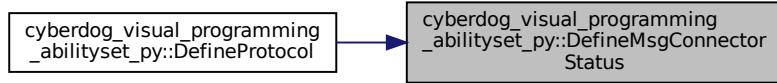
这是这个函数的调用关系图:



### 7.2.1.36 DefineMsgConnectorStatus()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgConnectorStatus (py::object )
```

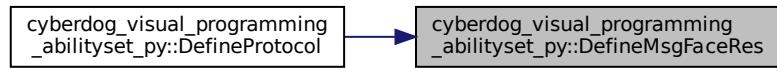
定义 MsgConnectorStatus 这是这个函数的调用关系图:



### 7.2.1.37 DefineMsgFaceRes()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgFaceRes (py::object )
```

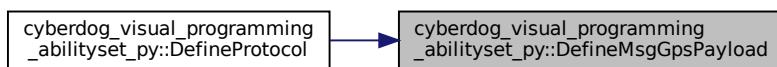
定义 MsgFaceRes 这是这个函数的调用关系图:



### 7.2.1.38 DefineMsgGpsPayload()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgGpsPayload (py::object )
```

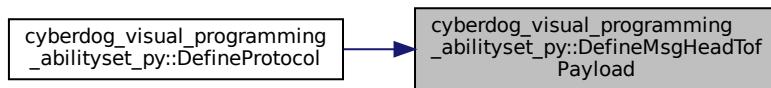
定义 GpsPayload 这是这个函数的调用关系图:



### 7.2.1.39 DefineMsgHeadToPayload()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgHeadToPayload (py::object )
```

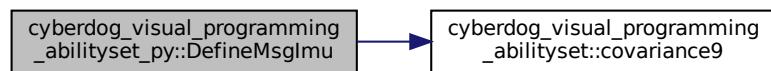
定义 HeadToPayload 这是这个函数的调用关系图:



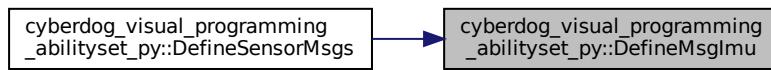
### 7.2.1.40 DefineMsgImu()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgImu (py::object )
```

定义 Imu 函数调用图:



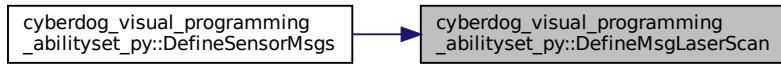
这是这个函数的调用关系图:



### 7.2.1.41 DefineMsgLaserScan()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgLaserScan (
    py::object    )
```

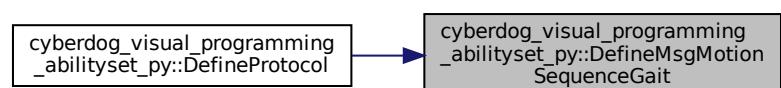
定义 LaserScan 这是这个函数的调用关系图:



### 7.2.1.42 DefineMsgMotionSequenceGait()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgMotionSequenceGait (
    py::object    )
```

定义 MotionSequencePace 这是这个函数的调用关系图:



### 7.2.1.43 DefineMsgMotionSequenceGaitList()

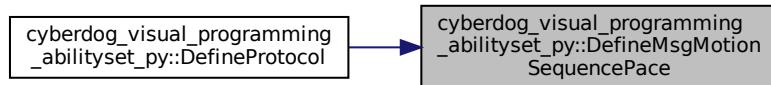
```
void cyberdog_visual_programming_abilityset_py::DefineMsgMotionSequenceGaitList (
    py::object    )
```

定义 MsgMotionSequenceGaitList

#### 7.2.1.44 DefineMsgMotionSequencePace()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgMotionSequencePace (py::object )
```

定义 MotionSequencePace 这是这个函数的调用关系图:



#### 7.2.1.45 DefineMsgMotionSequencePaceList()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgMotionSequencePaceList (py::object )
```

定义 MsgMotionSequencePaceList

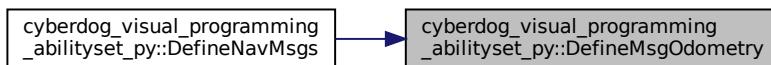
#### 7.2.1.46 DefineMsgOdometry()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgOdometry (py::object )
```

定义 MsgOdometry 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.47 DefineMsgPersonnel()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgPersonnel (
    py::object    )
```

定义 MsgPersonnel

### 7.2.1.48 DefineMsgPersonnelList()

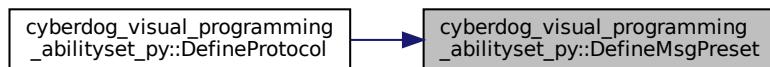
```
void cyberdog_visual_programming_abilityset_py::DefineMsgPersonnelList (
    py::object    )
```

定义 MsgPersonnelList

### 7.2.1.49 DefineMsgPreset()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgPreset (
    py::object    )
```

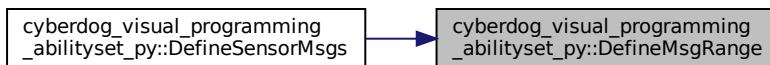
定义 MsgPreset 这是这个函数的调用关系图:



### 7.2.1.50 DefineMsgRange()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgRange (
    py::object    )
```

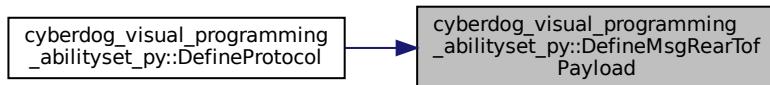
定义 Range 这是这个函数的调用关系图:



### 7.2.1.51 DefineMsgRearTofPayload()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgRearTofPayload (py::object )
```

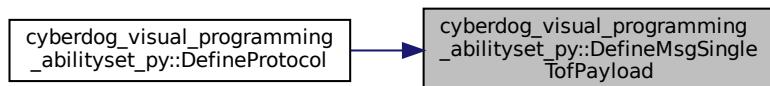
定义 RearTofPayload 这是这个函数的调用关系图:



### 7.2.1.52 DefineMsgSingleTofPayload()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgSingleTofPayload (py::object )
```

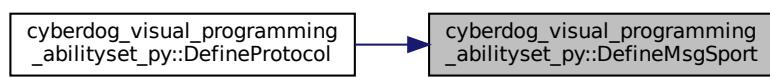
定义 SingleTofPayload 这是这个函数的调用关系图:



### 7.2.1.53 DefineMsgSport()

```
void cyberdog_visual_programming_abilityset_py::DefineMsgSport (py::object )
```

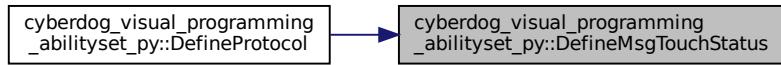
定义 MsgSport 这是这个函数的调用关系图:



#### **7.2.1.54 DefineMsgTouchStatus()**

```
void cyberdog_visual_programming_abilityset_py::DefineMsgTouchStatus (
```

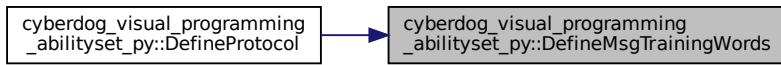
定义 `MsgTouchStatus` 这是这个函数的调用关系图：



#### **7.2.1.55 DefineMsgTrainingWords()**

```
void cyberdog_visual_programming_abilityset.py::DefineMsgTrainingWords (
```

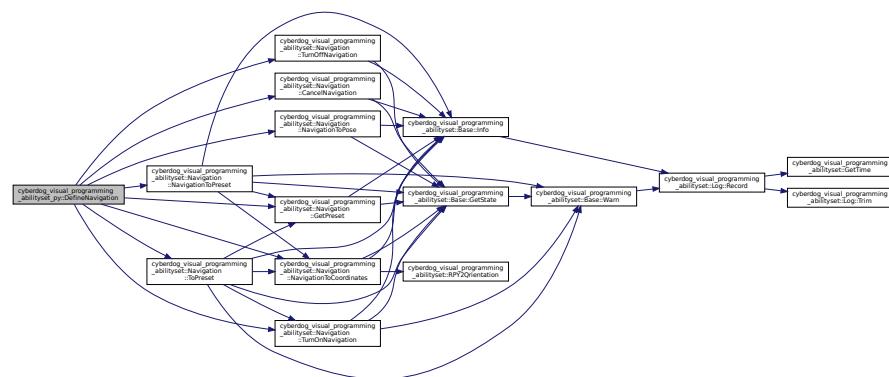
定义 `MsgTrainingWords` 这是这个函数的调用关系图:



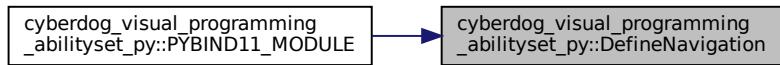
### 7.2.1.56 DefineNavigation()

```
void cyberdog_visual_programming_abilityset_py::DefineNavigation (
```

定义 Navigation 函数调用图:



这是这个函数的调用关系图:



#### 7.2.1.57 DefineNavigationActionResponse()

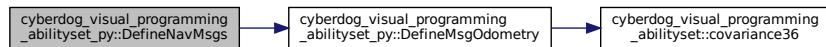
```
void cyberdog_visual_programming_abilityset_py::DefineNavigationActionResponse (py::object )
```

定义 NavigationActionResponse

#### 7.2.1.58 DefineNavMsgs()

```
void cyberdog_visual_programming_abilityset_py::DefineNavMsgs (py::object )
```

定义 nav\_msgs 函数调用图:



#### 7.2.1.59 DefineNetwork()

```
void cyberdog_visual_programming_abilityset_py::DefineNetwork (py::object )
```

定义 Network 这是这个函数的调用关系图:



### 7.2.1.60 DefineObstacleMeta()

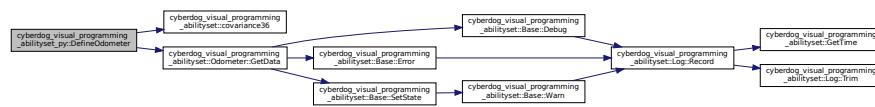
```
void cyberdog_visual_programming_abilityset_py::DefineObstacleMeta (
    py::object )
```

定义 TofObstacle

### 7.2.1.61 DefineOdometer()

```
void cyberdog_visual_programming_abilityset_py::DefineOdometer (
    py::object )
```

定义 Odometer 函数调用图:



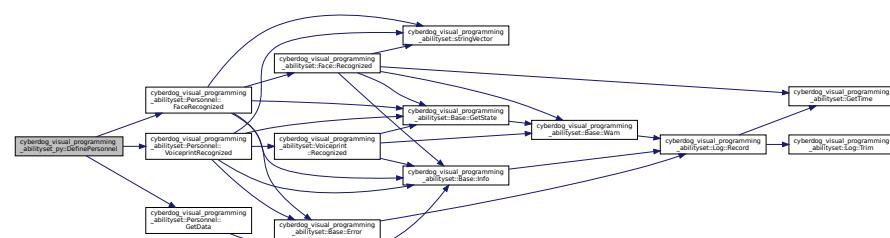
这是这个函数的调用关系图:



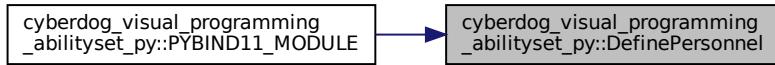
### 7.2.1.62 DefinePersonnel()

```
void cyberdog_visual_programming_abilityset_py::DefinePersonnel (
    py::object )
```

定义 Personnel 函数调用图:



这是这个函数的调用关系图:



#### 7.2.1.63 DefinePoint()

```
void cyberdog_visual_programming_abilityset_py::DefinePoint (py::object )
```

定义 Point 这是这个函数的调用关系图:



#### 7.2.1.64 DefinePose()

```
void cyberdog_visual_programming_abilityset_py::DefinePose (py::object )
```

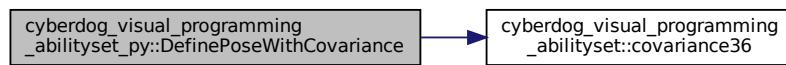
定义 Pose 这是这个函数的调用关系图:



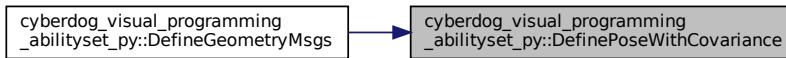
### 7.2.1.65 DefinePoseWithCovariance()

```
void cyberdog_visual_programming_abilityset_py::DefinePoseWithCovariance (
    py::object    )
```

定义 PoseWithCovariance 函数调用图:



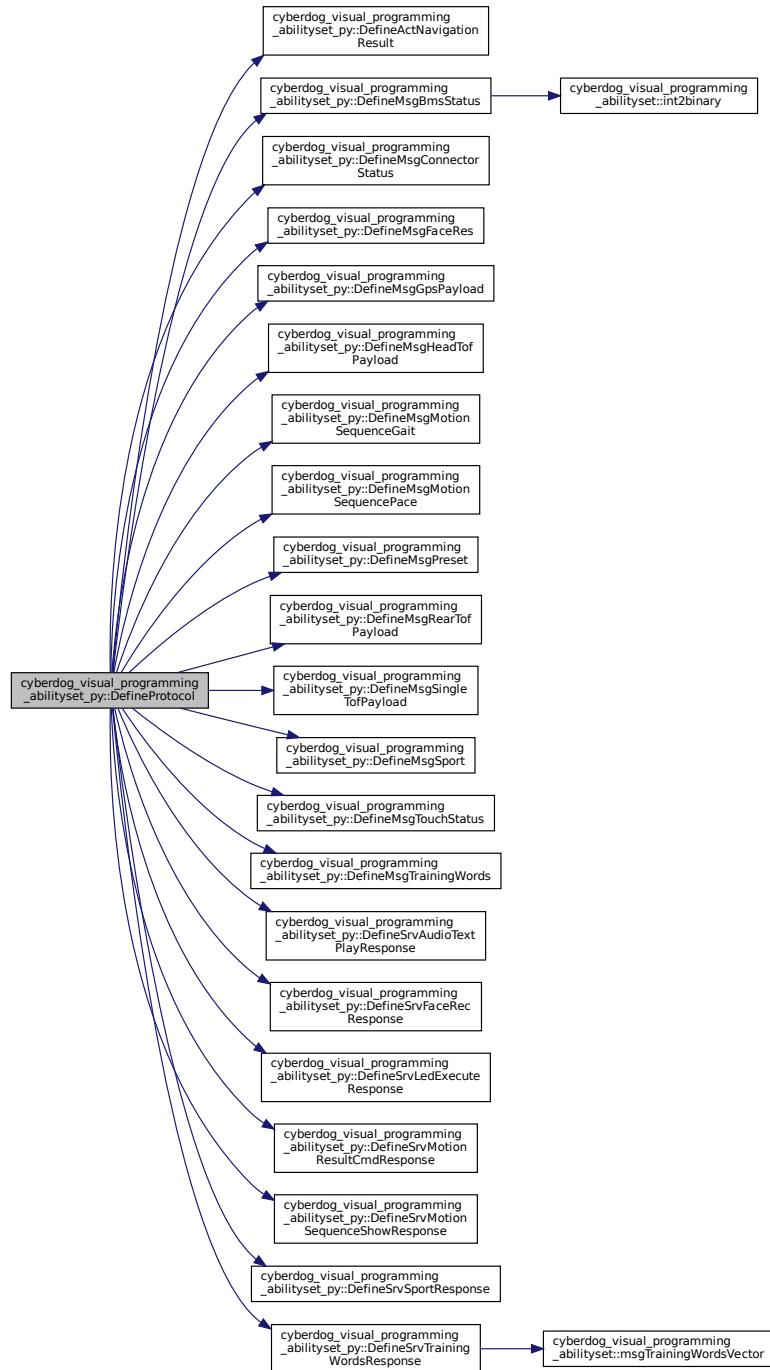
这是这个函数的调用关系图:



### 7.2.1.66 DefineProtocol()

```
void cyberdog_visual_programming_abilityset_py::DefineProtocol (
    py::object    )
```

定义 protocol 函数调用图:



### 7.2.1.67 DefineQuaternion()

```
void cyberdog_visual_programming_abilityset_py::DefineQuaternion (
    py::object )

```

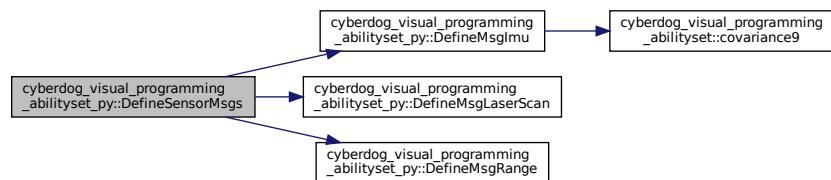
定义 Quaternion 这是这个函数的调用关系图:



### 7.2.1.68 DefineSensorMsgs()

```
void cyberdog_visual_programming_abilityset_py::DefineSensorMsgs (
    py::object    )
```

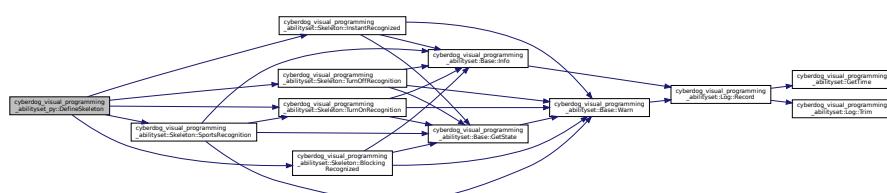
定义 sensor\_msgs 函数调用图:



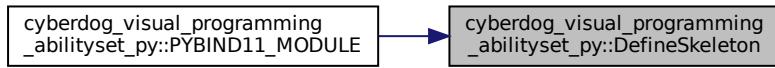
### 7.2.1.69 DefineSkeleton()

```
void cyberdog_visual_programming_abilityset_py::DefineSkeleton (
    py::object    )
```

定义 Skeleton 函数调用图:



这是这个函数的调用关系图:



#### 7.2.1.70 DefineSkeletonRecognizedMessageResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSkeletonRecognizedMessageResponse (py::object )
```

定义 SkeletonRecognizedMessageResponse

#### 7.2.1.71 DefineSkeletonRecognizedSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSkeletonRecognizedSeviceResponse (py::object )
```

定义 SkeletonRecognizedSeviceResponse

#### 7.2.1.72 DefineSkeletonType()

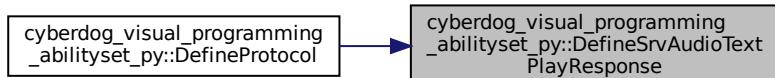
```
void cyberdog_visual_programming_abilityset_py::DefineSkeletonType (py::object )
```

定义 SkeletonType

#### 7.2.1.73 DefineSrvAudioTextPlayResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvAudioTextPlayResponse (py::object )
```

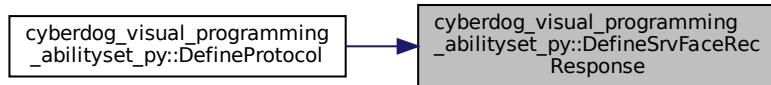
定义 SrvAudioTextPlayResponse 这是这个函数的调用关系图:



### 7.2.1.74 DefineSrvFaceRecResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvFaceRecResponse (py::object )
```

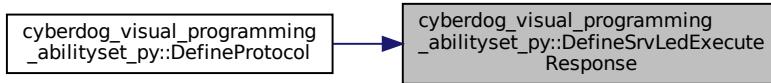
定义 SrvFaceRecResponse 这是这个函数的调用关系图:



### 7.2.1.75 DefineSrvLedExecuteResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvLedExecuteResponse (py::object )
```

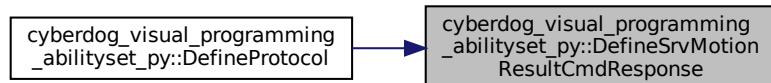
定义 SrvLedExecuteResponse 这是这个函数的调用关系图:



### 7.2.1.76 DefineSrvMotionResultCmdResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvMotionResultCmdResponse (py::object )
```

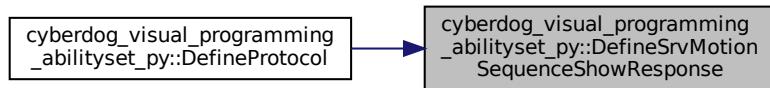
定义 SrvMotionResultCmdResponse 这是这个函数的调用关系图:



### 7.2.1.77 DefineSrvMotionSequenceShowResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvMotionSequenceShowResponse (py::object )
```

定义 SrvMotionSequenceShowResponse 这是这个函数的调用关系图:



### 7.2.1.78 DefineSrvPersonnelResponse()

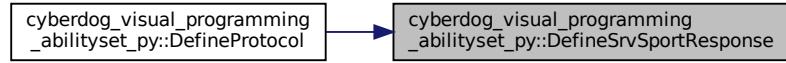
```
void cyberdog_visual_programming_abilityset_py::DefineSrvPersonnelResponse (py::object )
```

定义 SrvPersonnelResponse

### 7.2.1.79 DefineSrvSportResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvSportResponse (py::object )
```

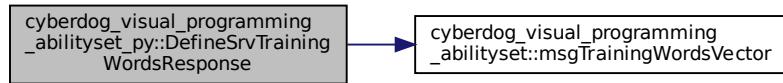
定义 SrvSportResponse 这是这个函数的调用关系图:



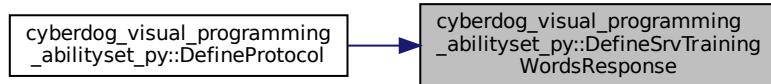
### 7.2.1.80 DefineSrvTrainingWordsResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineSrvTrainingWordsResponse (
    py::object    )
```

定义 SrvTrainingWordsResponse 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.81 DefineState()

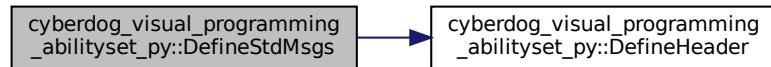
```
void cyberdog_visual_programming_abilityset_py::DefineState (
    py::object    )
```

定义 State

### 7.2.1.82 DefineStdMsgs()

```
void cyberdog_visual_programming_abilityset_py::DefineStdMsgs (
    py::object    )
```

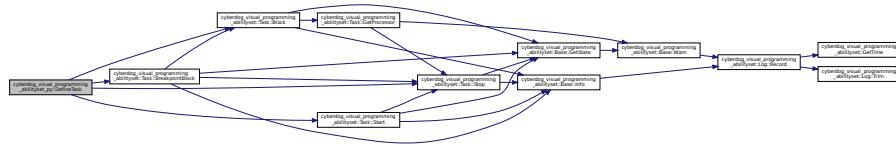
定义 std\_msgs 函数调用图:



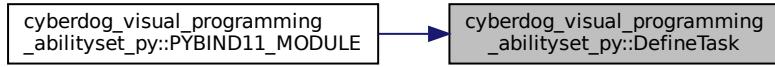
### 7.2.1.83 DefineTask()

```
void cyberdog_visual_programming_abilityset_py::DefineTask (
    py::object    )
```

## 定义 Task 函数调用图：



这是这个函数的调用关系图:



#### 7.2.1.84 DefineTimer()

```
void cyberdog_visual_programming_abilityset.py::DefineTimer (py::object )
```

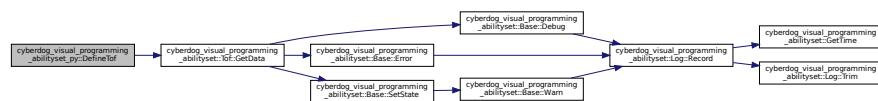
定义 Timer 这是这个函数的调用关系图:



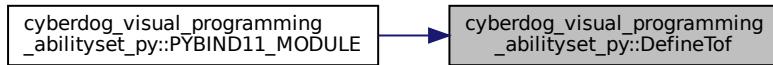
### 7.2.1.85 DefineTof()

```
void cyberdog_visual_programming_abilityset_py::DefineTof (
    py::object    )
```

定义 Tof 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.86 DefineTofObstacle()

```
void cyberdog_visual_programming_abilityset_py::DefineTofObstacle (
    py::object    )
```

定义 MotionSequence

### 7.2.1.87 DefineTofPayload()

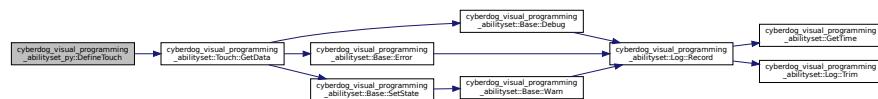
```
void cyberdog_visual_programming_abilityset_py::DefineTofPayload (
    py::object    )
```

定义 TofPayload

### 7.2.1.88 DefineTouch()

```
void cyberdog_visual_programming_abilityset_py::DefineTouch (
    py::object )
```

定义 Touch 函数调用图:



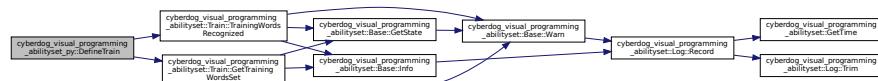
这是这个函数的调用关系图:



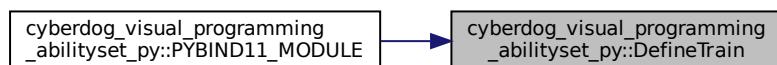
### 7.2.1.89 DefineTrain()

```
void cyberdog_visual_programming_abilityset_py::DefineTrain (
    py::object )
```

定义 Train 函数调用图:



这是这个函数的调用关系图:



### 7.2.1.90 DefineTrainingWordsRecognizedMessageResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineTrainingWordsRecognizedMessageResponse (py::object )
```

定义 TrainingWordsRecognizedMessag...

### 7.2.1.91 DefineTrainingWordsRecognizedSeviceResponse()

```
void cyberdog_visual_programming_abilityset_py::DefineTrainingWordsRecognizedSeviceResponse (py::object )
```

定义 TrainingWordsRecognizedSevice...

### 7.2.1.92 DefineTwist()

```
void cyberdog_visual_programming_abilityset_py::DefineTwist (py::object )
```

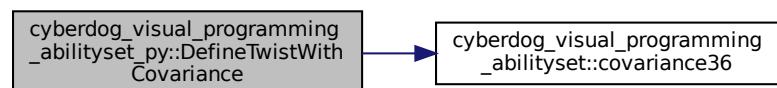
定义 Twist 这是这个函数的调用关系图:



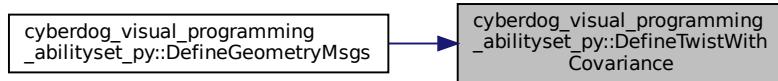
### 7.2.1.93 DefineTwistWithCovariance()

```
void cyberdog_visual_programming_abilityset_py::DefineTwistWithCovariance (py::object )
```

定义 TwistWithCovariance 函数调用图:



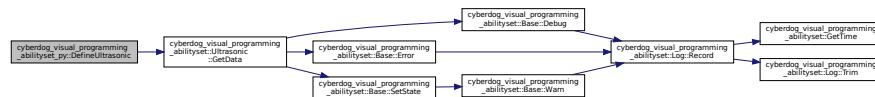
这是这个函数的调用关系图:



#### 7.2.1.94 DefineUltrasonic()

```
void cyberdog_visual_programming_abilityset_py::DefineUltrasonic (
    py::object )
```

定义 Ultrasonic 函数调用图:



这是这个函数的调用关系图:



#### 7.2.1.95 DefineVector3()

```
void cyberdog_visual_programming_abilityset_py::DefineVector3 (
    py::object )
```

定义 Vector3 这是这个函数的调用关系图:



### 7.2.1.96 DefineVoiceprintRecognizedResponse()

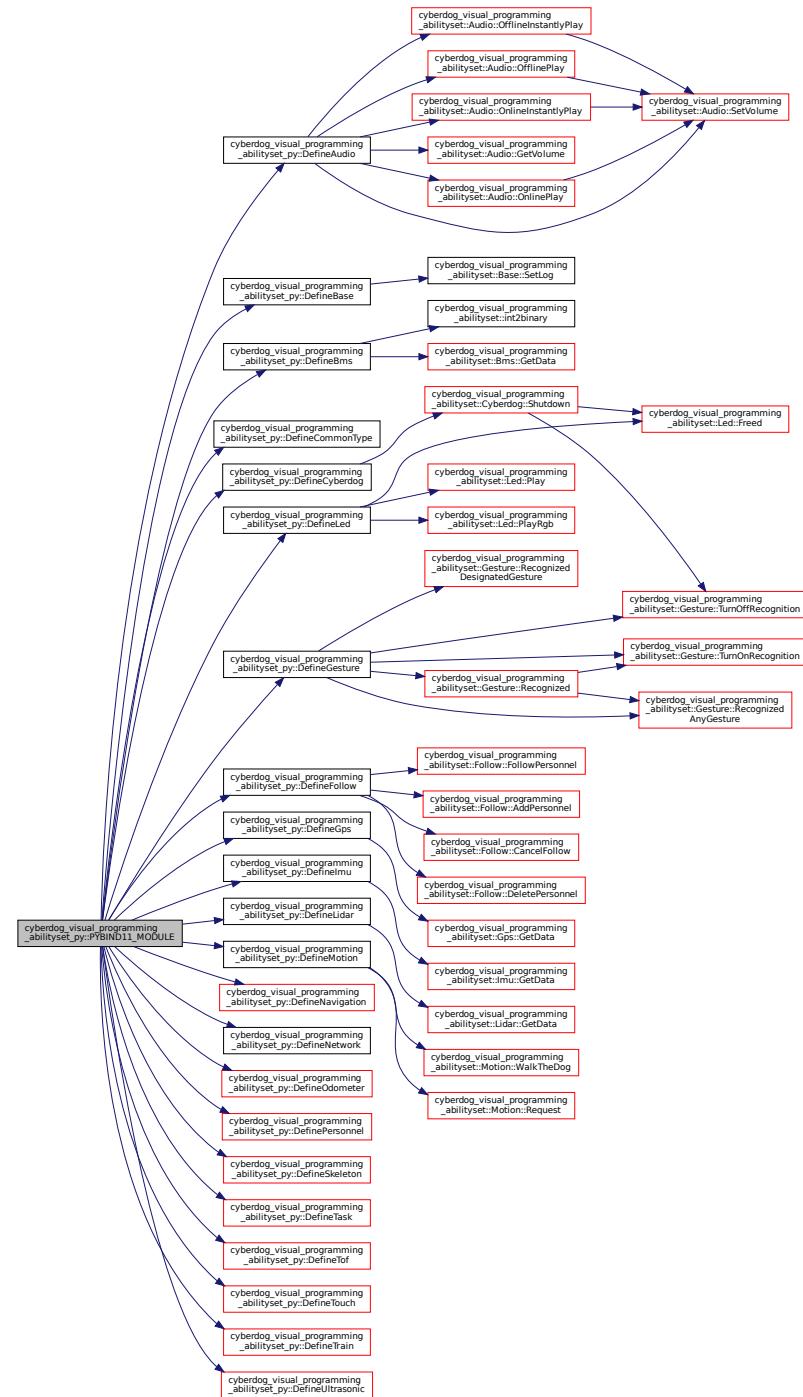
```
void cyberdog_visual_programming_abilityset_py::DefineVoiceprintRecognizedResponse (
    py::object    )
```

定义 VoiceprintRecognizedResponse

### 7.2.1.97 PYBIND11\_MODULE()

```
cyberdog_visual_programming_abilityset_py::PYBIND11_MODULE (
    abilityset ,
    m   )
```

函数调用图:



## 7.3 cyberdog\_visual\_programming\_engine 命名空间参考

类

- class [ArtificialIntelligence](#)
- class [BackendMessage](#)

- struct `BackendMsg`
- class `Base`
- class `FrontendMessage`
- class `Fsm`
- class `Interact`
- class `LinuxAt`
- class `LinuxCron`
- class `Module`
- class `PythonInterpreter`
- class `Slam`
- class `Task`

## 类型定义

- using `CyberdogJson` = `cyberdog::common::CyberdogJson`
- using `TrainingWordsMsg` = `protocol::msg::TrainPlan`
- using `AudioPlayExtendMsg` = `protocol::msg::AudioPlayExtend`
- using `PersonnelMsg` = `protocol::msg::UserInformation`
- using `PresetMsg` = `protocol::msg::Label`
- using `GRPCMMsg` = `std_msgs::msg::String`
- using `ASRMsg` = `protocol::msg::TrainPlan`
- using `OperateMsg` = `protocol::msg::VisualProgrammingOperate`
- using `OperateSrv` = `protocol::srv::VisualProgrammingOperate`
- using `PersonnelSrv` = `protocol::srv::AllUserSearch`
- using `TrainingWordsSrv` = `protocol::srv::TrainPlanAll`
- using `PresetSrv` = `protocol::srv::GetMapLabel`
- using `HTPSrv` = `protocol::srv::BesHttp`
- using `ShellEnum` = `ShellEnum_`
- using `TimeMode` = `TimeMode_`

## 枚举

- enum `ShellEnum_` { `shell` = 1993, `command`, `command_popen`, `command_error` }
- enum `TimeMode_` { `Ms1970` = 0, `_Y_M_D_H_M_S`, `STANDARD` }
- enum `StateEnum` {
 `normally` = 0, `abnormally_mkdir_path` = 21, `abnormally_open_file`, `abnormally_build`,  
`abnormally_register`, `abnormally_update_list`, `abnormally_perform`, `abnormally_operate`,  
`abnormally_request`, `abnormally_other_errors`, `service_request_interrupted`, `service_appear_timeout`,  
`service_request_timeout` }

## 函数

- std::string `GetTime` (int nowModo=0)
- bool `Shell` (const std::string &, int &, std::string &)
- std::vector< std::string > `GetVector` (const std::string &, char, const std::string & \_head="")
- bool `JudgeConfFile` (std::string \_file)
- bool `GetWorkspace` (std::string &)
- bool `Mkdir` (std::string &)
- std::string `Subreplace` (const std::string &, const std::string &, const std::string &)

## 变量

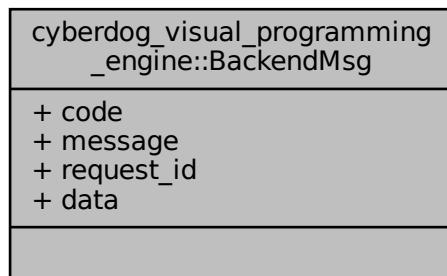
- py::object `get_module_header`
- py::object `get_task_header`
- py::object `generate_derivative_file`
- py::object `IPythonDemo`

### 7.3.1 类说明

#### 7.3.1.1 struct `cyberdog_visual_programming_engine::BackendMsg`

参数约束:后端消息

`cyberdog_visual_programming_engine::BackendMsg` 的协作图:



## 类成员

string	code	状态码
vector< string >	data	数据
string	message	消息
string	request_id	请求id

### 7.3.2 类型定义说明

#### 7.3.2.1 ASRMsg

```
using cyberdog_visual_programming_engine::ASRMsg = typedef protocol::msg::TrainPlan
```

[topic 类型]自动语音识别 消息

### 7.3.2.2 `AudioPlayExtendMsg`

```
using cyberdog_visual_programming_engine::AudioPlayExtendMsg = typedef protocol::msg::Audio<→  
PlayExtend
```

语音消息:在线

### 7.3.2.3 `CyberdogJson`

```
using cyberdog_visual_programming_engine::CyberdogJson = typedef cyberdog::common::CyberdogJson  
json 解析类型
```

### 7.3.2.4 `GRPCMsg`

```
using cyberdog_visual_programming_engine::GRPCMsg = typedef std_msgs::msg::String  
[topic 类型]GRPC 消息
```

### 7.3.2.5 `HTPSrv`

```
using cyberdog_visual_programming_engine::HTPSrv = typedef protocol::srv::BesHttp  
[service 类型]后端通信
```

### 7.3.2.6 `OperateMsg`

```
using cyberdog_visual_programming_engine::OperateMsg = typedef protocol::msg::VisualProgramming<→  
Operate
```

[topic 类型]任务操作

### 7.3.2.7 `OperateSrv`

```
using cyberdog_visual_programming_engine::OperateSrv = typedef protocol::srv::VisualProgramming<→  
Operate  
[service 类型]任务操作
```

### 7.3.2.8 `PersonnelMsg`

```
using cyberdog_visual_programming_engine::PersonnelMsg = typedef protocol::msg::UserInformation  
人员信息
```

### 7.3.2.9 PersonnelSrv

```
using cyberdog_visual_programming_engine::PersonnelSrv = typedef protocol::srv::AllUserSearch  
[service 类型]人员底库
```

### 7.3.2.10 PresetMsg

```
using cyberdog_visual_programming_engine::PresetMsg = typedef protocol::msg::Label  
[topic 类型]预置点消息
```

### 7.3.2.11 PresetSrv

```
using cyberdog_visual_programming_engine::PresetSrv = typedef protocol::srv::GetMapLabel  
[service 类型]预置点
```

### 7.3.2.12 ShellEnum

```
using cyberdog_visual_programming_engine::ShellEnum = typedef ShellEnum_  
shell 接口返回值合法类型
```

### 7.3.2.13 TimeMode

```
using cyberdog_visual_programming_engine::TimeMode = typedef TimeMode_  
时间获取接口入参类型
```

### 7.3.2.14 TrainingWordsMsg

```
using cyberdog_visual_programming_engine::TrainingWordsMsg = typedef protocol::msg::TrainPlan  
训练词消息
```

### 7.3.2.15 TrainingWordsSrv

```
using cyberdog_visual_programming_engine::TrainingWordsSrv = typedef protocol::srv::TrainPlanAll  
[service 类型]训练词集合
```

## 7.3.3 枚举类型说明

### 7.3.3.1 ShellEnum\_

```
enum cyberdog_visual_programming_engine::ShellEnum_
```

约束调用 shell 接口时的异常返回值的合法值

枚举值

shell	异常结束, 执行shell命令错误
command	异常结束, 待执行命令错误
command_popen	异常结束, 无法执行命令
command_error	异常结束, 执行命令失败

### 7.3.3.2 StateEnum

```
enum cyberdog_visual_programming_engine::StateEnum
```

通用状态约束

枚举值

normally	正常结束
abnormally_mkdir_path	21:异常结束, 无法创建路径
abnormally_open_file	22:异常结束, 无法打开文件
abnormally_build	23:异常结束, 无法构建
abnormally_register	24:异常结束, 无法注册
abnormally_update_list	25:异常结束, 无法更新列表
abnormally_perform	26:异常结束, 无法执行
abnormally_operate	27:异常结束, 当前操作非法
abnormally_request	28:异常结束, 请求错误
abnormally_other_errors	29:异常结束, 其他错误
service_request_interrupted	30:异常结束, 服务被打断
service_appear_timeout	31:异常结束, 等待服务上线超时
service_request_timeout	32:异常结束, 请求服务超时

### 7.3.3.3 TimeMode

```
enum cyberdog_visual_programming_engine::TimeMode
```

约束获取时间接口入参的合法值

枚举值

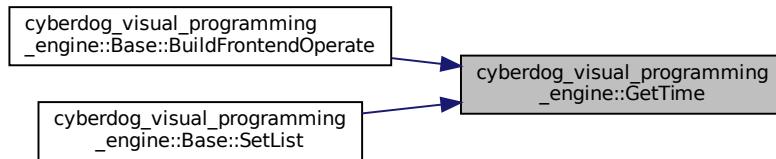
Ms1970	1970年1月1日到现在的时间(毫秒)
_Y_M_D_H_M↔_S	"*Y*M*D*H*M*S"
STANDARD	"*Y.*M.*D-*H:*M::*S"

## 7.3.4 函数说明

### 7.3.4.1 GetTime()

```
std::string cyberdog_visual_programming_engine::GetTime (
    int nowModo = 0 )
```

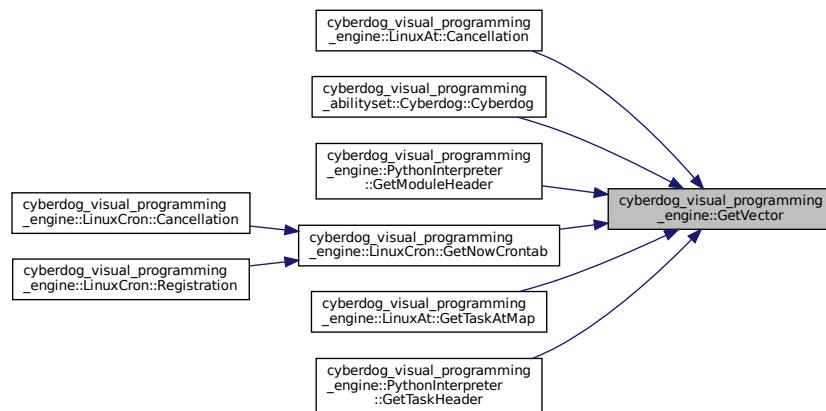
{0:Ms1970, 1:Y\_M\_D\_H\_M\_S, 2:STANDARD} 获取时间戳 这是这个函数的调用关系图:



### 7.3.4.2 GetVector()

```
std::vector< std::string > cyberdog_visual_programming_engine::GetVector (
    const std::string & _message,
    char _delim,
    const std::string & _head = "" )
```

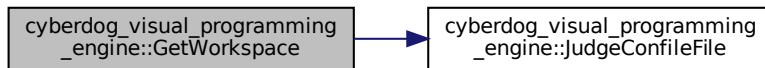
获取向量 这是这个函数的调用关系图:



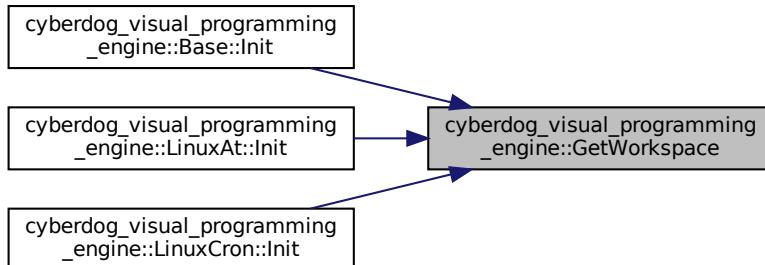
### 7.3.4.3 GetWorkspace()

```
bool cyberdog_visual_programming_engine::GetWorkspace ( std::string & _workspace )
```

获取工作空间 函数调用图:



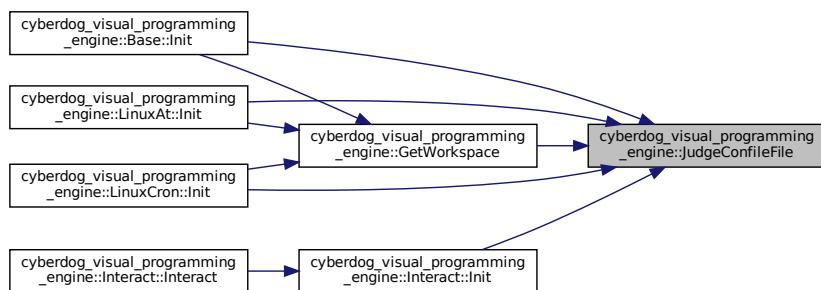
这是这个函数的调用关系图:



### 7.3.4.4 JudgeConfFile()

```
bool cyberdog_visual_programming_engine::JudgeConfFile ( std::string _file )
```

判断配置文件权限 这是这个函数的调用关系图:



### 7.3.4.5 Mkdir()

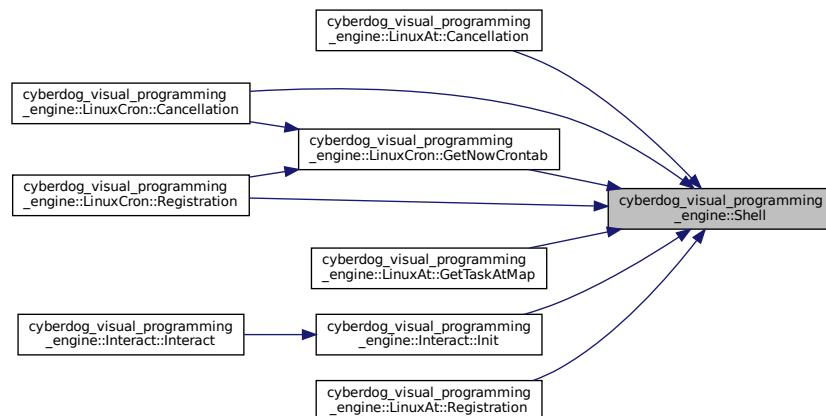
```
bool cyberdog_visual_programming_engine::Mkdir (
    std::string & _tar_path )
```

创建文件夹

### 7.3.4.6 Shell()

```
bool cyberdog_visual_programming_engine::Shell (
    const std::string & _command,
    int & _code,
    std::string & _message )
```

执行shell 这是这个函数的调用关系图:



### 7.3.4.7 Subreplace()

```
std::string cyberdog_visual_programming_engine::Subreplace (
    const std::string & resource_str,
    const std::string & sub_str,
    const std::string & new_str )
```

对字符串中所有指定的子串进行替换

## 7.3.5 变量说明

### 7.3.5.1 `generate_derivative_file`

```
py::object cyberdog_visual_programming_engine::generate_derivative_file
```

### 7.3.5.2 `get_module_header`

```
py::object cyberdog_visual_programming_engine::get_module_header
```

### 7.3.5.3 `get_task_header`

```
py::object cyberdog_visual_programming_engine::get_task_header
```

### 7.3.5.4 `IPythonDemo`

```
py::object cyberdog_visual_programming_engine::IPythonDemo
```

## 7.4 `cyberdog_visual_programming_terminal` 命名空间参考

### 类

- class `DebugAbilityset`
- class `DebugEngine`
- class `Debugger`
- class `Interface`
- class `Visual`

### 类型定义

- using `CyberdogJson` = `cyberdog::common::CyberdogJson`
- using `CyberdogToml` = `cyberdog::common::CyberdogToml`
- using `GRPCMsg` = `std::msgs::msg::String`
- using `OperateMsg` = `protocol::msg::VisualProgrammingOperate`

### 枚举

- enum `TimeMode` { `Ms1970` = 0, `_Y_M_D_H_M_S`, `STANDARD` }

## 函数

- std::string `GetTime` (int nowModo=0)
- bool `JudgeConfileFile` (std::string \_file)
- std::vector< std::string > `GetVector` (const std::string &, char, const std::string &\_head="")
- void `CoutJson` (const std::string &, const std::string &)

## 变量

- std::mutex `cursor_mutex_`

### 7.4.1 类型定义说明

#### 7.4.1.1 CyberdogJson

```
using cyberdog_visual_programming_terminal::CyberdogJson = typedef cyberdog::common::CyberdogJson
```

json 解析类型

#### 7.4.1.2 CyberdogToml

```
using cyberdog_visual_programming_terminal::CyberdogToml = typedef cyberdog::common::CyberdogToml
```

Toml 解析及构建模块类型

#### 7.4.1.3 GRPCMsg

```
using cyberdog_visual_programming_terminal::GRPCMsg = typedef std_msgs::msg::String
```

[topic 类型]GRPC 消息

#### 7.4.1.4 OperateMsg

```
using cyberdog_visual_programming_terminal::OperateMsg = typedef protocol::msg::VisualProgrammingOperate
```

[topic 类型]任务操作

### 7.4.2 枚举类型说明

#### 7.4.2.1 TimeMode

```
enum cyberdog_visual_programming_terminal::TimeMode
```

约束获取时间接口入参的合法值

枚举值

Ms1970	1970年1月1日到现在的时间(毫秒)
_Y_M_D_H_M↔_S	"*Y*M*D*H*M*S"
STANDARD	"*Y.*M.*D-*H:*M:*S"

### 7.4.3 函数说明

#### 7.4.3.1 CoutJson()

```
void cyberdog_visual_programming_terminal::CoutJson (
    const std::string & _msg_head,
    const std::string & _msg )
```

输出Json消息

#### 7.4.3.2 GetTime()

```
std::string cyberdog_visual_programming_terminal::GetTime (
    int nowModo = 0 )
```

获取时间戳

#### 7.4.3.3 GetVector()

```
std::vector< std::string > cyberdog_visual_programming_terminal::GetVector (
    const std::string & _message,
    char _delim,
    const std::string & _head = "" )
```

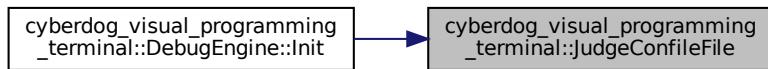
获取向量 这是这个函数的调用关系图:



#### 7.4.3.4 JudgeConfileFile()

```
bool cyberdog_visual_programming_terminal::JudgeConfileFile (
    std::string _file )
```

判断配置文件权限 这是这个函数的调用关系图:



#### 7.4.4 变量说明

##### 7.4.4.1 cursor\_mutex\_

```
std::mutex cyberdog_visual_programming_terminal::cursor_mutex_
```

光标互斥锁

### 7.5 cyberdog\_visual\_programming\_terminal\_py 命名空间参考

#### 函数

- void DefineDebugAbilityset (py::object m)
- void DefineDebugEngine (py::object m)
- void DefineDebugger (py::object m)
- void DefineInterfaceNetwork (py::object m)
- void DefineInterfaceFollow (py::object m)
- void DefineInterfaceMotion (py::object m)
- void DefineInterfaceNavigation (py::object m)
- void DefineInterfaceTask (py::object m)
- void DefineInterfaceTrain (py::object m)
- void DefineInterfacePersonnel (py::object m)
- void DefineInterfaceGesture (py::object m)
- void DefineInterfaceSkeleton (py::object m)
- void DefineInterfaceBms (py::object m)
- void DefineInterfaceLed (py::object m)
- void DefineInterfaceAudio (py::object m)
- void DefineInterfaceTouch (py::object m)
- void DefineInterfaceGps (py::object m)
- void DefineInterfaceTof (py::object m)
- void DefineInterfaceLidar (py::object m)
- void DefineInterfaceUltrasonic (py::object m)

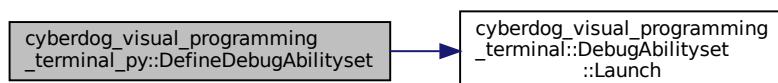
- void DefineInterfaceOdometer (py::object m)
- void DefineInterfaceImu (py::object m)
- void DefineInterfaceCyberdog (py::object m)
- void DefineInterfaceVisualInterface (py::object m)
- void DefineInterfaceVisualDebugAbilityset (py::object m)
- void DefineInterfaceVisualDebuggerEngine (py::object m)
- void DefineInterfaceVisualDebugger (py::object m)
- void DefineInterfaceVisual (py::object m)
- void DefineInterfaceTypeEnum (py::object m)
- void DefineInterfaceTypeClass (py::object m)
- void DefineInterfaceType (py::object m)
- void DefineInterface (py::object m)
- PYBIND11\_MODULE (terminal, m)
- void DefineVisual (py::object m)

## 7.5.1 函数说明

### 7.5.1.1 DefineDebugAbilityset()

```
void cyberdog_visual_programming_terminal_py::DefineDebugAbilityset (py::object )
```

定义 DebugAbilityset 函数调用图:



这是这个函数的调用关系图:



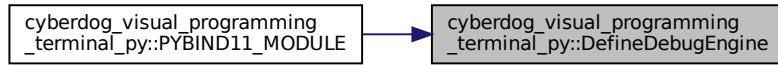
### 7.5.1.2 DefineDebugEngine()

```
void cyberdog_visual_programming_terminal_py::DefineDebugEngine (
    py::object    )
```

定义 DebugEngine 函数调用图:



这是这个函数的调用关系图:



### 7.5.1.3 DefineDebugger()

```
void cyberdog_visual_programming_terminal_py::DefineDebugger (
    py::object    )
```

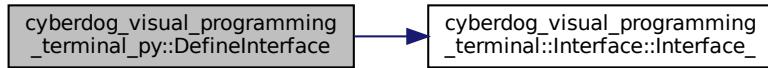
定义 Debugger 这是这个函数的调用关系图:



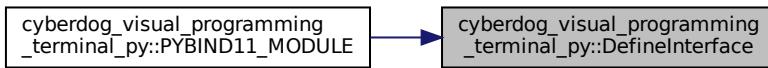
#### 7.5.1.4 DefineInterface()

```
void cyberdog_visual_programming_terminal_py::DefineInterface (
    py::object    )
```

定义 Interface 函数调用图:



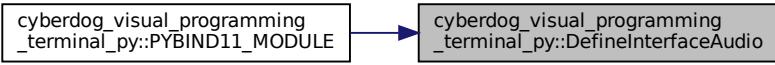
这是这个函数的调用关系图:



#### 7.5.1.5 DefineInterfaceAudio()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceAudio (
    py::object    )
```

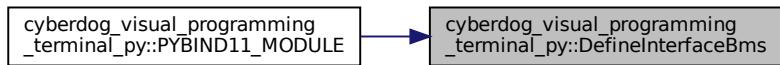
定义 Audio 这是这个函数的调用关系图:



### 7.5.1.6 DefineInterfaceBms()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceBms (py::object )
```

定义 Bms 这是这个函数的调用关系图:



### 7.5.1.7 DefineInterfaceCyberdog()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceCyberdog (py::object )
```

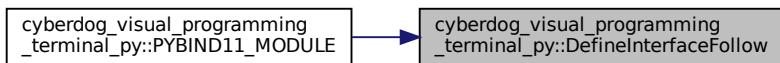
定义 InterfaceCyberdog 这是这个函数的调用关系图:



### 7.5.1.8 DefineInterfaceFollow()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceFollow (py::object )
```

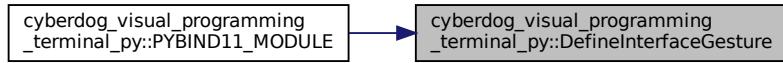
定义 Follow 这是这个函数的调用关系图:



### 7.5.1.9 DefineInterfaceGesture()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceGesture (py::object )
```

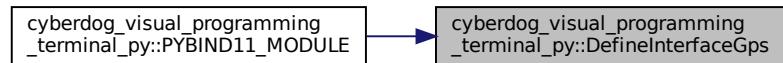
定义 Gesture 这是这个函数的调用关系图:



### 7.5.1.10 DefineInterfaceGps()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceGps (py::object )
```

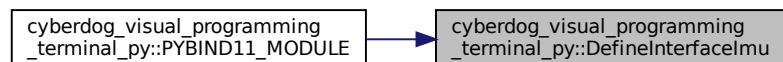
定义 Gps 这是这个函数的调用关系图:



### 7.5.1.11 DefineInterfaceImu()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceImu (py::object )
```

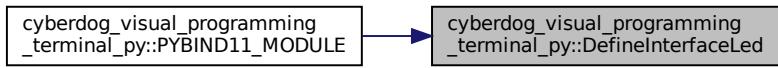
定义 Imu 这是这个函数的调用关系图:



### 7.5.1.12 DefineInterfaceLed()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceLed (py::object )
```

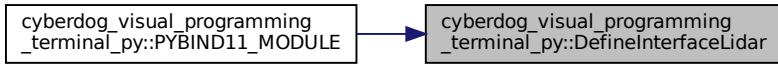
定义 Led 这是这个函数的调用关系图:



### 7.5.1.13 DefineInterfaceLidar()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceLidar (py::object )
```

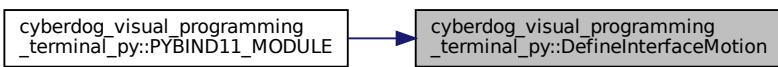
定义 Lidar 这是这个函数的调用关系图:



### 7.5.1.14 DefineInterfaceMotion()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceMotion (py::object )
```

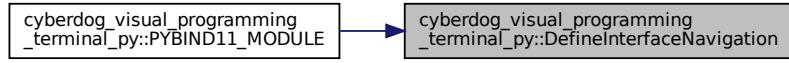
定义 Motion 这是这个函数的调用关系图:



### 7.5.1.15 DefineInterfaceNavigation()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceNavigation (
    py::object )
```

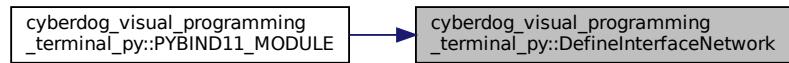
定义 Navigation 这是这个函数的调用关系图:



### 7.5.1.16 DefineInterfaceNetwork()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceNetwork (
    py::object )
```

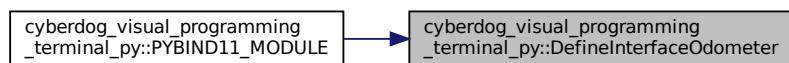
定义 Network 这是这个函数的调用关系图:



### 7.5.1.17 DefineInterfaceOdometer()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceOdometer (
    py::object )
```

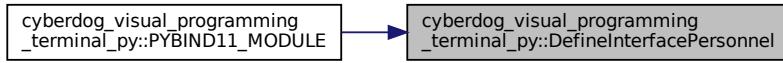
定义 Odometer 这是这个函数的调用关系图:



### 7.5.1.18 DefineInterfacePersonnel()

```
void cyberdog_visual_programming_terminal_py::DefineInterfacePersonnel (py::object )
```

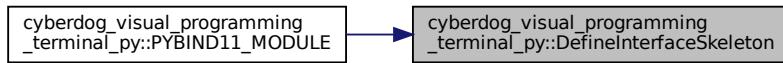
定义 Personnel 这是这个函数的调用关系图:



### 7.5.1.19 DefineInterfaceSkeleton()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceSkeleton (py::object )
```

定义 Skeleton 这是这个函数的调用关系图:



### 7.5.1.20 DefineInterfaceTask()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTask (py::object )
```

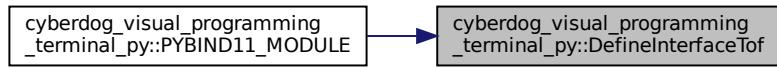
定义 Task 这是这个函数的调用关系图:



### 7.5.1.21 DefineInterfaceTof()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTof (py::object )
```

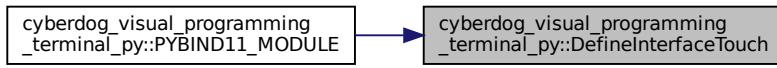
定义 Tof 这是这个函数的调用关系图:



### 7.5.1.22 DefineInterfaceTouch()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTouch (py::object )
```

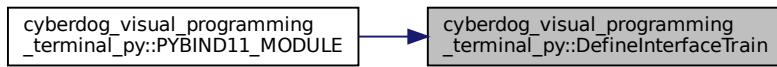
定义 Touch 这是这个函数的调用关系图:



### 7.5.1.23 DefineInterfaceTrain()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTrain (py::object )
```

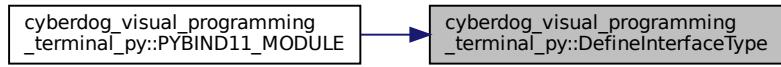
定义 Train 这是这个函数的调用关系图:



### 7.5.1.24 DefineInterfaceType()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceType (py::object )
```

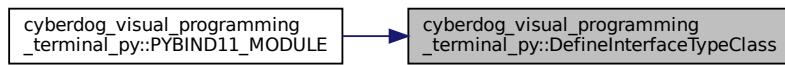
定义 InterfaceType 这是这个函数的调用关系图:



### 7.5.1.25 DefineInterfaceTypeClass()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTypeClass (py::object )
```

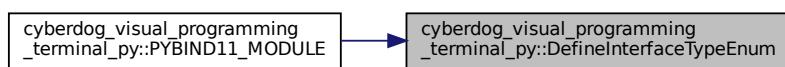
定义 InterfaceTypeClass 这是这个函数的调用关系图:



### 7.5.1.26 DefineInterfaceTypeEnum()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceTypeEnum (py::object )
```

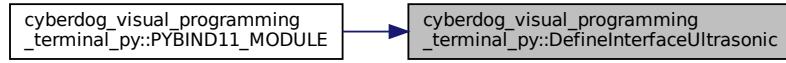
定义 InterfaceTypeEnum 这是这个函数的调用关系图:



### 7.5.1.27 DefineInterfaceUltrasonic()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceUltrasonic (py::object )
```

定义 Ultrasonic 这是这个函数的调用关系图:



### 7.5.1.28 DefineInterfaceVisual()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceVisual (py::object )
```

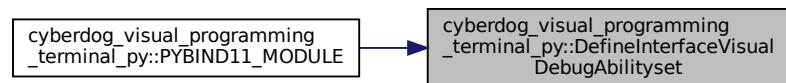
定义 InterfaceVisual 这是这个函数的调用关系图:



### 7.5.1.29 DefineInterfaceVisualDebugAbilityset()

```
void cyberdog_visual_programming_terminal.py::DefineInterfaceVisualDebugAbilityset (py::object )
```

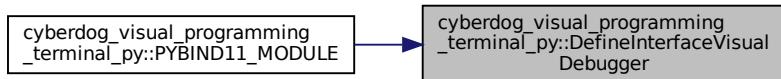
定义 InterfaceVisualDebugAbilityset 这是这个函数的调用关系图:



### 7.5.1.30 DefineInterfaceVisualDebugger()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceVisualDebugger (py::object )
```

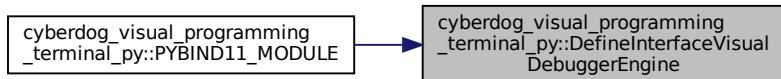
定义 InterfaceVisualDebugger 这是这个函数的调用关系图:



### 7.5.1.31 DefineInterfaceVisualDebuggerEngine()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceVisualDebuggerEngine (py::object )
```

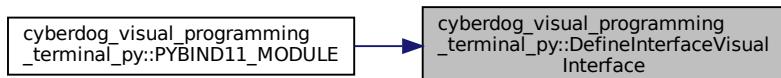
定义 InterfaceVisualDebuggerEngine 这是这个函数的调用关系图:



### 7.5.1.32 DefineInterfaceVisualInterface()

```
void cyberdog_visual_programming_terminal_py::DefineInterfaceVisualInterface (py::object )
```

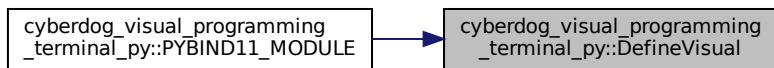
定义 InterfaceVisualInterface 这是这个函数的调用关系图:



### 7.5.1.33 DefineVisual()

```
void cyberdog_visual_programming_terminal_py::DefineVisual (
    py::object    )
```

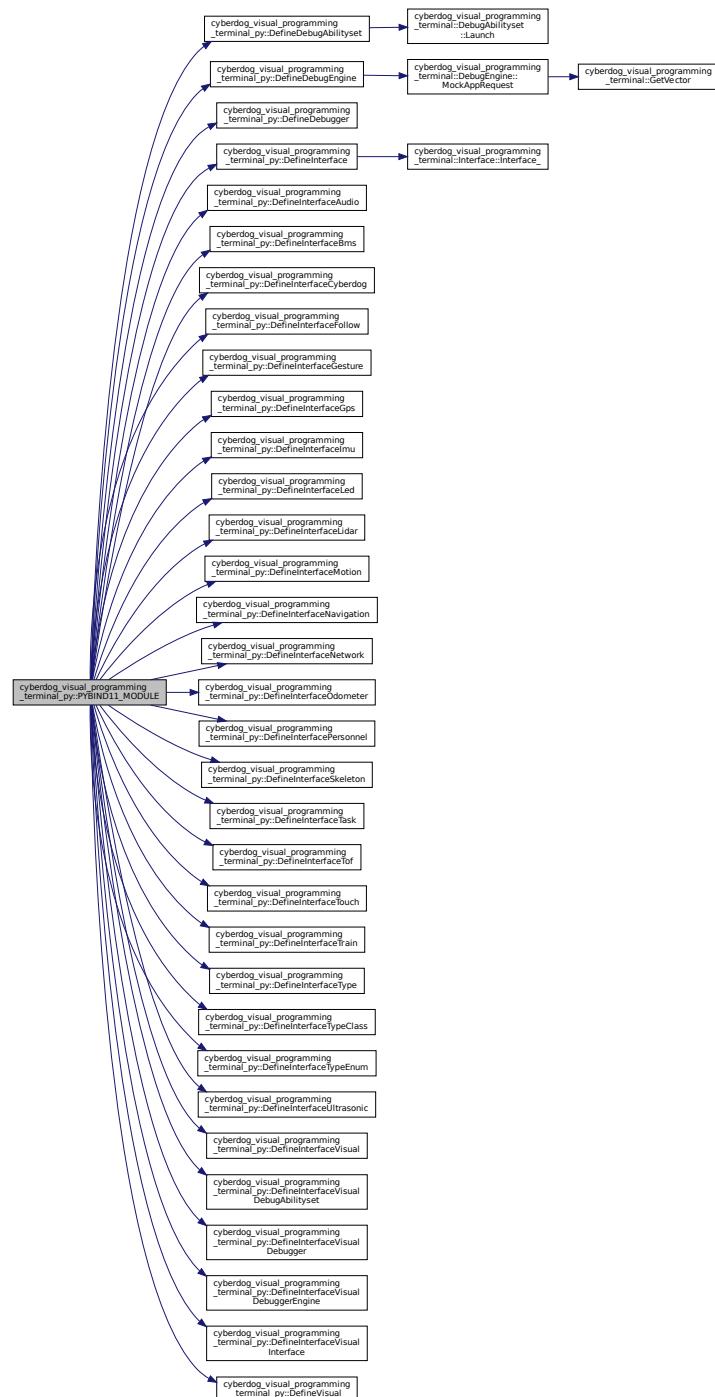
定义 Visual 这是这个函数的调用关系图:



### 7.5.1.34 PYBIND11\_MODULE()

```
cyberdog_visual_programming_terminal_py::PYBIND11_MODULE (
    terminal ,
    m    )
```

函数调用图:



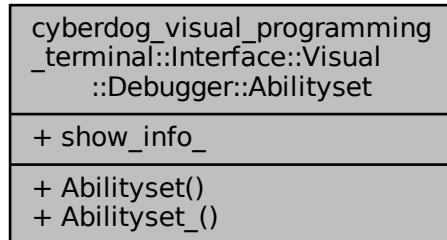
# Chapter 8

## 类说明

### 8.1 `cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset` 结构体 参考

```
#include <interface.hpp>
```

`cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset` 的协作图:



#### Public 成员函数

- `Abilityset()`
- `void Abilityset_(const std::string &fun)`

#### Public 属性

- `std::function< void(std::string) > show_info_`

## 8.1.1 构造及析构函数说明

### 8.1.1.1 Abilityset()

```
cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset::Abilityset ()  
[inline]
```

## 8.1.2 成员函数说明

### 8.1.2.1 Abilityset\_()

```
void cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset::Abilityset_<-  
- (  
    const std::string &fun) [inline]
```

## 8.1.3 类成员变量说明

### 8.1.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Visual::<-  
Debugger::Abilityset::show_info_
```

## 8.2 cyberdog\_visual\_programming\_engine::ArtificialIntelligence类 参考

```
#include <artificial_intelligence.hpp>  
cyberdog_visual_programming_engine::ArtificialIntelligence 的协作图:
```

cyberdog_visual_programming _engine::ArtificialIntelligence
+ ArtificialIntelligence() + ~ArtificialIntelligence() + Init() + RespondToRequests()

## Public 成员函数

- `ArtificialIntelligence ()`
- `~ArtificialIntelligence ()`
- `bool Init (const rclcpp::Node::SharedPtr &, const toml::value &)`
- `bool RespondToRequests (const OperateMsg &, GRPCMsg &)`

### 8.2.1 构造及析构函数说明

#### 8.2.1.1 ArtificialIntelligence()

```
cyberdog_visual_programming_engine::ArtificialIntelligence::ArtificialIntelligence ( )
```

#### 8.2.1.2 ~ArtificialIntelligence()

```
cyberdog_visual_programming_engine::ArtificialIntelligence::~ArtificialIntelligence ( )
```

## 8.2.2 成员函数说明

#### 8.2.2.1 Init()

```
bool cyberdog_visual_programming_engine::ArtificialIntelligence::Init (
    const rclcpp::Node::SharedPtr & _node_ptr,
    const toml::value & _params_toml )
```

初始化

#### 8.2.2.2 RespondToRequests()

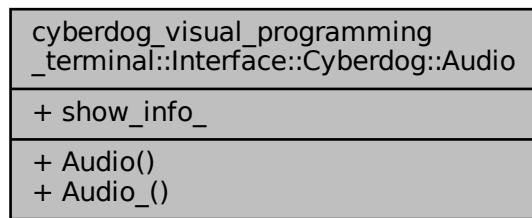
```
bool cyberdog_visual_programming_engine::ArtificialIntelligence::RespondToRequests (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

执行请求

## 8.3 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog:: ::Audio 结构体 参 考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Audio 的协作图:



### Public 成员函数

- `Audio ()`
- void `Audio_ (const std::string &fun)`

### Public 属性

- std::function< void(std::string)> `show_info_`

#### 8.3.1 构造及析构函数说明

##### 8.3.1.1 `Audio()`

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio::Audio ( ) [inline]
```

#### 8.3.2 成员函数说明

##### 8.3.2.1 `Audio_()`

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio::Audio_ (
    const std::string &fun ) [inline]
```

### 8.3.3 类成员变量说明

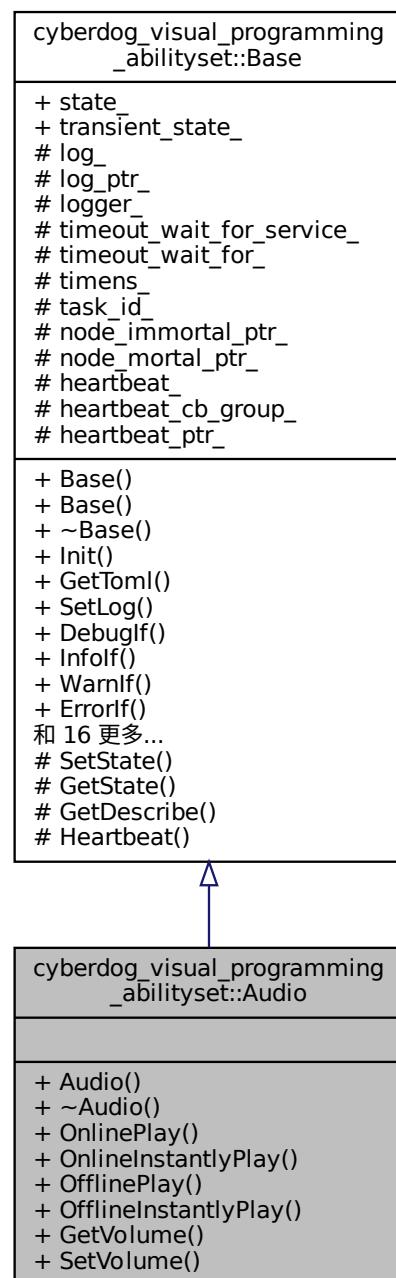
#### 8.3.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←  
Audio::show_info_
```

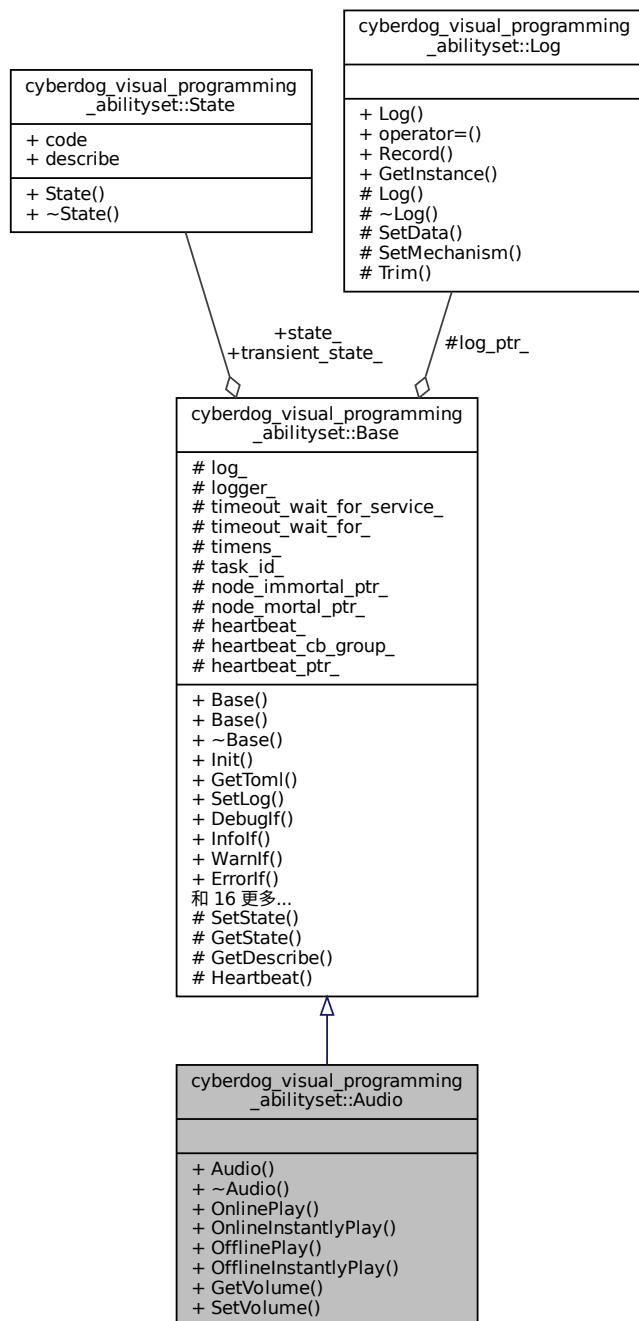
## 8.4 **cyberdog\_visual\_programming\_abilityset::Audio**类 参考

```
#include <audio.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Audio 继承关系图:



cyberdog\_visual\_programming\_abilityset::Audio 的协作图:



## Public 成员函数

- `Audio ()`
- `~Audio ()`
- `AudioPlaySeviceResponse OnlinePlay (const std::string message="我是谁", const int8_t volume=-1)`
- `State OnlineInstantlyPlay (const std::string message="我是谁", const int8_t volume=-1)`
- `AudioPlaySeviceResponse OfflinePlay (const uint16_t type=4000, const int8_t volume=-1)`

- `State OfflineInstantlyPlay (const uint16_t type=4000, const int8_t volume=-1)`
- `AudioGetVolumeSeviceResponse GetVolume ()`
- `AudioSetVolumeSeviceResponse SetVolume (const uint8_t volume)`

额外继承的成员函数

### 8.4.1 构造及析构函数说明

#### 8.4.1.1 Audio()

```
cyberdog_visual_programming_abilityset::Audio::Audio ( ) [inline]
```

#### 8.4.1.2 ~Audio()

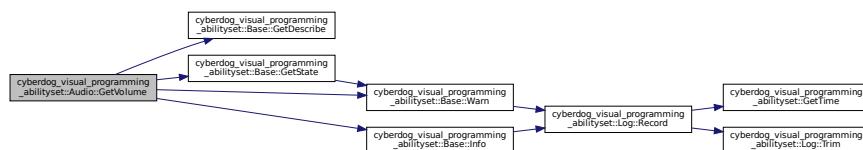
```
cyberdog_visual_programming_abilityset::Audio::~Audio ( ) [inline]
```

### 8.4.2 成员函数说明

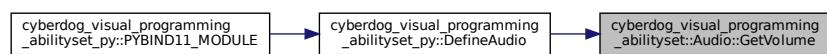
#### 8.4.2.1 GetVolume()

```
AudioGetVolumeSeviceResponse cyberdog_visual_programming_abilityset::Audio::GetVolume ( )
```

获取音量 函数调用图:



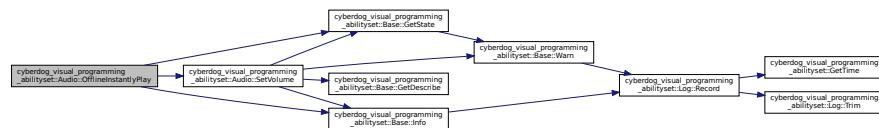
这是这个函数的调用关系图:



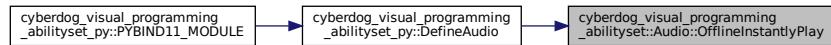
### 8.4.2.2 OfflineInstantlyPlay()

```
State cyberdog_visual_programming_abilityset::Audio::OfflineInstantlyPlay (
    const uint16_t type = 4000,
    const int8_t volume = -1 )
```

离线立即播放音效 函数调用图:



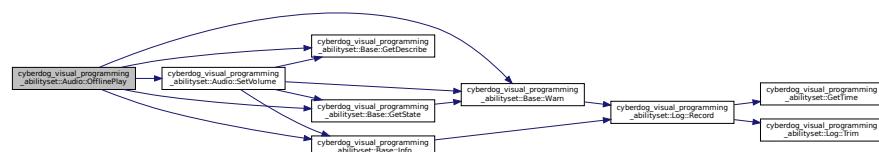
这是这个函数的调用关系图:



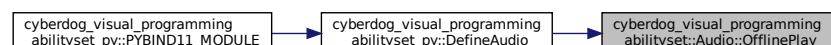
### 8.4.2.3 OfflinePlay()

```
AudioPlaySeviceResponse cyberdog_visual_programming_abilityset::Audio::OfflinePlay (
    const uint16_t type = 4000,
    const int8_t volume = -1 )
```

离线播放音效 函数调用图:



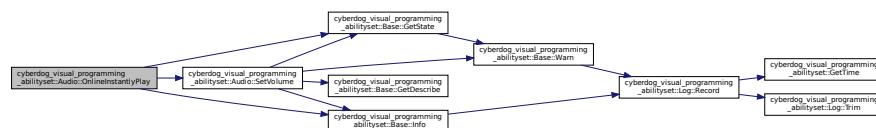
这是这个函数的调用关系图:



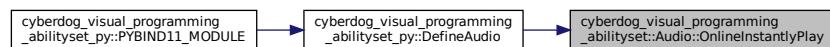
#### 8.4.2.4 OnlineInstantlyPlay()

```
State cyberdog_visual_programming_abilityset::Audio::OnlineInstantlyPlay (
    const std::string message = "我是谁",
    const int8_t volume = -1 )
```

在线立即播放音效 函数调用图:



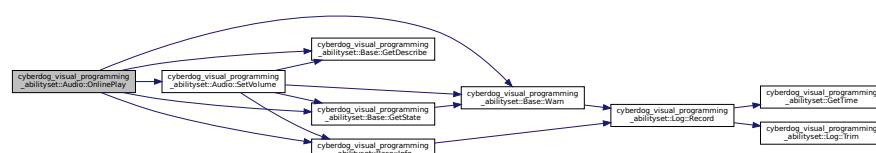
这是这个函数的调用关系图:



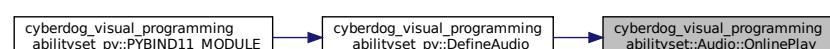
#### 8.4.2.5 OnlinePlay()

```
AudioPlayServiceResponse cyberdog_visual_programming_abilityset::Audio::OnlinePlay (
    const std::string message = "我是谁",
    const int8_t volume = -1 )
```

在线播放音效 函数调用图:



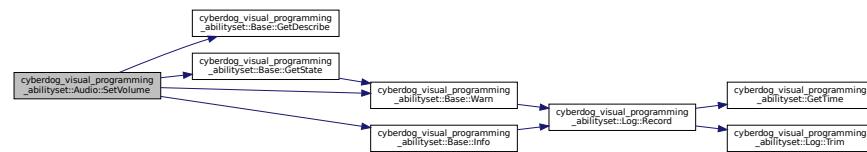
这是这个函数的调用关系图:



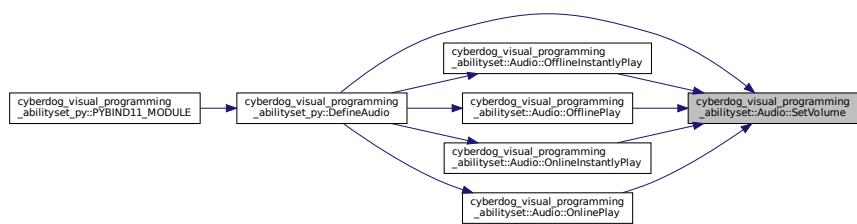
#### 8.4.2.6 SetVolume()

```
AudioSetVolumeSeviceResponse cyberdog_visual_programming_abilityset::Audio::SetVolume (
    const uint8_t volume )
```

设置音量(0-100) 函数调用图:



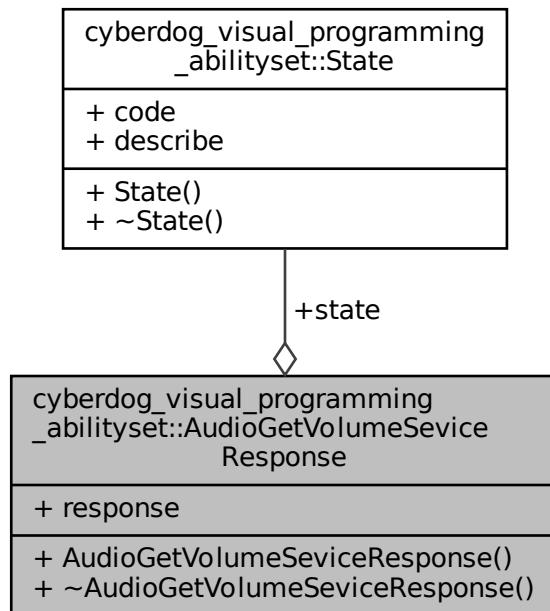
这是这个函数的调用关系图:



## 8.5 cyberdog\_visual\_programming\_abilityset::AudioGetVolumeSeviceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::AudioGetVolumeSeviceResponse 的协作图:



## Public 成员函数

- `AudioGetVolumeSeviceResponse ()`
- `~AudioGetVolumeSeviceResponse ()`

## Public 属性

- `State state`
- `SrvAudioGetVolume::Response response`

### 8.5.1 详细描述

audio 获取音量服务反馈

### 8.5.2 构造及析构函数说明

### 8.5.2.1 AudioGetVolumeSeviceResponse()

```
cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse::AudioGetVolumeSevice<--  
Response ( ) [inline]
```

### 8.5.2.2 ~AudioGetVolumeSeviceResponse()

```
cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse::~AudioGetVolumeSevice<--  
Response ( ) [inline]
```

## 8.5.3 类成员变量说明

### 8.5.3.1 response

```
SrvAudioGetVolume::Response cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse<--  
::response
```

反馈

### 8.5.3.2 state

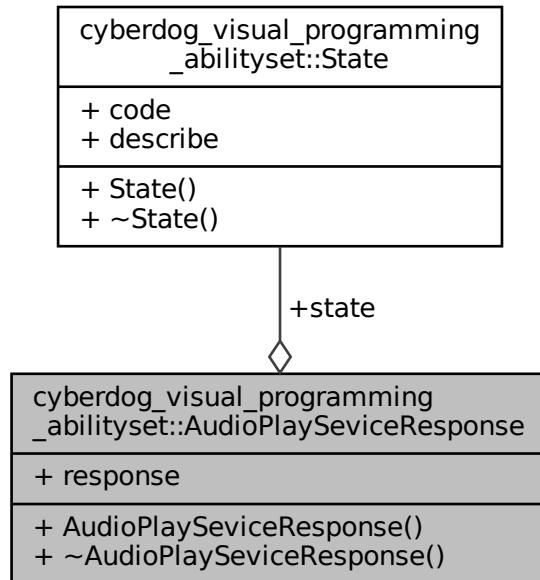
```
State cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse::state
```

状态

## 8.6 cyberdog\_visual\_programming\_abilityset::AudioPlaySevice<-- Response类 参 考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::AudioPlaySeviceResponse 的协作图:



## Public 成员函数

- `AudioPlaySeviceResponse ()`
- `~AudioPlaySeviceResponse ()`

## Public 属性

- `State state`
- `SrvAudioTextPlay::Response response`

### 8.6.1 详细描述

audio 播放服务反馈

### 8.6.2 构造及析构函数说明

#### 8.6.2.1 `AudioPlaySeviceResponse()`

```
cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse::AudioPlaySeviceResponse ( )
[inline]
```

### 8.6.2.2 ~AudioPlaySeviceResponse()

```
cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse::~AudioPlaySeviceResponse ()  
[inline]
```

## 8.6.3 类成员变量说明

### 8.6.3.1 response

```
SrvAudioTextPlay::Response cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse<-->  
::response
```

反馈

### 8.6.3.2 state

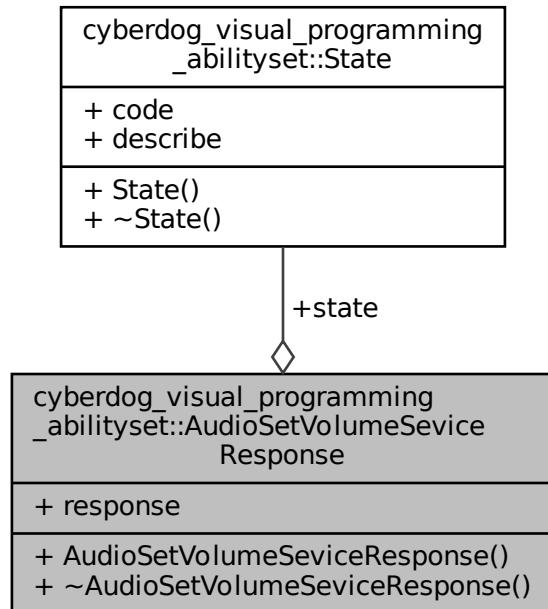
```
State cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse::state
```

状态

## 8.7 cyberdog\_visual\_programming\_abilityset::AudioSetVolumeSeviceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::AudioSetVolumeSeviceResponse 的协作图:



## Public 成员函数

- `AudioSetVolumeSeviceResponse ()`
- `~AudioSetVolumeSeviceResponse ()`

## Public 属性

- `State state`
- `SrvAudioSetVolume::Response response`

### 8.7.1 详细描述

audio 获取音量服务反馈

### 8.7.2 构造及析构函数说明

### 8.7.2.1 AudioSetVolumeSeviceResponse()

```
cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse::AudioSetVolumeSeviceResponse ( ) [inline]
```

### 8.7.2.2 ~AudioSetVolumeSeviceResponse()

```
cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse::~AudioSetVolumeSeviceResponse ( ) [inline]
```

## 8.7.3 类成员变量说明

### 8.7.3.1 response

```
SrvAudioSetVolume::Response cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse::response
```

反馈

### 8.7.3.2 state

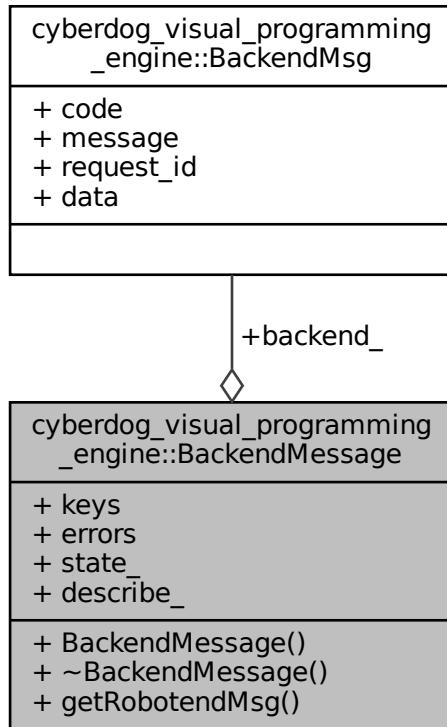
```
State cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse::state
```

状态

## 8.8 cyberdog\_visual\_programming\_engine::BackendMessage类 参考

```
#include <backend_message.hpp>
```

cyberdog\_visual\_programming\_engine::BackendMessage 的协作图:



## Public 类型

- using `CommonEnum` = enum { efficient=0, invalid, code, message, request\_id, data, }
- using `HttpCode` = enum { Success=200, AccessTimeout=400, UnauthorizedAccess=401, AccessDenied=403, ResourceDoesNotExist=404, NotSupported=405, ServerAbnormally=500, ParameterEmpty=10001, }

## Public 成员函数

- `BackendMessage (const std::string &msg)`
- `~BackendMessage ()`
- `void getRobotendMsg (GRPCMsg &)`

## Public 属性

- `const std::unordered_map<CommonEnum, std::string> keys`
- `const std::unordered_map<int, std::string> errors`
- `CommonEnum state_`
- `std::string describe_ {"")}`
- `BackendMsg backend_`

## 8.8.1 成员类型定义说明

### 8.8.1.1 CommonEnum

```
using cyberdog_visual_programming_engine::BackendMessage::CommonEnum = enum { efficient = 0, invalid, code, message, request_id, data, }
```

### 8.8.1.2 HttpStatusCode

```
using cyberdog_visual_programming_engine::BackendMessage::HttpCode = enum { Success = 200, AccessTimeout = 400, UnauthorizedAccess = 401, AccessDenied = 403, ResourceDoesNotExist = 404, NotSupported = 405, ServerAbnormally = 500, ParameterEmpty = 10001, }
```

## 8.8.2 构造及析构函数说明

### 8.8.2.1 BackendMessage()

```
cyberdog_visual_programming_engine::BackendMessage::BackendMessage ( const std::string & msg ) [explicit]
```

### 8.8.2.2 ~BackendMessage()

```
cyberdog_visual_programming_engine::BackendMessage::~BackendMessage ( ) [inline]
```

## 8.8.3 成员函数说明

### 8.8.3.1 getRobotendMsg()

```
void cyberdog_visual_programming_engine::BackendMessage::getRobotendMsg ( GRPCMsg & msg_ )
```

获取机器人端消息

## 8.8.4 类成员变量说明

### 8.8.4.1 backend\_

`BackendMsg` `cyberdog.visual_programming_engine::BackendMessage::backend_`

后端消息

### 8.8.4.2 describe\_

`std::string` `cyberdog.visual_programming_engine::BackendMessage::describe_ {"")}`

描述

### 8.8.4.3 errors

`const std::unordered_map<int, std::string>` `cyberdog.visual_programming_engine::BackendMessage::errors`

初始值:

```
= {
    {HttpCode::Success, "success"},
    {HttpCode::AccessTimeout, "Access Timeout"},
    {HttpCode::UnauthorizedAccess, "Access Denied"},
    {HttpCode::AccessDenied, "Unauthorized Access"},
    {HttpCode::ResourceDoesNotExist, "Resource does not exist"},
    {HttpCode::NotSupported, "The current request method is not supported"},
    {HttpCode::ServerAbnormally, "The server is running abnormally"},
    {HttpCode::ParameterEmpty, "The parameter cannot be empty"},
}
```

### 8.8.4.4 keys

`const std::unordered_map<CommonEnum, std::string>` `cyberdog.visual_programming_engine::BackendMessage::keys`

初始值:

```
= {
    {CommonEnum::code, "code"},
    {CommonEnum::message, "message"},
    {CommonEnum::request_id, "request_id"},
    {CommonEnum::data, "data"},

}
```

### 8.8.4.5 state\_

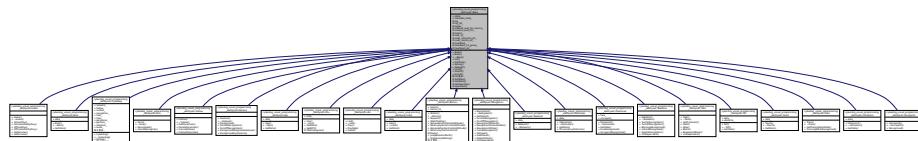
`CommonEnum` `cyberdog.visual_programming_engine::BackendMessage::state_`

状态

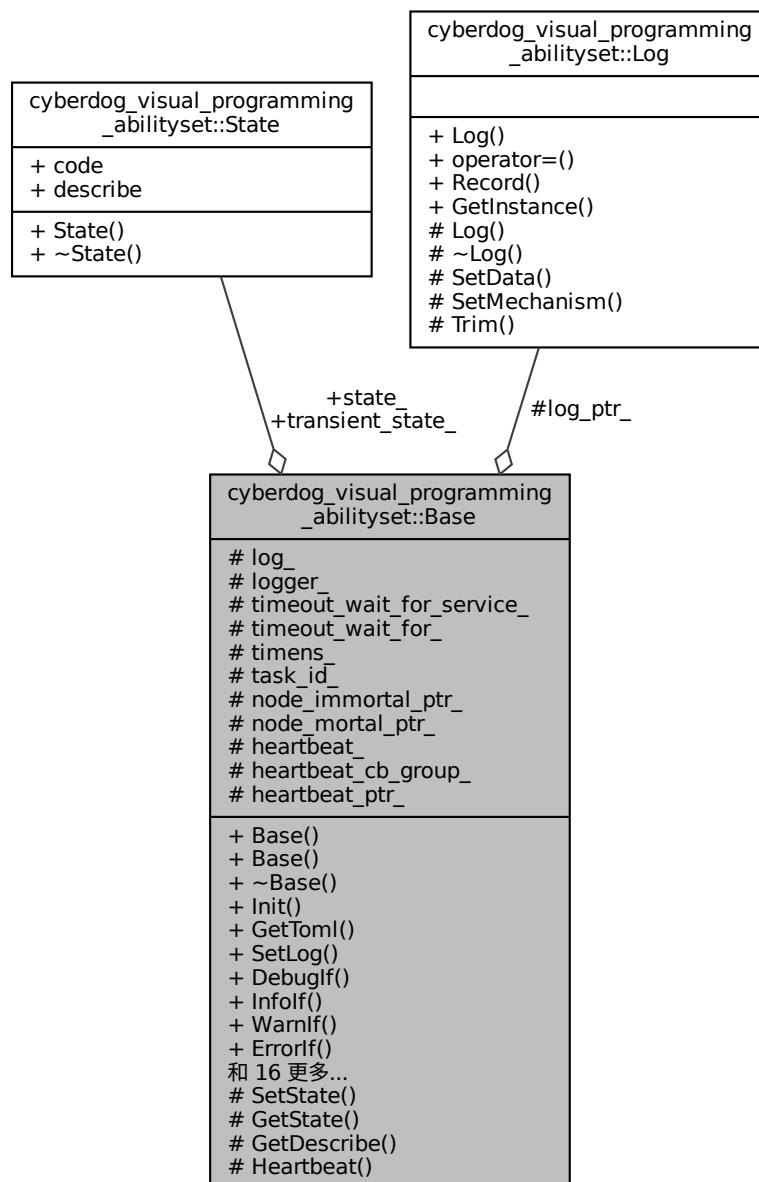
## 8.9 cyberdog\_visual\_programming\_abilityset::Base类 参考

```
#include <base.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Base 继承关系图:



cyberdog\_visual\_programming\_abilityset::Base 的协作图:



## Public 成员函数

- `Base (std::string _logger)`
- `Base (std::string _logger, bool _heartbeat)`
- `virtual ~Base ()`
- `bool Init (const std::string, const rclcpp::Node::SharedPtr, const rclcpp::Node::SharedPtr, const toml::value &)`
- `bool GetToml (const std::string &, toml::value &)`
- `virtual void SetLog (const bool)`
- template<typename ... Args>  
  `void DebugIf (bool condition, const char *format, const Args &... args)`
- template<typename ... Args>  
  `void InfoIf (bool condition, const char *format, const Args &... args)`
- template<typename ... Args>  
  `void WarnIf (bool condition, const char *format, const Args &... args)`
- template<typename ... Args>  
  `void ErrorIf (bool condition, const char *format, const Args &... args)`
- template<typename ... Args>  
  `void FatalIf (bool condition, const char *format, const Args &... args)`
- template<typename ... Args>  
  `void Debug (const char *format, const Args &... args)`
- template<typename ... Args>  
  `void Info (const char *format, const Args &... args)`
- template<typename ... Args>  
  `void Warn (const char *format, const Args &... args)`
- template<typename ... Args>  
  `void Error (const char *format, const Args &... args)`
- template<typename ... Args>  
  `void Fatal (const char *format, const Args &... args)`
- `void DebugIf (bool condition, const char *message)`
- `void InfoIf (bool condition, const char *message)`
- `void WarnIf (bool condition, const char *message)`
- `void ErrorIf (bool condition, const char *message)`
- `void FatalIf (bool condition, const char *message)`
- `void Debug (const char *message)`
- `void Info (const char *message)`
- `void Warn (const char *message)`
- `void Error (const char *message)`
- `void Fatal (const char *message)`

## Public 属性

- `State state_`
- `State transient_state_`

## Protected 成员函数

- `void SetState (const StateCode)`
- `State GetState (const std::string, const StateCode)`
- `std::string GetDescribe (const std::string, const StateCode)`
- `void Heartbeat ()`

## Protected 属性

- bool `log_` {true}
- `Log * log_ptr_` {nullptr}
- std::string `logger_` {"")}
- uint16\_t `timeout_wait_for_service_` {60}
- uint16\_t `timeout_wait_for_` {10}
- uint64\_t `timens_` {0}
- std::string `task_id_` {"")}
- rclcpp::Node::SharedPtr `node_immortal_ptr_` {nullptr}
- rclcpp::Node::SharedPtr `node_mortal_ptr_` {nullptr}
- bool `heartbeat_` {false}
- rclcpp::CallbackGroup::SharedPtr `heartbeat_cb_group_` {nullptr}
- rclcpp::TimerBase::SharedPtr `heartbeat_ptr_` {nullptr}

### 8.9.1 构造及析构函数说明

#### 8.9.1.1 Base() [1/2]

```
cyberdog_visual_programming_abilityset::Base::Base (
    std::string _logger )  [inline], [explicit]
```

#### 8.9.1.2 Base() [2/2]

```
cyberdog_visual_programming_abilityset::Base::Base (
    std::string _logger,
    bool _heartbeat )  [inline]
```

#### 8.9.1.3 ~Base()

```
virtual cyberdog_visual_programming_abilityset::Base::~Base ( )  [inline], [virtual]
```

### 8.9.2 成员函数说明

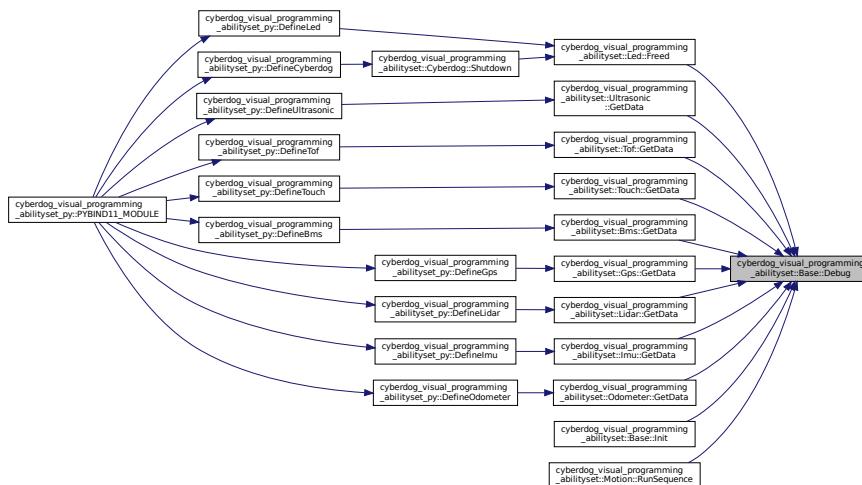
### 8.9.2.1 Debug() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::Debug (
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



这是这个函数的调用关系图:



### 8.9.2.2 Debug() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Debug (
    const char * message ) [inline]
```

函数调用图:



### 8.9.2.3 DebugIf() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::DebugIf (
    bool condition,
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



### 8.9.2.4 DebugIf() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::DebugIf (
    bool condition,
    const char * message ) [inline]
```

函数调用图:



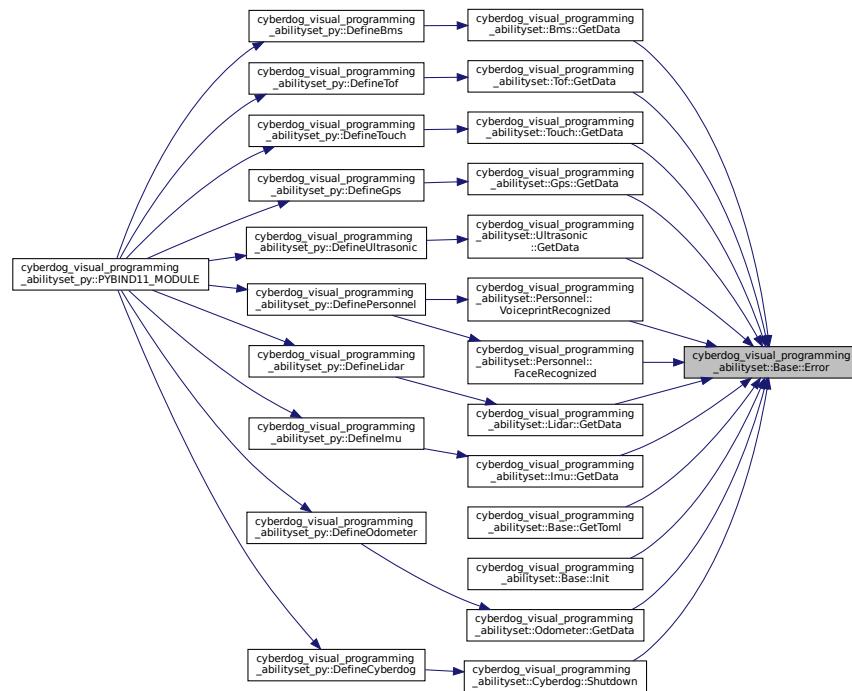
### 8.9.2.5 Error() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::Error (
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



这是这个函数的调用关系图:



### 8.9.2.6 Error() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Error (
    const char * message ) [inline]
```

函数调用图:



### 8.9.2.7 ErrorIf() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::ErrorIf (
    bool condition,
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



### 8.9.2.8 ErrorIf() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::ErrorIf (
    bool condition,
    const char * message ) [inline]
```

函数调用图:



### 8.9.2.9 Fatal() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::Fatal (
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



### 8.9.2.10 Fatal() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Fatal (
    const char * message ) [inline]
```

函数调用图:



### 8.9.2.11 FatalIf() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::FatalIf (
    bool condition,
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



### 8.9.2.12 FatalIf() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::FatalIf (
    bool condition,
    const char * message ) [inline]
```

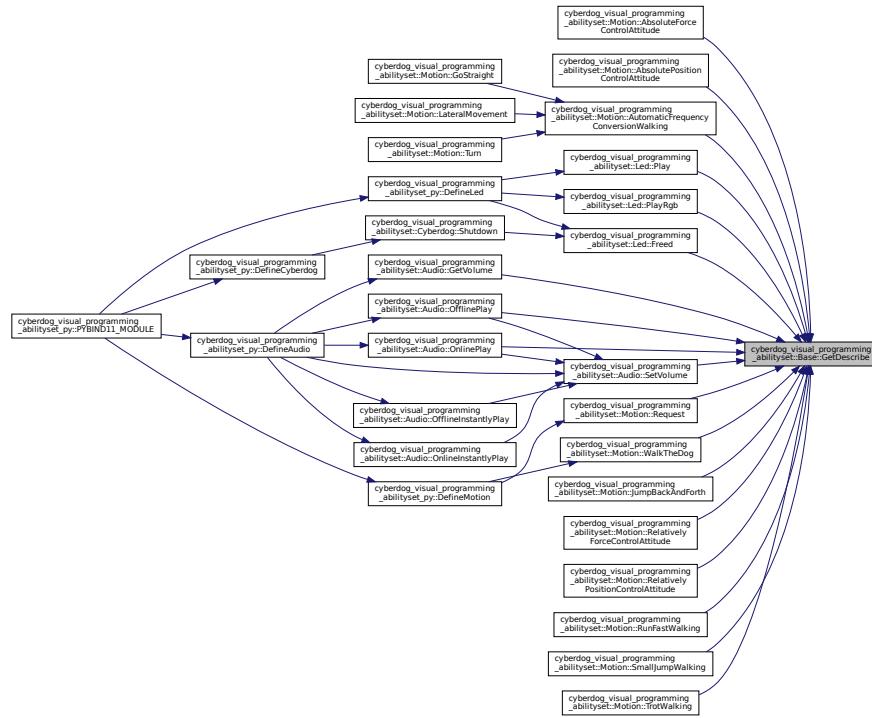
函数调用图:



### 8.9.2.13 GetDescribe()

```
std::string cyberdog_visual_programming_abilityset::Base::GetDescribe (
    const std::string _fun,
    const StatusCode _code ) [protected]
```

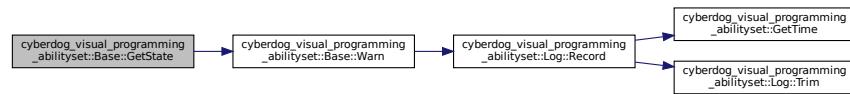
获取状态描述 这是这个函数的调用关系图:



### 8.9.2.14 GetState()

```
State cyberdog_visual_programming_abilityset::Base::GetState (
    const std::string _fun,
    const StatusCode _code ) [protected]
```

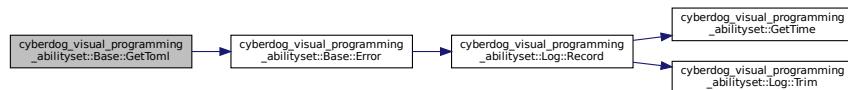
获取状态 函数调用图:



### 8.9.2.15 GetToml()

```
bool cyberdog_visual_programming_abilityset::Base::GetToml (
    const std::string & _toml_file,
    toml::value & _toml )
```

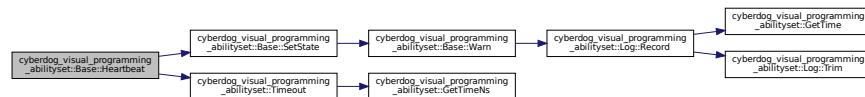
获取 toml 函数调用图:



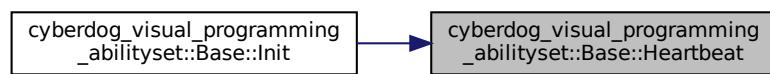
### 8.9.2.16 Heartbeat()

```
void cyberdog_visual_programming_abilityset::Base::Heartbeat ( ) [protected]
```

心跳 函数调用图:



这是这个函数的调用关系图:



## 8.9.2.17 Info() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::Info (
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



## 8.9.2.18 Info() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Info (
    const char * message ) [inline]
```

函数调用图:



## 8.9.2.19 InfoIf() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::InfoIf (
    bool condition,
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



### 8.9.2.20 Infof() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Infof (
    bool condition,
    const char * message ) [inline]
```

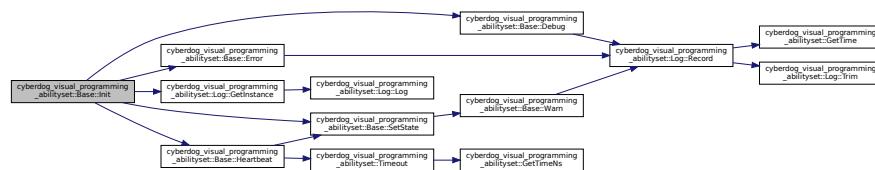
函数调用图:



### 8.9.2.21 Init()

```
bool cyberdog_visual_programming_abilityset::Base::Init (
    const std::string &task_id,
    const rclcpp::Node::SharedPtr &node_immortal_ptr_,
    const rclcpp::Node::SharedPtr &node_mortal_ptr_,
    const toml::value & params_toml )
```

初始化 函数调用图:



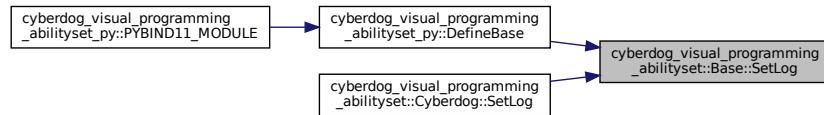
### 8.9.2.22 SetLog()

```
void cyberdog_visual_programming_abilityset::Base::SetLog (
    const bool log ) [virtual]
```

设置日志

被 [cyberdog\\_visual\\_programming\\_abilityset::Cyberdog](#) 重载.

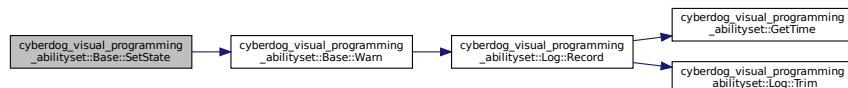
这是这个函数的调用关系图:



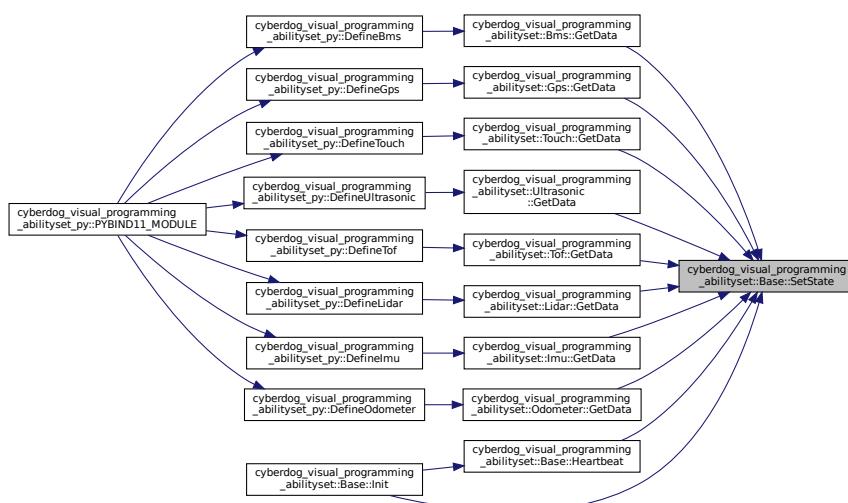
### 8.9.2.23 SetState()

```
void cyberdog_visual_programming_abilityset::Base::SetState (
    const StateCode _code ) [protected]
```

设置状态 函数调用图:



这是这个函数的调用关系图:



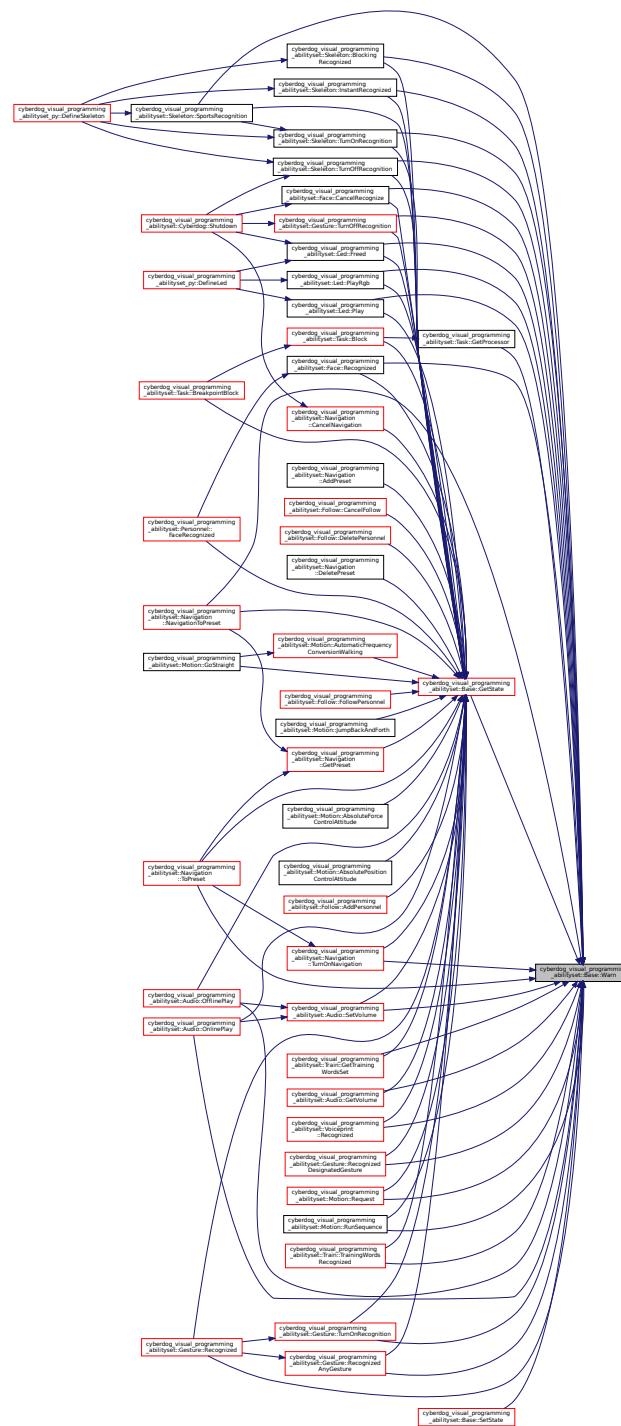
### 8.9.2.24 Warn() [1/2]

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::Warn (
    const char * format,
    const Args &... args ) [inline]
```

函数调用图:



这是这个函数的调用关系图:



### 8.9.2.25 Warn() [2/2]

```
void cyberdog_visual_programming_abilityset::Base::Warn (
```

函数调用图:



### 8.9.2.26 WarnIf() [1/2]

```

template<typename ... Args>
void cyberdog_visual_programming_abilityset::Base::WarnIf (
    bool condition,
    const char * format,
    const Args &... args ) [inline]
  
```

函数调用图:



### 8.9.2.27 WarnIf() [2/2]

```

void cyberdog_visual_programming_abilityset::Base::WarnIf (
    bool condition,
    const char * message ) [inline]
  
```

函数调用图:



### 8.9.3 类成员变量说明

#### 8.9.3.1 heartbeat\_

```
bool cyberdog_visual_programming_abilityset::Base::heartbeat_ {false} [protected]
```

是否开启心跳

#### 8.9.3.2 heartbeat\_cb\_group\_

```
rclcpp::CallbackGroup::SharedPtr cyberdog_visual_programming_abilityset::Base::heartbeat_cb_group_ {nullptr} [protected]
```

[回调组]心跳

#### 8.9.3.3 heartbeat\_ptr\_

```
rclcpp::TimerBase::SharedPtr cyberdog_visual_programming_abilityset::Base::heartbeat_ptr_ {nullptr} [protected]
```

[定时器]心跳

#### 8.9.3.4 log\_

```
bool cyberdog_visual_programming_abilityset::Base::log_ {true} [protected]
```

日志开关

#### 8.9.3.5 log\_ptr\_

```
Log* cyberdog_visual_programming_abilityset::Base::log_ptr_ {nullptr} [protected]
```

日志

#### 8.9.3.6 logger\_

```
std::string cyberdog_visual_programming_abilityset::Base::logger_ {"") [protected]
```

日志名称

#### 8.9.3.7 node\_immortal\_ptr\_

```
rclcpp::Node::SharedPtr cyberdog_visual_programming_abilityset::Base::node_immortal_ptr_ {nullptr} [protected]
```

神仙节点

### 8.9.3.8 node\_mortal\_ptr\_

```
rclcpp::Node::SharedPtr cyberdog_visual_programming_abilityset::Base::node_mortal_ptr_ {nullptr}  
[protected]
```

凡人节点

### 8.9.3.9 state\_

```
State cyberdog_visual_programming_abilityset::Base::state_
```

整体状态

### 8.9.3.10 task\_id\_

```
std::string cyberdog_visual_programming_abilityset::Base::task_id_ {"") [protected]
```

任务id

### 8.9.3.11 timens\_

```
uint64_t cyberdog_visual_programming_abilityset::Base::timens_ {0} [protected]
```

数据时间

### 8.9.3.12 timeout\_wait\_for\_

```
uint16_t cyberdog_visual_programming_abilityset::Base::timeout_wait_for_ {10} [protected]
```

等待服务响应超时

### 8.9.3.13 timeout\_wait\_for\_service\_

```
uint16_t cyberdog_visual_programming_abilityset::Base::timeout_wait_for_service_ {60} [protected]
```

等待服务启动超时

### 8.9.3.14 transient\_state\_

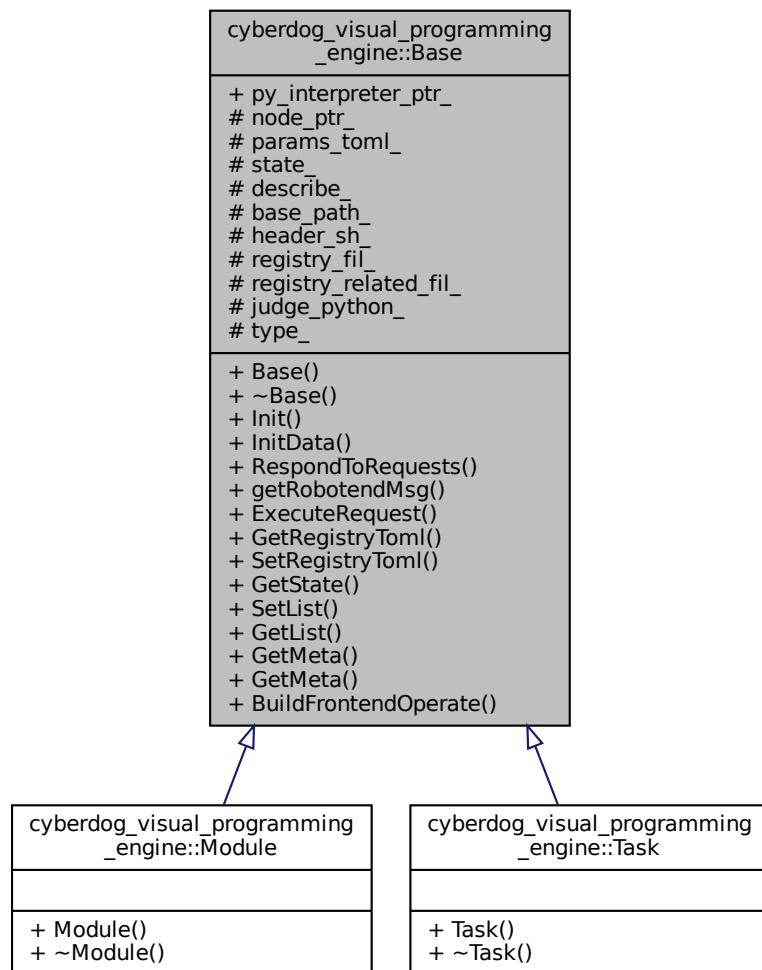
```
State cyberdog_visual_programming_abilityset::Base::transient_state_
```

瞬时状态

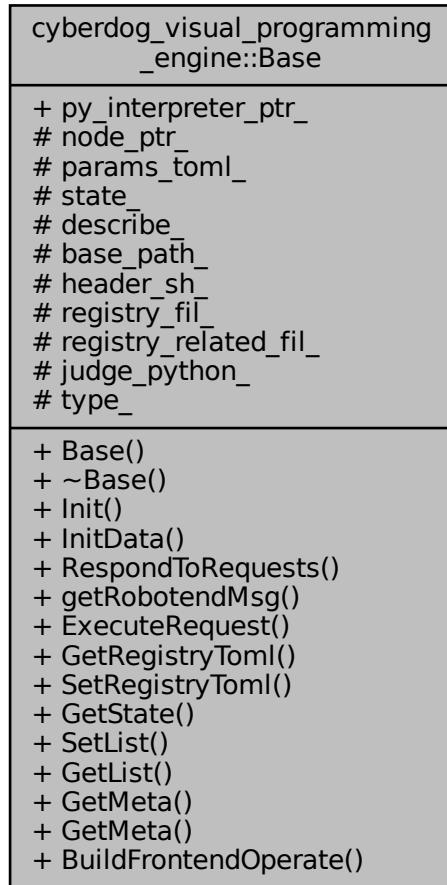
## 8.10 cyberdog\_visual\_programming\_engine::Base类 参考

```
#include <base.hpp>
```

类 cyberdog\_visual\_programming\_engine::Base 继承关系图:



cyberdog\_visual\_programming\_engine::Base 的协作图:



## Public 成员函数

- `Base (std::string _type)`
- `~Base ()`
- `bool Init (const rclcpp::Node::SharedPtr &, const std::shared_ptr< PythonInterpreter > &, const toml::value &)`
- `virtual bool InitData ()`
- `bool RespondToRequests (const OperateMsg &, GRPCMsg &)`
- `void getRobotendMsg (const OperateMsg &, GRPCMsg &)`
- `virtual bool ExecuteRequest (const OperateMsg &, GRPCMsg &)`
- `bool GetRegistryToml (toml::value &, bool _is_registry_file=true)`
- `bool SetRegistryToml (const toml::value &, bool _is_registry_file=true)`
- `bool GetState (const std::string, std::string &)`
- `bool SetList (const OperateMsg &, const std::string &, const std::string &, std::vector< std::string > &)`
- `bool GetList (const OperateMsg &, GRPCMsg &)`
- `bool GetMeta (const toml::value &, const std::string &, OperateMsg &)`
- `bool GetMeta (const std::string &, OperateMsg &)`
- `bool BuildFrontendOperate (const std::string, std::string &)`

**Public 属性**

- std::shared\_ptr< PythonInterpreter > py\_interpreter\_ptr\_ {nullptr}

**Protected 属性**

- rclcpp::Node::SharedPtr node\_ptr\_ {nullptr}
- toml::value params\_toml\_
- StateEnum state\_
- std::string describe\_ {"")}
- std::string base\_path\_ {"")}
- std::vector< std::string > header\_sh\_
- std::string registry\_fil\_ {"")}
- std::string registry\_related\_fil\_ {"")}
- const std::string judge\_python\_ {"pyflakes "}
- std::string type\_ {"")}

**8.10.1 构造及析构函数说明****8.10.1.1 Base()**

```
cyberdog_visual_programming_engine::Base::Base (
    std::string _type ) [explicit]
```

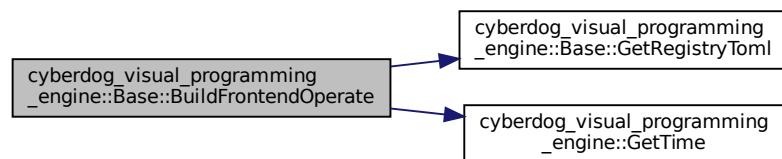
**8.10.1.2 ~Base()**

```
cyberdog_visual_programming_engine::Base::~Base ()
```

**8.10.2 成员函数说明****8.10.2.1 BuildFrontendOperate()**

```
bool cyberdog_visual_programming_engine::Base::BuildFrontendOperate (
    const std::string _operate,
    std::string & _frontend_msg )
```

构建前端操作 函数调用图:



### 8.10.2.2 ExecuteRequest()

```
bool cyberdog_visual_programming_engine::Base::ExecuteRequest (
    const OperateMsg & _msg,
    GRPCMsg & ) [virtual]
```

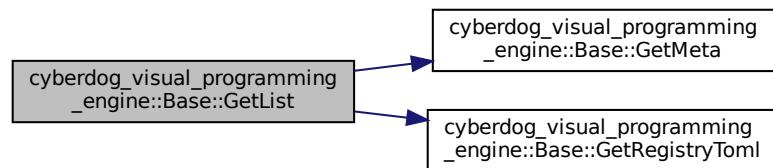
执行请求 这是这个函数的调用关系图:



### 8.10.2.3 GetList()

```
bool cyberdog_visual_programming_engine::Base::GetList (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

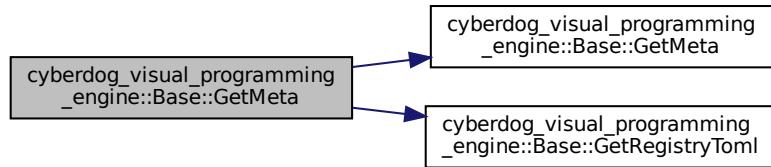
获取列表 函数调用图:



### 8.10.2.4 GetMeta() [1/2]

```
bool cyberdog_visual_programming_engine::Base::GetMeta (
    const std::string & _id,
    OperateMsg & _meta )
```

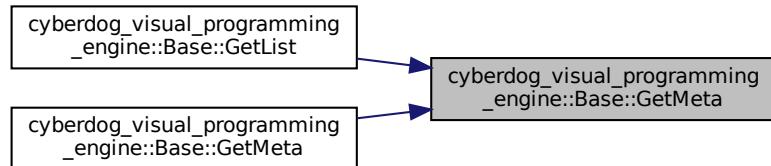
获取任务 函数调用图:



#### 8.10.2.5 GetMeta() [2/2]

```
bool cyberdog_visual_programming_engine::Base::GetMeta (
    const toml::value & _toml,
    const std::string & _id,
    OperateMsg & _meta )
```

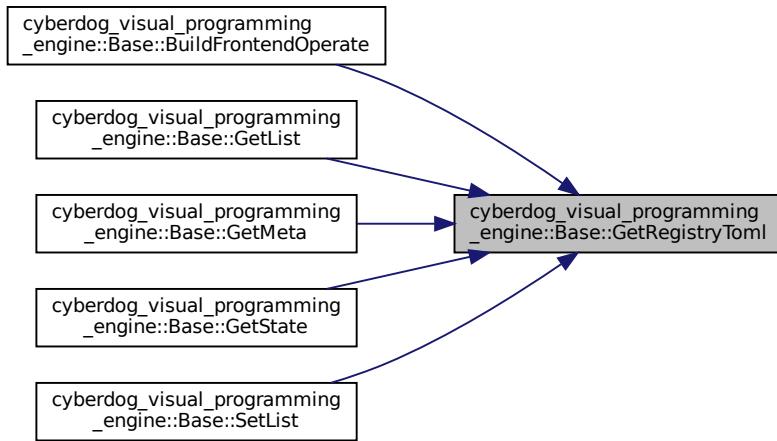
获取任务 这是这个函数的调用关系图:



#### 8.10.2.6 GetRegistryToml()

```
bool cyberdog_visual_programming_engine::Base::GetRegistryToml (
    toml::value & _registry_toml,
    bool _is_registry_file = true )
```

获取注册表 这是这个函数的调用关系图:



#### 8.10.2.7 getRobotendMsg()

```
void cyberdog_visual_programming_engine::Base::getRobotendMsg (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

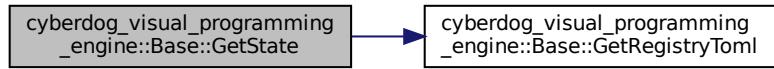
获取后端消息 这是这个函数的调用关系图:



#### 8.10.2.8 GetState()

```
bool cyberdog_visual_programming_engine::Base::GetState (
    const std::string id,
    std::string & now_state )
```

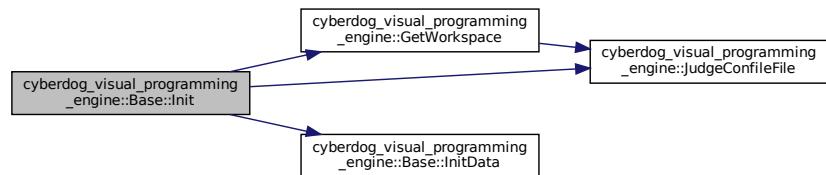
获取状态 函数调用图:



### 8.10.2.9 Init()

```
bool cyberdog_visual_programming_engine::Base::Init (
    const rclcpp::Node::SharedPtr & _node_ptr,
    const std::shared_ptr< PythonInterpreter > & _py_interpreter_ptr,
    const toml::value & _params_toml )
```

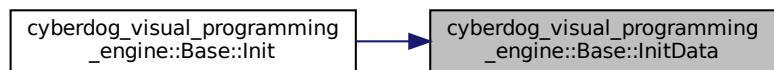
初始化 函数调用图:



### 8.10.2.10 InitData()

```
bool cyberdog_visual_programming_engine::Base::InitData ( ) [virtual]
```

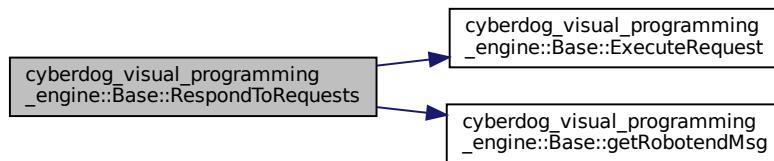
初始化数据 这是这个函数的调用关系图:



### 8.10.2.11 RespondToRequests()

```
bool cyberdog_visual_programming_engine::Base::RespondToRequests (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

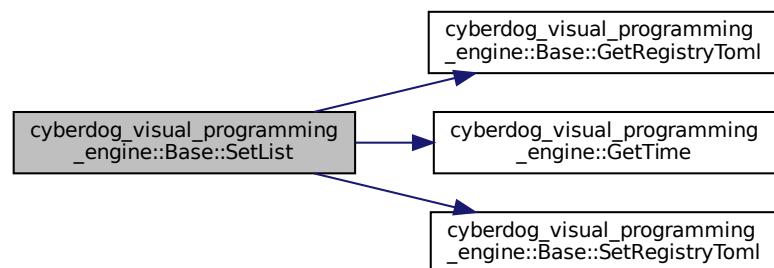
响应请求 函数调用图:



### 8.10.2.12 SetList()

```
bool cyberdog_visual_programming_engine::Base::SetList (
    const OperateMsg & _msg,
    const std::string & _state,
    const std::string & _file,
    std::vector< std::string > & _dependent )
```

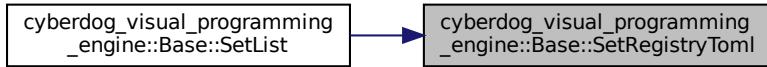
设置列表 函数调用图:



### 8.10.2.13 SetRegistryToml()

```
bool cyberdog_visual_programming_engine::Base::SetRegistryToml (
    const toml::value & _registry_toml,
    bool _is_registry_file = true )
```

设置注册表 这是这个函数的调用关系图:



## 8.10.3 类成员变量说明

### 8.10.3.1 base\_path\_

```
std::string cyberdog_visual_programming_engine::Base::base_path_ {" "} [protected]
```

任务路径

### 8.10.3.2 describe\_

```
std::string cyberdog_visual_programming_engine::Base::describe_ {" "} [protected]
```

描述

### 8.10.3.3 header\_sh\_

```
std::vector<std::string> cyberdog_visual_programming_engine::Base::header_sh_ [protected]
```

shell 文件头

### 8.10.3.4 judge\_python\_

```
const std::string cyberdog_visual_programming_engine::Base::judge_python_ {"pyflakes "} [protected]
```

判断 python

### 8.10.3.5 node\_ptr\_

```
rclcpp::Node::SharedPtr cyberdog_visual_programming_engine::Base::node_ptr_ {nullptr} [protected]
```

节点

### 8.10.3.6 params\_toml\_

```
toml::value cyberdog_visual_programming_engine::Base::params_toml_ [protected]
```

配置文件数据

### 8.10.3.7 py\_interpreter\_ptr\_

```
std::shared_ptr<PythonInterpreter> cyberdog_visual_programming_engine::Base::py_interpreter_ptr_ {nullptr}
```

python 解释器

### 8.10.3.8 registry\_fil\_

```
std::string cyberdog_visual_programming_engine::Base::registry_fil_ {""} [protected]
```

注册表文件

### 8.10.3.9 registry\_related\_fil\_

```
std::string cyberdog_visual_programming_engine::Base::registry_related_fil_ {""} [protected]
```

注册表相关文件

### 8.10.3.10 state\_

```
StateEnum cyberdog_visual_programming_engine::Base::state_ [protected]
```

状态

### 8.10.3.11 type\_

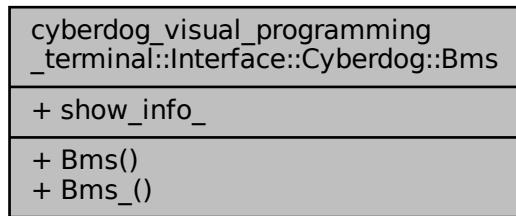
```
std::string cyberdog_visual_programming_engine::Base::type_ {""} [protected]
```

类型

## 8.11 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Bms结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Bms 的协作图:



### Public 成员函数

- `Bms()`
- void `Bms_(const std::string _fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

#### 8.11.1 构造及析构函数说明

##### 8.11.1.1 Bms()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms::Bms ( ) [inline]
```

#### 8.11.2 成员函数说明

##### 8.11.2.1 Bms\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms::Bms_ (
    const std::string _fun) [inline]
```

### 8.11.3 类成员变量说明

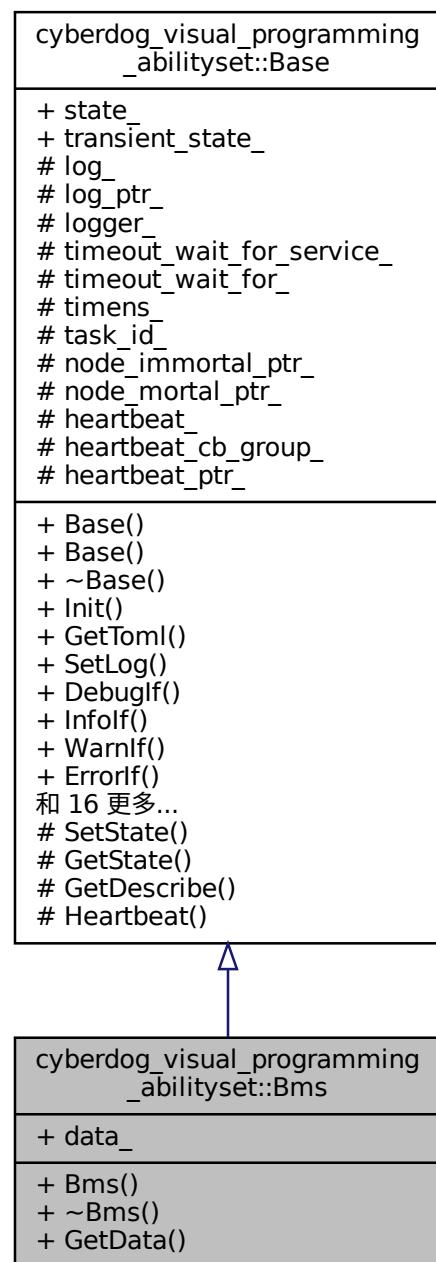
#### 8.11.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Bms::show_info_
```

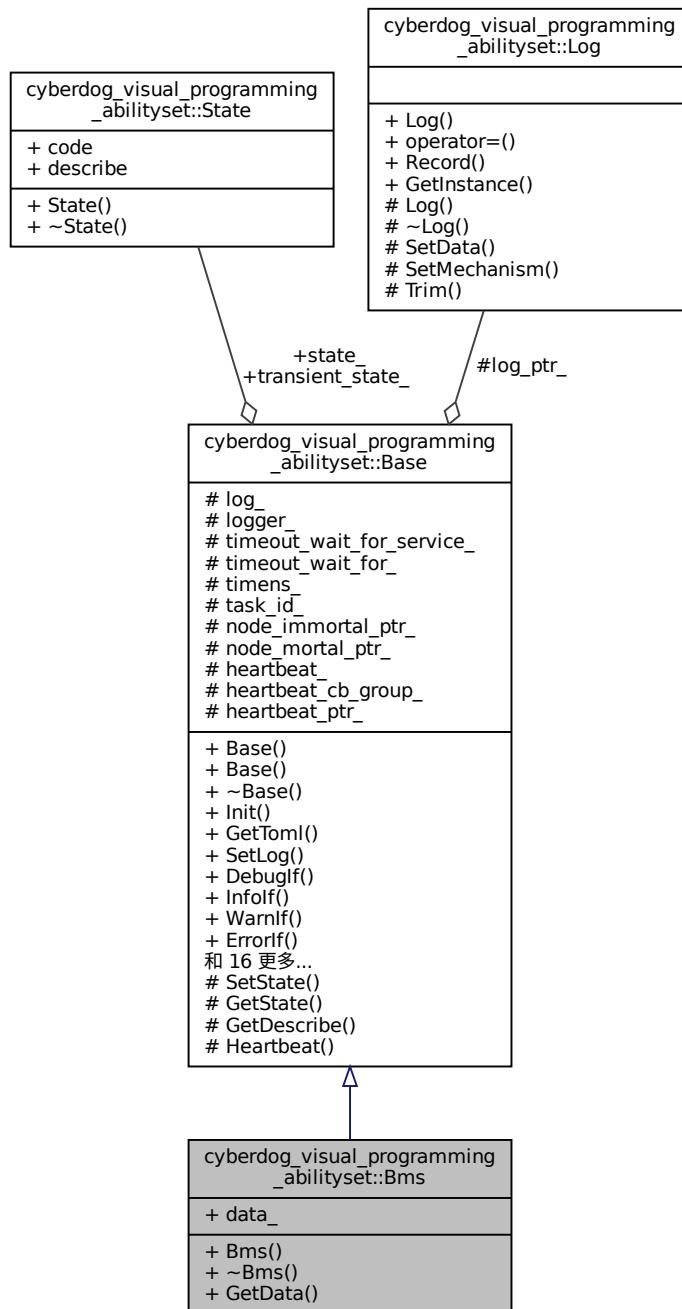
## 8.12 cyberdog\_visual\_programming\_abilityset::Bms类 参考

```
#include <bms.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Bms 继承关系图:



cyberdog\_visual\_programming\_abilityset::Bms 的协作图:



## Public 成员函数

- [Bms \(\)](#)
- [~Bms \(\)](#)
- [MsgBmsStatus GetData \(const int \\_timeout=5\)](#)

## Public 属性

- `MsgBmsStatus data_`

额外继承的成员函数

### 8.12.1 构造及析构函数说明

#### 8.12.1.1 Bms()

```
cyberdog_visual_programming_abilityset::Bms::Bms ( ) [inline]
```

#### 8.12.1.2 ~Bms()

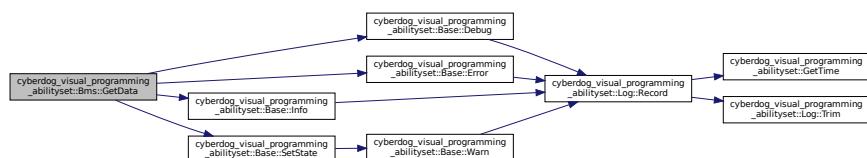
```
cyberdog_visual_programming_abilityset::Bms::~Bms ( ) [inline]
```

### 8.12.2 成员函数说明

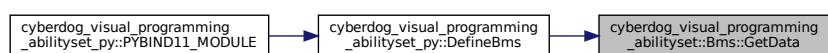
#### 8.12.2.1 GetData()

```
MsgBmsStatus cyberdog_visual_programming_abilityset::Bms::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.12.3 类成员变量说明

#### 8.12.3.1 `data_`

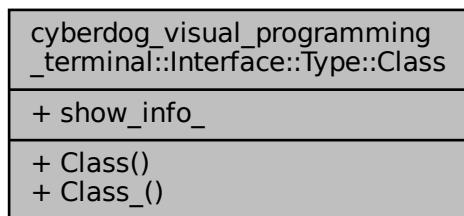
`MsgBmsStatus cyberdog_visual_programming_abilityset::Bms::data_`

数据

## 8.13 `cyberdog_visual_programming_terminal::Interface::Type::Class` 结构体 参考

#include <interface.hpp>

`cyberdog_visual_programming_terminal::Interface::Type::Class` 的协作图:



### Public 成员函数

- `Class ()`
- `void Class_ (const std::string &fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

### 8.13.1 构造及析构函数说明

### 8.13.1.1 Class()

```
cyberdog_visual_programming_terminal::Interface::Type::Class::Class ( ) [inline]
```

## 8.13.2 成员函数说明

### 8.13.2.1 Class\_()

```
void cyberdog_visual_programming_terminal::Interface::Type::Class::Class_ (
    const std::string &fun ) [inline]
```

## 8.13.3 类成员变量说明

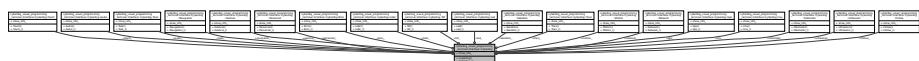
### 8.13.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Type::Class::show_info_
```

## 8.14 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog 的协作图:



## 类

- struct [Audio](#)
- struct [Bms](#)
- struct [Follow](#)
- struct [Gesture](#)
- struct [Gps](#)
- struct [Imu](#)
- struct [Led](#)
- struct [Lidar](#)
- struct [Motion](#)
- struct [Navigation](#)
- struct [Network](#)
- struct [Odometer](#)
- struct [Personnel](#)
- struct [Skeleton](#)
- struct [Task](#)
- struct [Tof](#)
- struct [Touch](#)
- struct [Train](#)
- struct [Ultrasonic](#)

## Public 成员函数

- `Cyberdog ()`
- `void Cyberdog_(std::string _fun)`

## Public 属性

- `std::function< void(std::string)> show_info_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Network network_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow follow_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion motion_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation navigation_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Task task_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Train train_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel personnel_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture gesture_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton skeleton_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms bms_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Led led_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio audio_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch touch_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps gps_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof tof_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar lidar_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic ultrasonic_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer odometer_`
- `struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu imu_`

### 8.14.1 构造及析构函数说明

#### 8.14.1.1 Cyberdog()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Cyberdog_() [inline]
```

### 8.14.2 成员函数说明

#### 8.14.2.1 Cyberdog\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Cyberdog_(
    std::string _fun) [inline]
```

### 8.14.3 类成员变量说明

#### 8.14.3.1 audio\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio cyberdog_visual_programming←
-terminal::Interface::Cyberdog::audio_
```

#### 8.14.3.2 bms\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms cyberdog_visual_programming←
-terminal::Interface::Cyberdog::bms_
```

#### 8.14.3.3 follow\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow cyberdog_visual←
-programming_terminal::Interface::Cyberdog::follow_
```

#### 8.14.3.4 gesture\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture cyberdog_visual←
-programming_terminal::Interface::Cyberdog::gesture_
```

#### 8.14.3.5 gps\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps cyberdog_visual_programming←
-terminal::Interface::Cyberdog::gps_
```

#### 8.14.3.6 imu\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu cyberdog_visual_programming←
-terminal::Interface::Cyberdog::imu_
```

### 8.14.3.7 led\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Led cyberdog_visual_programming←
-terminal::Interface::Cyberdog::led.
```

### 8.14.3.8 lidar\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar cyberdog_visual_programming←
-terminal::Interface::Cyberdog::lidar.
```

### 8.14.3.9 motion\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion cyberdog_visual←
-programming_terminal::Interface::Cyberdog::motion.
```

### 8.14.3.10 navigation\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation cyberdog_visual←
-programming_terminal::Interface::Cyberdog::navigation.
```

### 8.14.3.11 network\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Network cyberdog_visual←
-programming_terminal::Interface::Cyberdog::network.
```

### 8.14.3.12 odometer\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer cyberdog_visual←
-programming_terminal::Interface::Cyberdog::odometer.
```

### 8.14.3.13 personnel\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel cyberdog_visual←
-programming_terminal::Interface::Cyberdog::personnel.
```

#### 8.14.3.14 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_
```

#### 8.14.3.15 skeleton\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton cyberdog_visual_programming_terminal::Interface::Cyberdog::skeleton_
```

#### 8.14.3.16 task\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Task cyberdog_visual_programming_terminal::Interface::Cyberdog::task_
```

#### 8.14.3.17 tof\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof cyberdog_visual_programming_terminal::Interface::Cyberdog::tوف_
```

#### 8.14.3.18 touch\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch cyberdog_visual_programming_terminal::Interface::Cyberdog::touch_
```

#### 8.14.3.19 train\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Train cyberdog_visual_programming_terminal::Interface::Cyberdog::train_
```

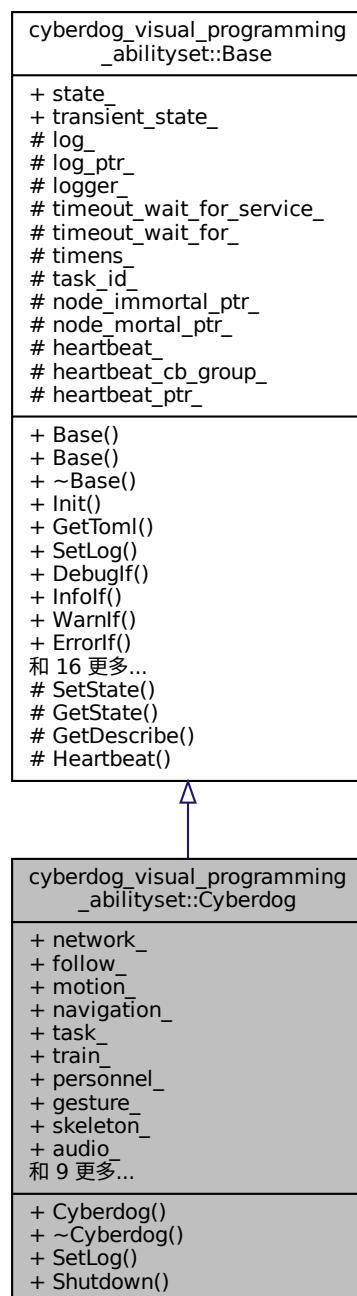
#### 8.14.3.20 ultrasonic\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic cyberdog_visual_programming_terminal::Interface::Cyberdog::ultrasonic_
```

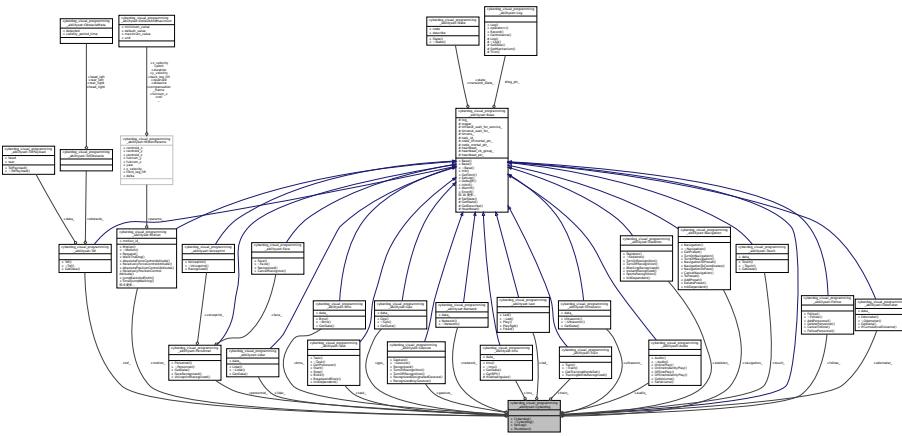
## 8.15 cyberdog\_visual\_programming\_abilityset::Cyberdog类 参考

```
#include <cyberdog.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Cyberdog 继承关系图:



cyberdog\_visual\_programming\_abilityset::Cyberdog 的协作图:



## Public 成员函数

- [Cyberdog](#) (std::string \_task="task", std::string \_namespace="namespace", bool \_ros=false, std::string \_parameters="")
- [~Cyberdog](#) ()
- void [SetLog](#) (const bool)
- void [Shutdown](#) (bool \_exit=true)

## Public 属性

- [Network](#) network\_
- [Follow](#) follow\_
- [Motion](#) motion\_
- [Navigation](#) navigation\_
- [Task](#) task\_
- [Train](#) train\_
- [Personnel](#) personnel\_
- [Gesture](#) gesture\_
- [Skeleton](#) skeleton\_
- [Audio](#) audio\_
- [Bms](#) bms\_
- [Led](#) led\_
- [Touch](#) touch\_
- [Imu](#) imu\_
- [Odometer](#) odometer\_
- [Gps](#) gps\_
- [Lidar](#) lidar\_
- [Tof](#) tof\_
- [Ultrasonic](#) ultrasonic\_

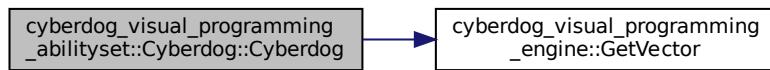
## 额外继承的成员函数

### 8.15.1 构造及析构函数说明

### 8.15.1.1 Cyberdog()

```
cyberdog_visual_programming_abilityset::Cyberdog::Cyberdog (
    std::string _task = "task",
    std::string _namespace = "namespace",
    bool _ros = false,
    std::string _parameters = "" )
```

函数调用图:



### 8.15.1.2 ~Cyberdog()

```
cyberdog_visual_programming_abilityset::Cyberdog::~Cyberdog () [inline]
```

## 8.15.2 成员函数说明

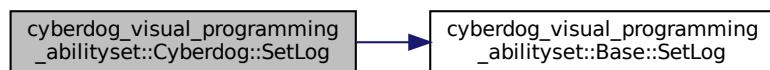
### 8.15.2.1 SetLog()

```
void cyberdog_visual_programming_abilityset::Cyberdog::SetLog (
    const bool _log ) [virtual]
```

设置日志

重载 [cyberdog\\_visual\\_programming\\_abilityset::Base](#) .

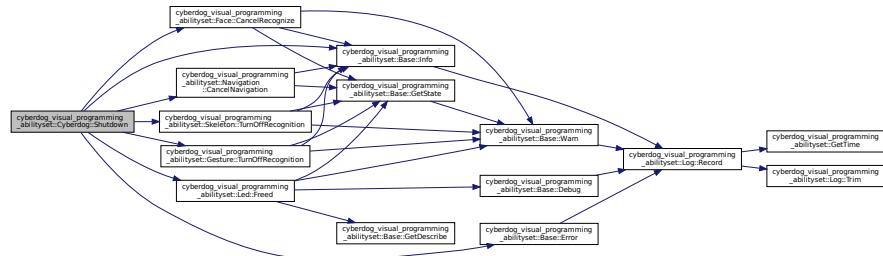
函数调用图:



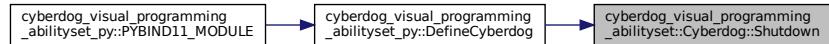
### 8.15.2.2 Shutdown()

```
void cyberdog_visual_programming_abilityset::Cyberdog::Shutdown (
    bool _exit = true )
```

终止函数调用图:



这是这个函数的调用关系图:



### 8.15.3 类成员变量说明

#### 8.15.3.1 audio\_

`Audio` `cyberdog_visual_programming_abilityset::Cyberdog::audio_`

语音模块句柄

#### 8.15.3.2 bms\_

`Bms` `cyberdog_visual_programming_abilityset::Cyberdog::bms_`

Bms模块句柄

#### 8.15.3.3 follow\_

`Follow` `cyberdog_visual_programming_abilityset::Cyberdog::follow_`

跟随模块句柄

#### 8.15.3.4 gesture\_

`Gesture` cyberdog\_visual\_programming\_abilityset::Cyberdog::gesture\_

手势识别模块句柄

#### 8.15.3.5 gps\_

`Gps` cyberdog\_visual\_programming\_abilityset::Cyberdog::gps\_

Gps模块句柄

#### 8.15.3.6 imu\_

`Imu` cyberdog\_visual\_programming\_abilityset::Cyberdog::imu\_

Imu模块句柄

#### 8.15.3.7 led\_

`Led` cyberdog\_visual\_programming\_abilityset::Cyberdog::led\_

Led模块句柄

#### 8.15.3.8 lidar\_

`Lidar` cyberdog\_visual\_programming\_abilityset::Cyberdog::lidar\_

雷达模块句柄

#### 8.15.3.9 motion\_

`Motion` cyberdog\_visual\_programming\_abilityset::Cyberdog::motion\_

运动模块句柄

#### 8.15.3.10 navigation\_

`Navigation` cyberdog\_visual\_programming\_abilityset::Cyberdog::navigation\_

导航模块句柄

#### 8.15.3.11 network\_

`Network` cyberdog\_visual\_programming\_abilityset::Cyberdog::network\_

网络模块句柄

### 8.15.3.12 odometer\_

`Odometer` cyberdog\_visual\_programming\_abilityset::Cyberdog::odometer\_

里程计模块句柄

### 8.15.3.13 personnel\_

`Personnel` cyberdog\_visual\_programming\_abilityset::Cyberdog::personnel\_

人员模块句柄

### 8.15.3.14 skeleton\_

`Skeleton` cyberdog\_visual\_programming\_abilityset::Cyberdog::skeleton\_

骨骼（点）模块句柄

### 8.15.3.15 task\_

`Task` cyberdog\_visual\_programming\_abilityset::Cyberdog::task\_

任务模块句柄

### 8.15.3.16 tof\_

`Tof` cyberdog\_visual\_programming\_abilityset::Cyberdog::tof\_

Tof模块句柄

### 8.15.3.17 touch\_

`Touch` cyberdog\_visual\_programming\_abilityset::Cyberdog::touch\_

触摸板模块句柄

### 8.15.3.18 train\_

`Train` cyberdog\_visual\_programming\_abilityset::Cyberdog::train\_

训练模块句柄

### 8.15.3.19 ultrasonic\_

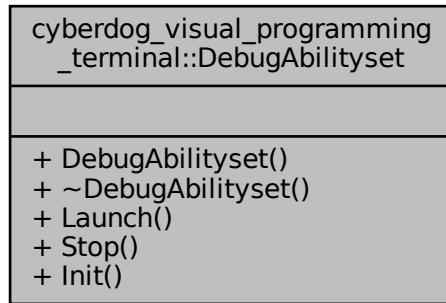
`Ultrasonic` cyberdog\_visual\_programming\_abilityset::Cyberdog::ultrasonic\_

超声波模块句柄

## 8.16 cyberdog\_visual\_programming\_terminal::DebugAbilityset类 参考

```
#include <debug_abilityset.hpp>
```

cyberdog\_visual\_programming\_terminal::DebugAbilityset 的协作图:



### Public 成员函数

- `DebugAbilityset ()`
- `~DebugAbilityset ()`
- `void Launch ()`
- `void Stop ()`
- `bool Init (const std::shared_ptr< VPA::Cyberdog > &)`

#### 8.16.1 构造及析构函数说明

##### 8.16.1.1 DebugAbilityset()

```
cyberdog_visual_programming_terminal::DebugAbilityset::DebugAbilityset ( )
```

##### 8.16.1.2 ~DebugAbilityset()

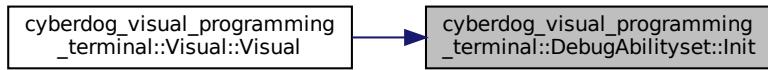
```
cyberdog_visual_programming_terminal::DebugAbilityset::~DebugAbilityset ( )
```

#### 8.16.2 成员函数说明

### 8.16.2.1 Init()

```
bool cyberdog_visual_programming_terminal::DebugAbilityset::Init ( const std::shared_ptr< VPA::Cyberdog > & _cyberdog_ptr )
```

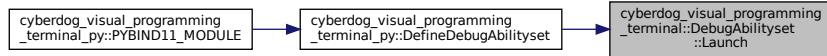
初始化 这是这个函数的调用关系图:



### 8.16.2.2 Launch()

```
void cyberdog_visual_programming_terminal::DebugAbilityset::Launch ( )
```

启动 这是这个函数的调用关系图:



### 8.16.2.3 Stop()

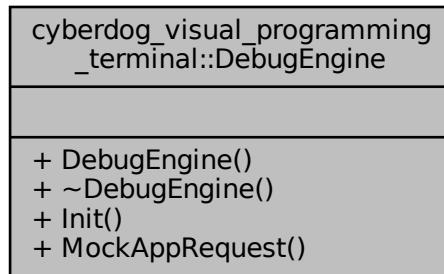
```
void cyberdog_visual_programming_terminal::DebugAbilityset::Stop ( )
```

停止

## 8.17 cyberdog\_visual\_programming\_terminal::DebugEngine类 参考

```
#include <debug_engine.hpp>
```

cyberdog\_visual\_programming\_terminal::DebugEngine 的协作图:



## Public 成员函数

- [DebugEngine \(\)](#)
- [~DebugEngine \(\)](#)
- [bool Init \(const std::string &\)](#)
- [void MockAppRequest \(const std::string &\)](#)

### 8.17.1 构造及析构函数说明

#### 8.17.1.1 [DebugEngine\(\)](#)

```
cyberdog_visual_programming_terminal::DebugEngine::DebugEngine ( )
```

#### 8.17.1.2 [~DebugEngine\(\)](#)

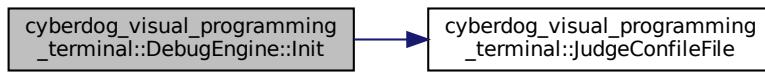
```
cyberdog_visual_programming_terminal::DebugEngine::~DebugEngine ( )
```

### 8.17.2 成员函数说明

### 8.17.2.1 Init()

```
bool cyberdog_visual_programming_terminal::DebugEngine::Init ( const std::string & _namespace )
```

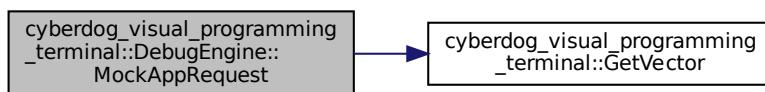
初始化 函数调用图:



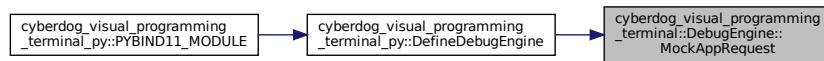
### 8.17.2.2 MockAppRequest()

```
void cyberdog_visual_programming_terminal::DebugEngine::MockAppRequest ( const std::string & _target )
```

模拟APP请求 函数调用图:



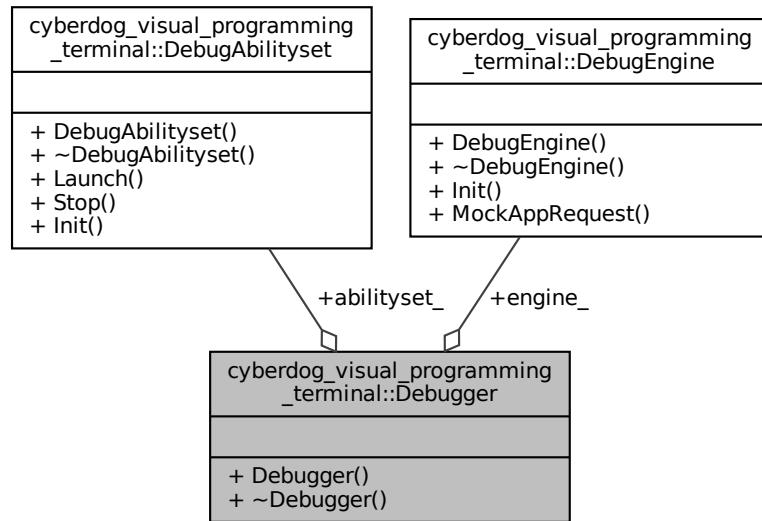
这是这个函数的调用关系图:



## 8.18 cyberdog\_visual\_programming\_terminal::Debugger类 参考

```
#include <debugger.hpp>
```

cyberdog\_visual\_programming\_terminal::Debugger 的协作图:



## Public 成员函数

- `Debugger ()`
- `~Debugger ()`

## Public 属性

- `DebugAbilityset abilityset_`
- `DebugEngine engine_`

### 8.18.1 构造及析构函数说明

#### 8.18.1.1 Debugger()

```
cyberdog_visual_programming_terminal::Debugger::Debugger ( )
```

#### 8.18.1.2 ~Debugger()

```
cyberdog_visual_programming_terminal::Debugger::~Debugger ( )
```

## 8.18.2 类成员变量说明

### 8.18.2.1 abilityset\_

`DebugAbilityset` `cyberdog_visual_programming_terminal::Debugger::abilityset_`

能力集

### 8.18.2.2 engine\_

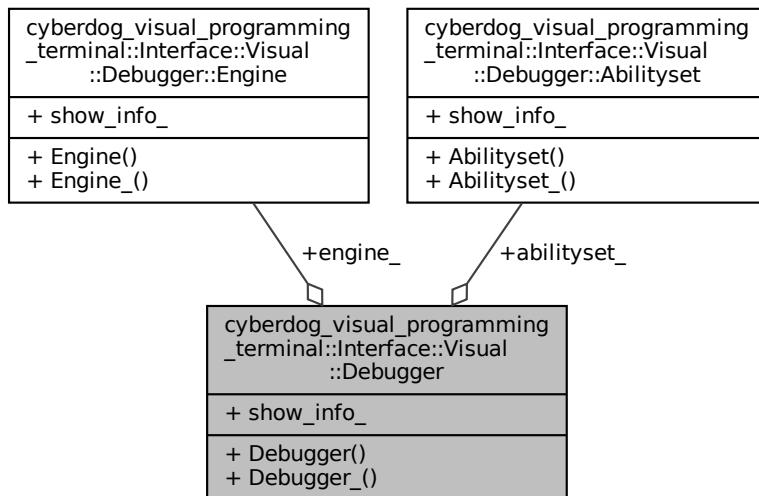
`DebugEngine` `cyberdog_visual_programming_terminal::Debugger::engine_`

引擎

## 8.19 `cyberdog_visual_programming_terminal::Interface::Visual::Debugger` 结构体 参考

#include <interface.hpp>

`cyberdog_visual_programming_terminal::Interface::Visual::Debugger` 的协作图:



类

- struct `Abilityset`
- struct `Engine`

## Public 成员函数

- `Debugger()`
- `void Debugger_(const std::string &fun)`

## Public 属性

- `std::function< void(std::string) > show_info_`
- `struct cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset abilityset_`
- `struct cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine engine_`

### 8.19.1 构造及析构函数说明

#### 8.19.1.1 Debugger()

```
cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Debugger() [inline]
```

### 8.19.2 成员函数说明

#### 8.19.2.1 Debugger\_()

```
void cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Debugger_(
    const std::string &fun) [inline]
```

### 8.19.3 类成员变量说明

#### 8.19.3.1 abilityset\_

```
struct cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset cyberdog_←
visual_programming_terminal::Interface::Visual::Debugger::abilityset_
```

#### 8.19.3.2 engine\_

```
struct cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine cyberdog_←
visual_programming_terminal::Interface::Visual::Debugger::engine_
```

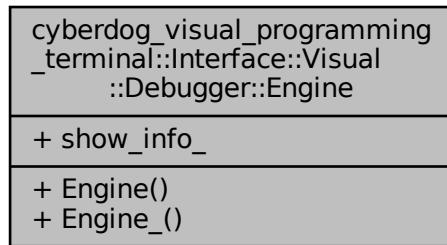
### 8.19.3.3 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Visual::Debugger::show_info_;
```

## 8.20 cyberdog\_visual\_programming\_terminal::Interface::Visual::Debugger::Engine结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Visual::Debugger::Engine 的协作图:



### Public 成员函数

- Engine ()
- void Engine\_ (const std::string &fun)

### Public 属性

- std::function< void(std::string)> show\_info\_;

## 8.20.1 构造及析构函数说明

### 8.20.1.1 Engine()

```
cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine::Engine ( ) [inline]
```

## 8.20.2 成员函数说明

### 8.20.2.1 Engine\_()

```
void cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine::Engine_ (
    const std::string _fun ) [inline]
```

## 8.20.3 类成员变量说明

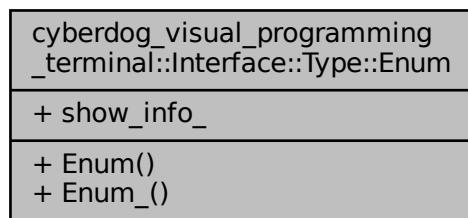
### 8.20.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Visual::←
Debugger::Engine::show_info_
```

## 8.21 cyberdog\_visual\_programming\_terminal::Interface::Type::Enum结构体参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Type::Enum 的协作图:



## Public 成员函数

- [Enum \(\)](#)
- [void \*\*Enum\\_\*\*\(const std::string \\_fun\)](#)

## Public 属性

- `std::function< void(std::string) > show_info_`

### 8.21.1 构造及析构函数说明

#### 8.21.1.1 `Enum()`

```
cyberdog_visual_programming_terminal::Interface::Type::Enum::Enum ( ) [inline]
```

### 8.21.2 成员函数说明

#### 8.21.2.1 `Enum_()`

```
void cyberdog_visual_programming_terminal::Interface::Type::Enum::Enum_ (
    const std::string &fun ) [inline]
```

### 8.21.3 类成员变量说明

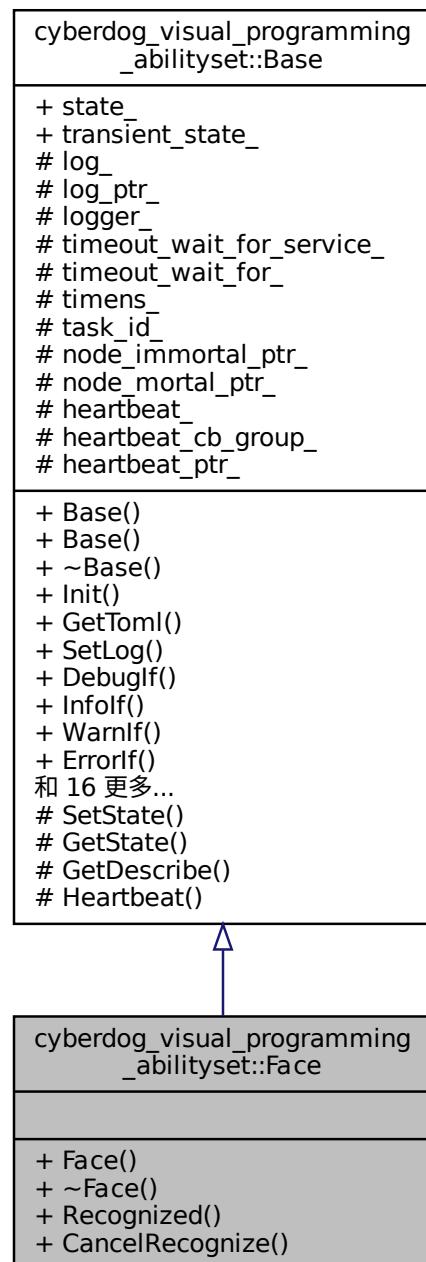
#### 8.21.3.1 `show_info_`

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Type::Enum::show_info_
```

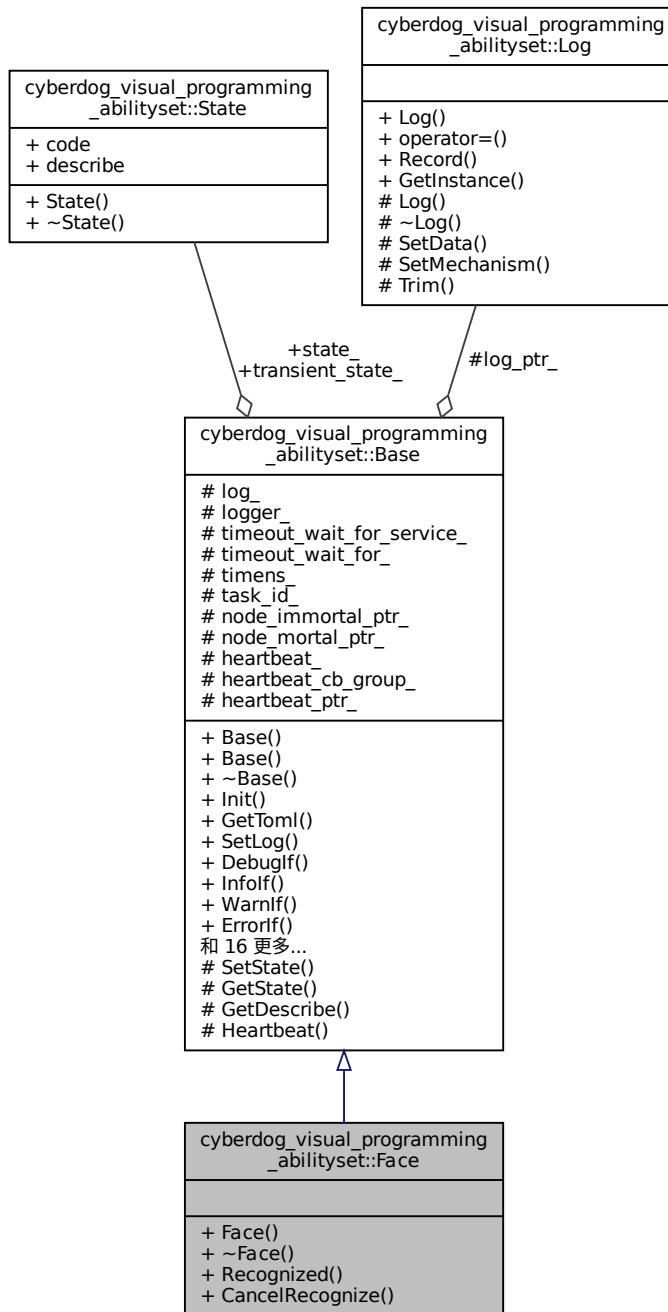
## 8.22 cyberdog\_visual\_programming\_abilityset::Face类 参考

```
#include <face.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Face 继承关系图:



cyberdog\_visual\_programming\_abilityset::Face 的协作图:



## Public 成员函数

- `Face ()`
- `~Face ()`
- `FaceRecognizedSeviceResponse Recognized (const std::vector< std::string > &, const bool _and_← operation=false, const int _duration=-1)`
- `FaceSeviceResponse CancelRecognize (const int _timeout=-1)`

## 额外继承的成员函数

### 8.22.1 构造及析构函数说明

#### 8.22.1.1 Face()

```
cyberdog_visual_programming_abilityset::Face::Face ( ) [inline]
```

#### 8.22.1.2 ~Face()

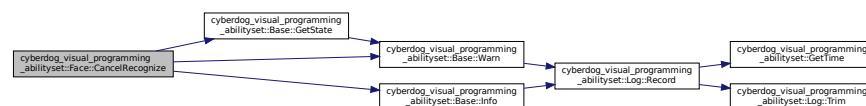
```
cyberdog_visual_programming_abilityset::Face::~Face ( ) [inline]
```

## 8.22.2 成员函数说明

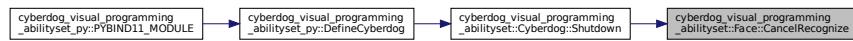
### 8.22.2.1 CancelRecognize()

```
FaceServiceResponse cyberdog_visual_programming_abilityset::Face::CancelRecognize (
    const int _timeout = -1 )
```

识别到目标人员人脸 函数调用图:



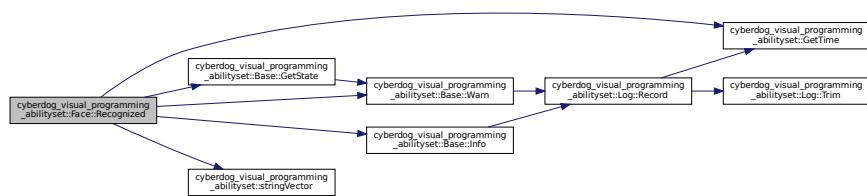
这是这个函数的调用关系图:



### 8.22.2.2 Recognized()

```
FaceRecognizedServiceResponse cyberdog_visual_programming_abilityset::Face::Recognized (
    const std::vector< std::string > & _voiceprint_target,
    const bool _and_operation = false,
    const int _duration = -1 )
```

识别到目标人员人脸 函数调用图:



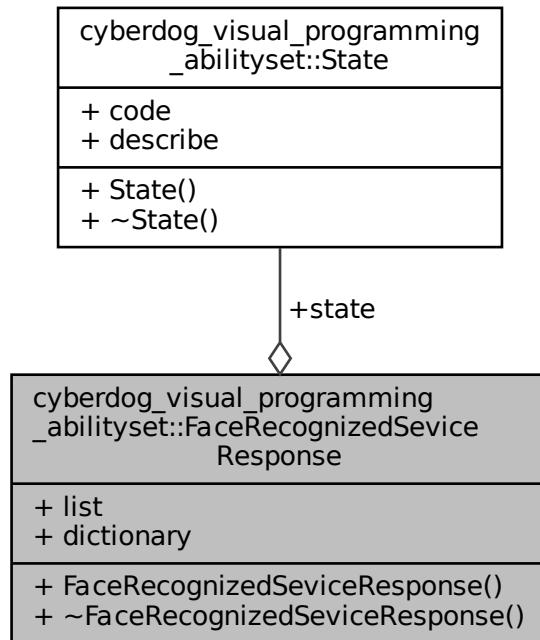
这是这个函数的调用关系图:



## 8.23 cyberdog\_visual\_programming\_abilityset::FaceRecognizedServiceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::FaceRecognizedSeviceResponse 的协作图:



## Public 成员函数

- `FaceRecognizedSeviceResponse ()`
- `~FaceRecognizedSeviceResponse ()`

## Public 属性

- `State state`
- `std::vector< MsgFaceRes > list`
- `std::map< std::string, MsgFaceRes > dictionary`

### 8.23.1 详细描述

人脸识别服务反馈

### 8.23.2 构造及析构函数说明

### 8.23.2.1 FaceRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse::FaceRecognizedSeviceResponse ( ) [inline]
```

### 8.23.2.2 ~FaceRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse::~FaceRecognizedSeviceResponse ( ) [inline]
```

## 8.23.3 类成员变量说明

### 8.23.3.1 dictionary

```
std::map<std::string, MsgFaceRes> cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse::dictionary
```

字典

### 8.23.3.2 list

```
std::vector<MsgFaceRes> cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse::list
```

列表

### 8.23.3.3 state

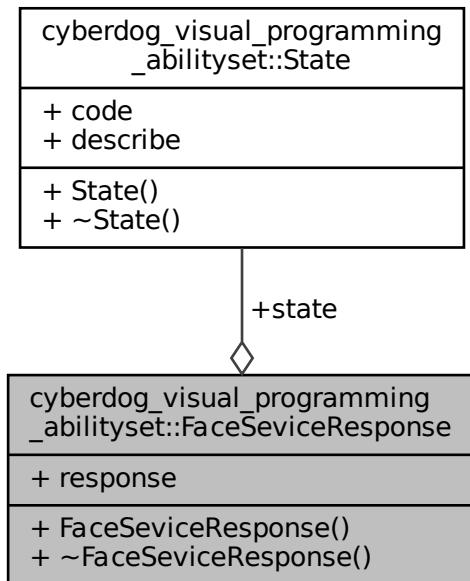
```
State cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse::state
```

状态

## 8.24 cyberdog\_visual\_programming\_abilityset::FaceSeviceResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::FaceSeviceResponse 的协作图:



### Public 成员函数

- `FaceSeviceResponse ()`
- `~FaceSeviceResponse ()`

### Public 属性

- `State state`
- `SrvFaceRec::Response response`

#### 8.24.1 详细描述

人脸识别服务反馈

#### 8.24.2 构造及析构函数说明

### 8.24.2.1 FaceSeviceResponse()

```
cyberdog_visual_programming_abilityset::FaceSeviceResponse::FaceSeviceResponse () [inline]
```

### 8.24.2.2 ~FaceSeviceResponse()

```
cyberdog_visual_programming_abilityset::FaceSeviceResponse::~FaceSeviceResponse () [inline]
```

## 8.24.3 类成员变量说明

### 8.24.3.1 response

```
SrvFaceRec::Response cyberdog_visual_programming_abilityset::FaceSeviceResponse::response
```

反馈

### 8.24.3.2 state

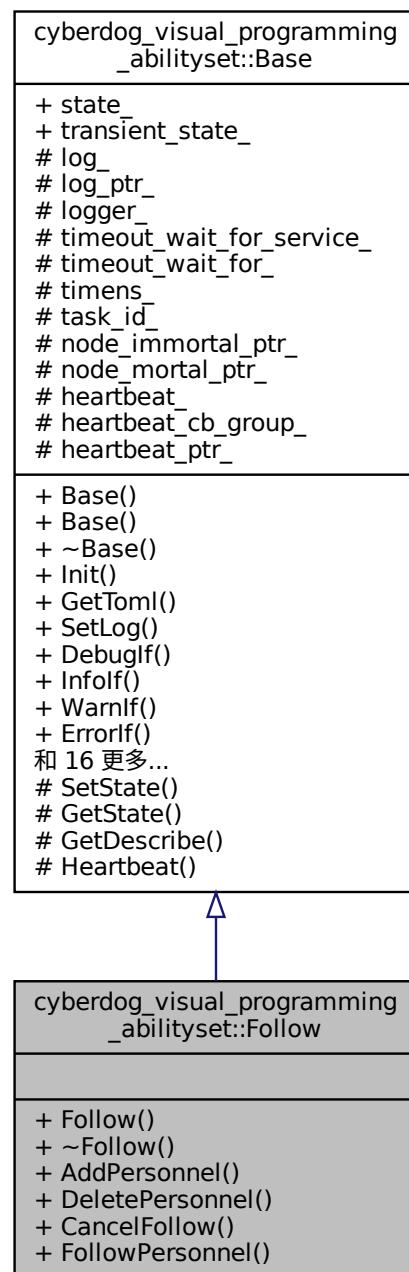
```
State cyberdog_visual_programming_abilityset::FaceSeviceResponse::state
```

状态

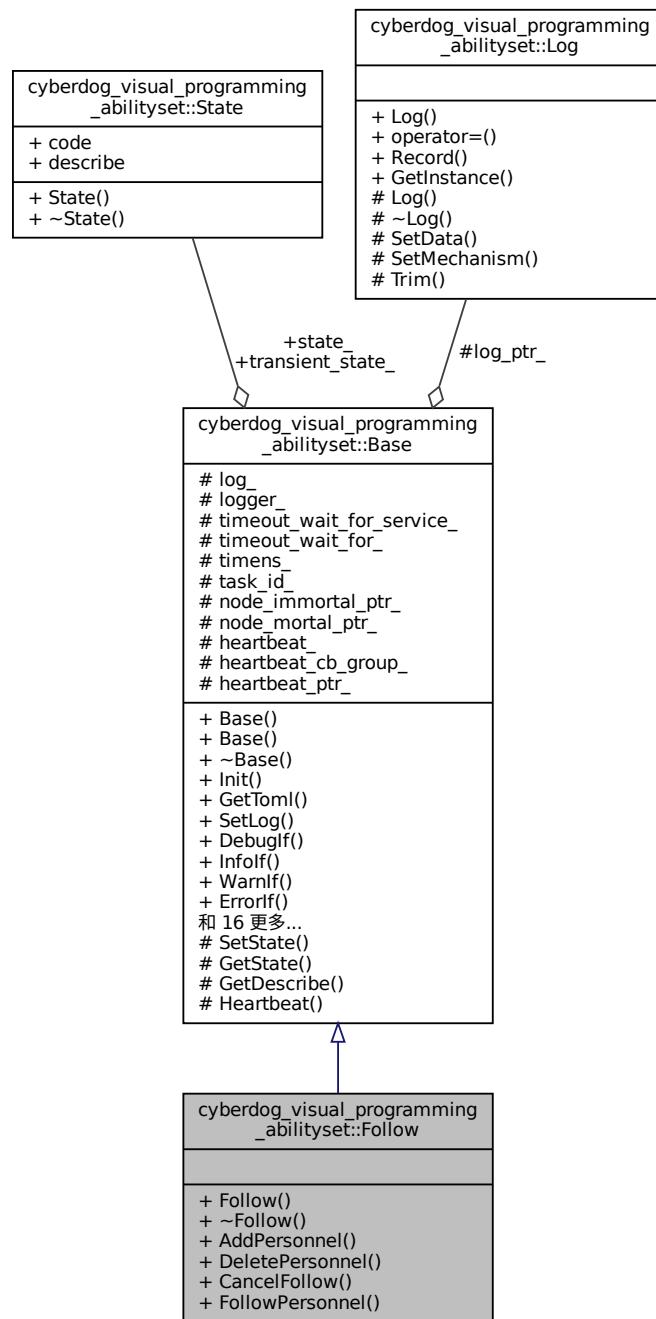
## 8.25 cyberdog\_visual\_programming\_abilityset::Follow类 参考

```
#include <follow.hpp>
```

类 `cyberdog_visual_programming_abilityset::Follow` 继承关系图:



cyberdog\_visual\_programming\_abilityset::Follow 的协作图:



## Public 成员函数

- **Follow ()**
- **~Follow ()**
- **State AddPersonnel (const std::string)**
- **State DeletePersonnel (const std::string)**
- **State CancelFollow ()**
- **State FollowPersonnel (const std::string, const double)**

## 额外继承的成员函数

### 8.25.1 构造及析构函数说明

#### 8.25.1.1 Follow()

```
cyberdog_visual_programming_abilityset::Follow::Follow () [inline]
```

#### 8.25.1.2 ~Follow()

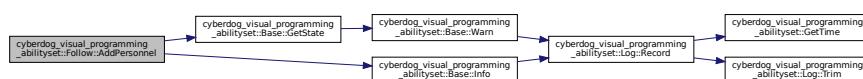
```
cyberdog_visual_programming_abilityset::Follow::~Follow () [inline]
```

### 8.25.2 成员函数说明

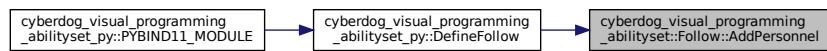
#### 8.25.2.1 AddPersonnel()

```
State cyberdog_visual_programming_abilityset::Follow::AddPersonnel (
    const std::string &_preset_name )
```

添加人员 函数调用图:



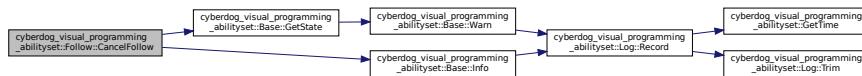
这是这个函数的调用关系图:



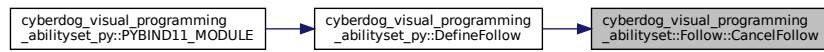
### 8.25.2.2 CancelFollow()

```
State cyberdog_visual_programming_abilityset::Follow::CancelFollow ( )
```

取消跟随 函数调用图:



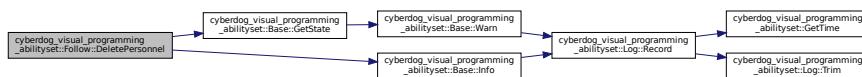
这是这个函数的调用关系图:



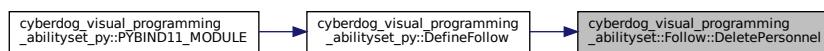
### 8.25.2.3 DeletePersonnel()

```
State cyberdog_visual_programming_abilityset::Follow::DeletePersonnel (
    const std::string & preset_name )
```

删除人员 函数调用图:



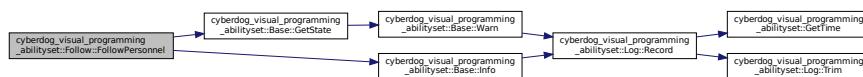
这是这个函数的调用关系图:



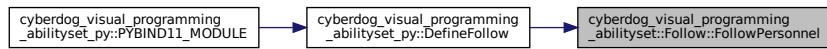
#### 8.25.2.4 FollowPersonnel()

```
State cyberdog_visual_programming_abilityset::Follow::FollowPersonnel (
    const std::string _preset_name,
    const double _intimacy )
```

跟踪人员 函数调用图:



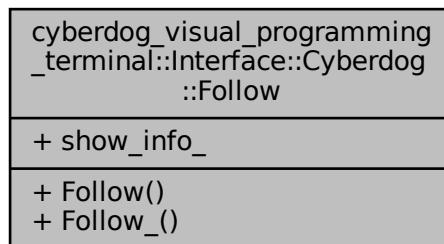
这是这个函数的调用关系图:



## 8.26 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog<-->::Follow 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Follow 的协作图:



### Public 成员函数

- `Follow ()`
- `void Follow_ (const std::string _fun)`

**Public 属性**

- std::function< void(std::string) > [show\\_info\\_](#)

**8.26.1 构造及析构函数说明****8.26.1.1 Follow()**

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow::Follow ( ) [inline]
```

**8.26.2 成员函数说明****8.26.2.1 Follow\_()**

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow::Follow_ (
    const std::string _fun ) [inline]
```

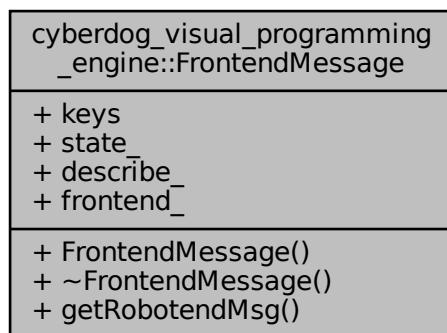
**8.26.3 类成员变量说明****8.26.3.1 show\_info\_**

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow::show.info_
```

**8.27 cyberdog\_visual\_programming\_engine::FrontendMessage类 参考**

```
#include <frontend_message.hpp>
```

cyberdog\_visual\_programming\_engine::FrontendMessage 的协作图:



## Public 类型

- using `CommonEnum` = enum { efficient=0, invalid, type, id, target\_id, describe, style, operate, mode, condition, body, }

## Public 成员函数

- `FrontendMessage` (const std::string &msg)
- `~FrontendMessage` ()
- void `getRobotendMsg` (GRPCMMsg &)

## Public 属性

- const std::unordered\_map< `CommonEnum`, std::string > `keys`
- `CommonEnum state_`
- std::string `describe_` {"")}
- `OperateMsg frontend_`

### 8.27.1 成员类型定义说明

#### 8.27.1.1 CommonEnum

```
using cyberdog_visual_programming_engine::FrontendMessage::CommonEnum = enum { efficient = 0,
invalid, type, id, target_id, describe, style, operate, mode, condition, body, }
```

### 8.27.2 构造及析构函数说明

#### 8.27.2.1 FrontendMessage()

```
cyberdog_visual_programming_engine::FrontendMessage::FrontendMessage (
    const std::string & msg ) [explicit]
```

#### 8.27.2.2 ~FrontendMessage()

```
cyberdog_visual_programming_engine::FrontendMessage::~FrontendMessage ( ) [inline]
```

### 8.27.3 成员函数说明

### 8.27.3.1 getRobotendMsg()

```
void cyberdog_visual_programming_engine::FrontendMessage::getRobotendMsg (
```

GRPCMsg & msg\_ )

获取机器人端消息

## 8.27.4 类成员变量说明

### 8.27.4.1 describe\_

```
std::string cyberdog_visual_programming_engine::FrontendMessage::describe_ {"")}
```

描述

### 8.27.4.2 frontend\_

```
OperateMsg cyberdog_visual_programming_engine::FrontendMessage::frontend_
```

前端消息

### 8.27.4.3 keys

```
const std::unordered_map<CommonEnum, std::string> cyberdog_visual_programming_engine::FrontendMessage::keys
```

初始值:

```
= {
```

{CommonEnum::type, "type"},  
  {CommonEnum::id, "id"},  
  {CommonEnum::target\_id, "target\_id"},  
  {CommonEnum::describe, "describe"},  
  {CommonEnum::style, "style"},  
  {CommonEnum::operate, "operate"},  
  {CommonEnum::mode, "mode"},  
  {CommonEnum::condition, "condition"},  
  {CommonEnum::body, "body"}  
}

### 8.27.4.4 state\_

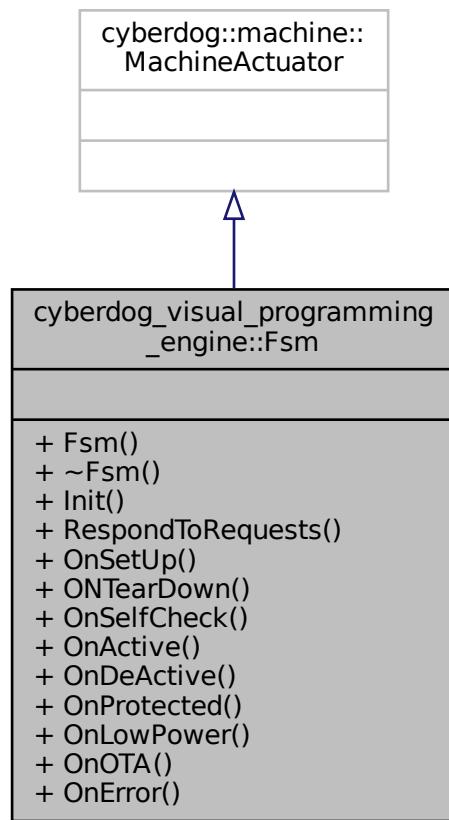
```
CommonEnum cyberdog_visual_programming_engine::FrontendMessage::state_
```

状态

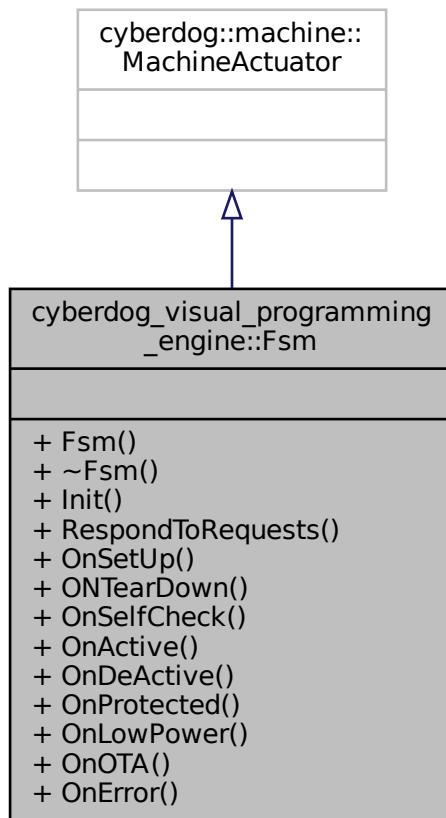
## 8.28 cyberdog\_visual\_programming\_engine::Fsm类 参考

```
#include <fsm.hpp>
```

类 cyberdog\_visual\_programming\_engine::Fsm 继承关系图:



cyberdog\_visual\_programming\_engine::Fsm 的协作图:



## Public 成员函数

- `Fsm()`
- `~Fsm()`
- `bool Init(const rclcpp::Node::SharedPtr &, const toml::value &)`
- `bool RespondToRequests(const OperateMsg &, GRPCMsg &)`
- `int32_t OnSetUp()`
- `int32_t ONTearDown()`
- `int32_t OnSelfCheck()`
- `int32_t OnActive()`
- `int32_t OnDeActive()`
- `int32_t OnProtected()`
- `int32_t OnLowPower()`
- `int32_t OnOTA()`
- `int32_t OnError()`

### 8.28.1 构造及析构函数说明

### 8.28.1.1 Fsm()

```
cyberdog_visual_programming_engine::Fsm::Fsm ( )
```

### 8.28.1.2 ~Fsm()

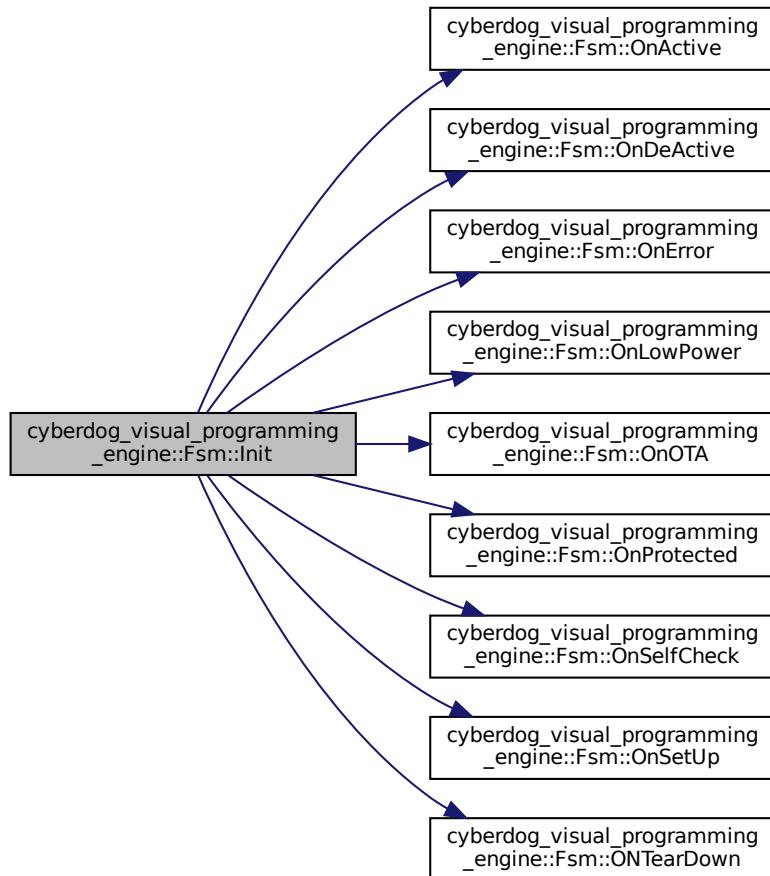
```
cyberdog_visual_programming_engine::Fsm::~Fsm ( )
```

## 8.28.2 成员函数说明

### 8.28.2.1 Init()

```
bool cyberdog_visual_programming_engine::Fsm::Init (
    const rclcpp::Node::SharedPtr & _node_ptr,
    const toml::value & _params_toml )
```

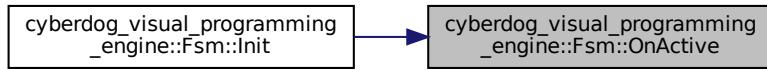
初始化 函数调用图:



### 8.28.2.2 OnActive()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnActive ( )
```

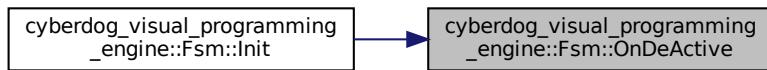
Active 这是这个函数的调用关系图:



### 8.28.2.3 OnDeActive()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnDeActive ( )
```

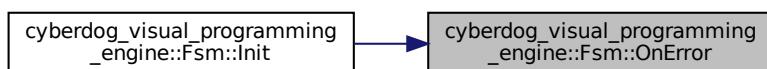
DeActive 这是这个函数的调用关系图:



### 8.28.2.4 OnError()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnError ( )
```

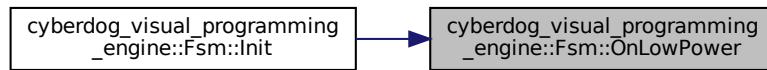
Error 这是这个函数的调用关系图:



### 8.28.2.5 OnLowPower()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnLowPower ( )
```

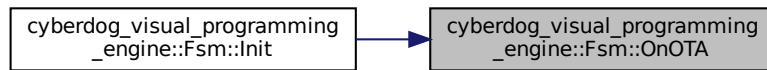
LowPower 这是这个函数的调用关系图:



### 8.28.2.6 OnOTA()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnOTA ( )
```

OTA 这是这个函数的调用关系图:



### 8.28.2.7 OnProtected()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnProtected ( )
```

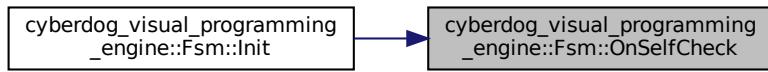
Protected 这是这个函数的调用关系图:



### 8.28.2.8 OnSelfCheck()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnSelfCheck ( )
```

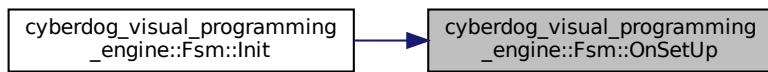
SelfCheck 这是这个函数的调用关系图:



### 8.28.2.9 OnSetUp()

```
int32_t cyberdog_visual_programming_engine::Fsm::OnSetUp ( )
```

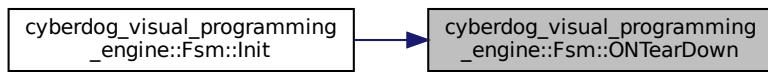
SetUp 这是这个函数的调用关系图:



### 8.28.2.10 ONTearDown()

```
int32_t cyberdog_visual_programming_engine::Fsm::ONTearDown ( )
```

TearDown 这是这个函数的调用关系图:



### 8.28.2.11 RespondToRequests()

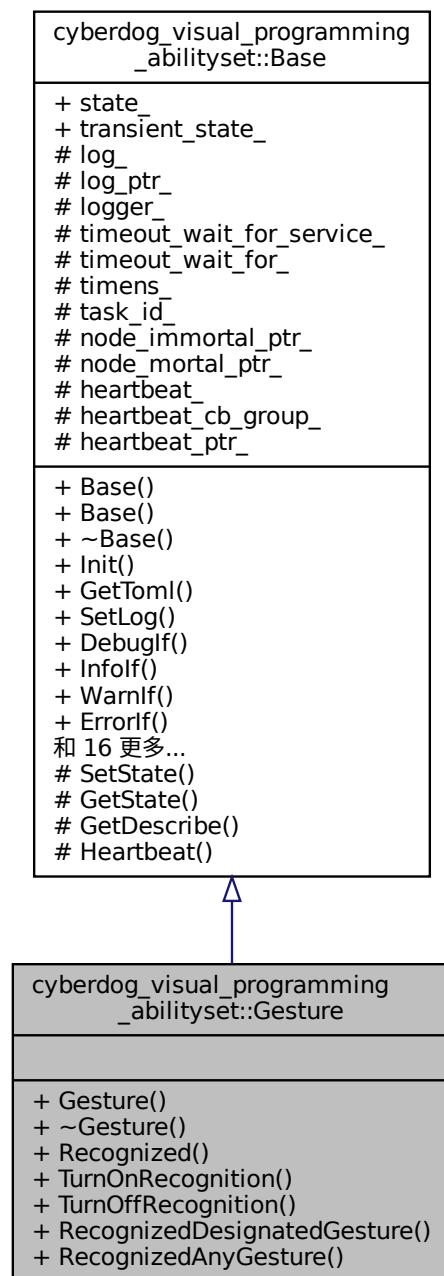
```
bool cyberdog_visual_programming_engine::Fsm::RespondToRequests (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

执行请求

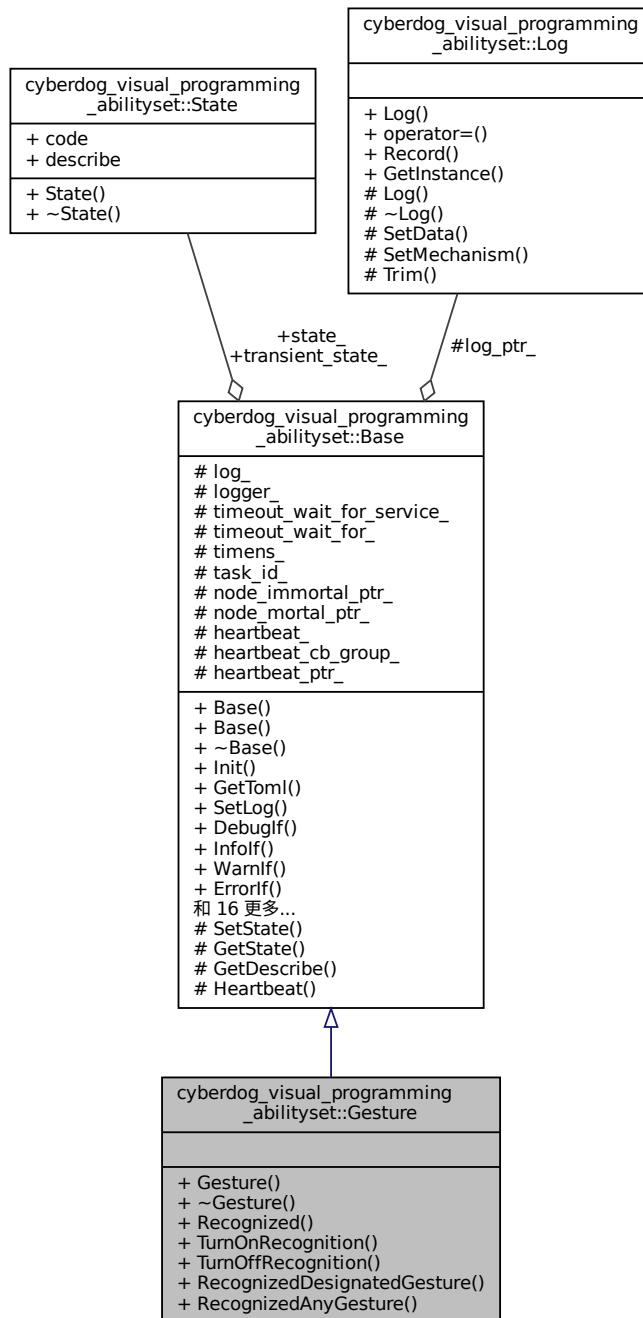
## 8.29 cyberdog\_visual\_programming\_abilityset::Gesture类 参考

```
#include <gesture.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Gesture 继承关系图:



cyberdog\_visual\_programming\_abilityset::Gesture 的协作图:



## Public 成员函数

- `Gesture ()`
- `~Gesture ()`
- `GestureRecognizedMessageResponse Recognized (const int _duration=-1, const int _sensitivity=1)`
- `GestureRecognizedSeviceResponse TurnOnRecognition (const int _duration=-1)`
- `GestureRecognizedSeviceResponse TurnOffRecognition ()`

- GestureRecognizedMessageResponse RecognizedDesignatedGesture (const int timeout=-1, const int gesture\_type=0)
- GestureRecognizedMessageResponse RecognizedAnyGesture (const int timeout=-1, const int sensitivity=1)

额外继承的成员函数

### 8.29.1 构造及析构函数说明

#### 8.29.1.1 Gesture()

```
cyberdog_visual_programming_abilityset::Gesture::Gesture ( ) [inline]
```

#### 8.29.1.2 ~Gesture()

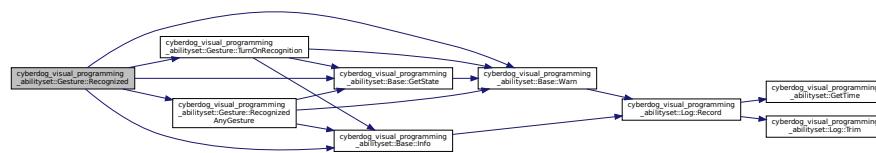
```
cyberdog_visual_programming_abilityset::Gesture::~Gesture ( ) [inline]
```

### 8.29.2 成员函数说明

#### 8.29.2.1 Recognized()

```
GestureRecognizedMessageResponse cyberdog_visual_programming_abilityset::Gesture::Recognized (
    const int _duration = -1,
    const int _sensitivity = 1 )
```

开始识别并识别到任意手势 函数调用图:



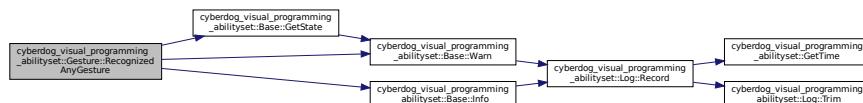
这是这个函数的调用关系图:



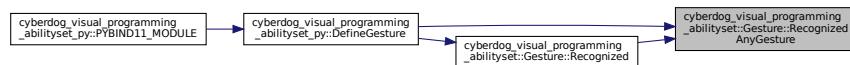
### 8.29.2.2 RecognizedAnyGesture()

```
GestureRecognizedMessageResponse cyberdog_visual_programming_abilityset::Gesture::Recognized<->
AnyGesture (
    const int timeout = -1,
    const int sensitivity = 1 )
```

识别到任意手势 函数调用图:



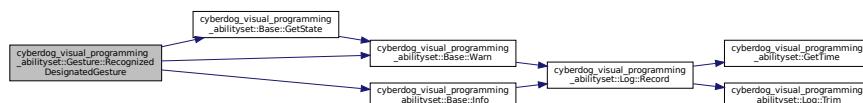
这是这个函数的调用关系图:



### 8.29.2.3 RecognizedDesignatedGesture()

```
GestureRecognizedMessageResponse cyberdog_visual_programming_abilityset::Gesture::Recognized<->
DesignatedGesture (
    const int timeout = -1,
    const int gesture_type = 0 )
```

识别到指定手势 函数调用图:



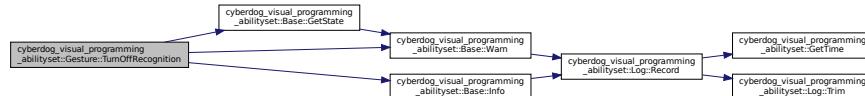
这是这个函数的调用关系图:



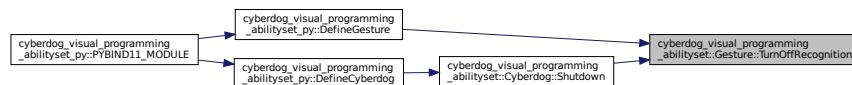
### 8.29.2.4 TurnOffRecognition()

```
GestureRecognizedServiceResponse cyberdog_visual_programming_abilityset::Gesture::TurnOffRecognition()
```

关闭识别手势功能 函数调用图:



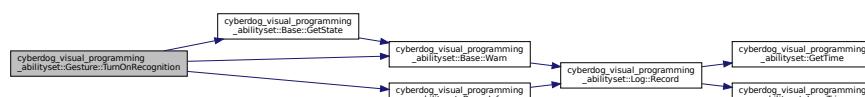
这是这个函数的调用关系图:



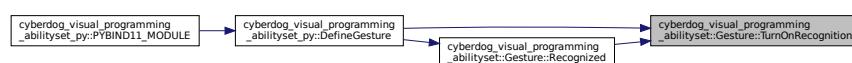
### 8.29.2.5 TurnOnRecognition()

```
GestureRecognizedServiceResponse cyberdog_visual_programming_abilityset::Gesture::TurnOnRecognition()
{
    const int _duration = -1
}
```

打开识别手势功能 函数调用图:



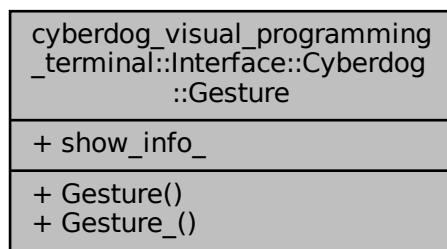
这是这个函数的调用关系图:



## 8.30 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Gesture 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Gesture 的协作图:



### Public 成员函数

- `Gesture ()`
- `void Gesture_(const std::string &fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

## 8.30.1 构造及析构函数说明

### 8.30.1.1 Gesture()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture::Gesture ( ) [inline]
```

## 8.30.2 成员函数说明

### 8.30.2.1 Gesture\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture::Gesture_ (
    const std::string _fun ) [inline]
```

## 8.30.3 类成员变量说明

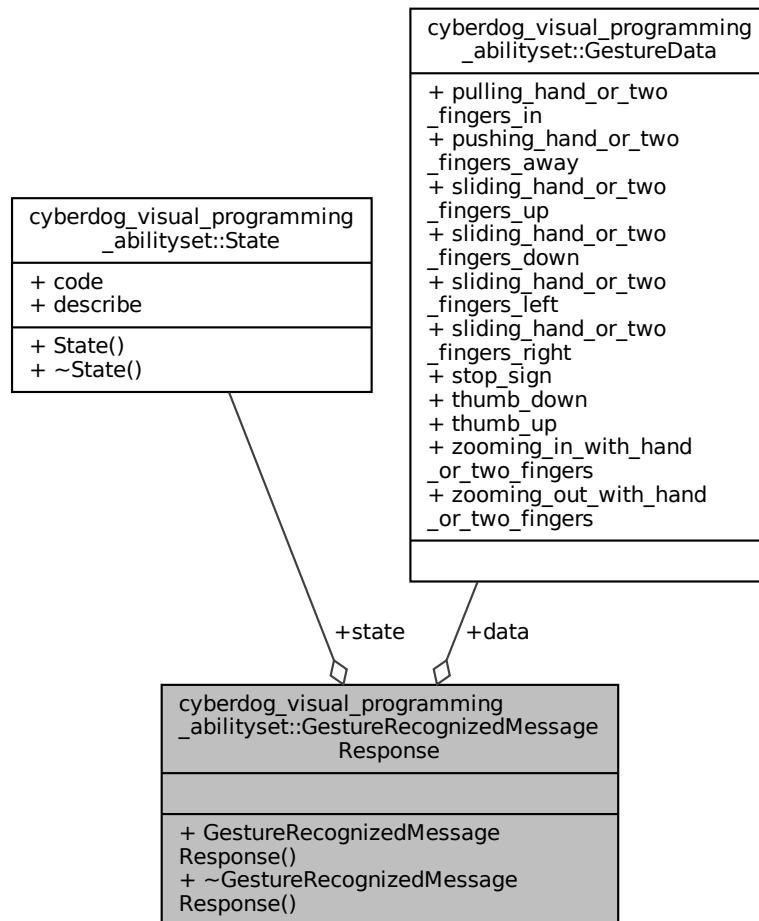
### 8.30.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_
```

## 8.31 **cyberdog\_visual\_programming\_abilityset::GestureRecognizedMessageResponse**类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::GestureRecognizedMessageResponse 的协作图:



## Public 成员函数

- `GestureRecognizedMessageResponse ()`
- `~GestureRecognizedMessageResponse ()`

## Public 属性

- `State state`
- `GestureData data`

### 8.31.1 详细描述

手势识别返回信息

### 8.31.2 构造及析构函数说明

#### 8.31.2.1 GestureRecognizedMessageResponse()

```
cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse::GestureRecognizedMessageResponse ( ) [inline]
```

#### 8.31.2.2 ~GestureRecognizedMessageResponse()

```
cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse::~GestureRecognizedMessageResponse ( ) [inline]
```

### 8.31.3 类成员变量说明

#### 8.31.3.1 data

```
GestureData cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse::data
```

数据

#### 8.31.3.2 state

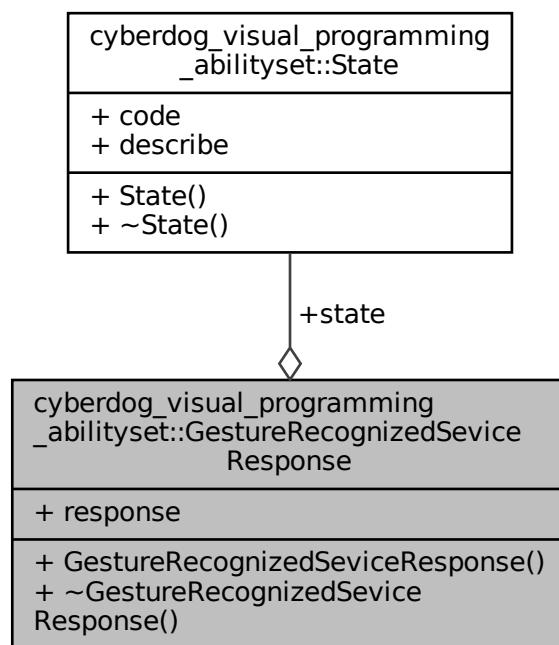
```
State cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse::state
```

状态

## 8.32 cyberdog\_visual\_programming\_abilityset::GestureRecognizedServiceResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::GestureRecognizedServiceResponse 的协作图:



### Public 成员函数

- `GestureRecognizedServiceResponse ()`
- `~GestureRecognizedServiceResponse ()`

### Public 属性

- `State state`
- `SrvGesture::Response response`

#### 8.32.1 详细描述

手势识别服务反馈

## 8.32.2 构造及析构函数说明

### 8.32.2.1 GestureRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse::GestureRecognizedSeviceResponse ( ) [inline]
```

### 8.32.2.2 ~GestureRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse::~GestureRecognizedSeviceResponse ( ) [inline]
```

## 8.32.3 类成员变量说明

### 8.32.3.1 response

```
SrvGesture::Response cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse::response
```

反馈

### 8.32.3.2 state

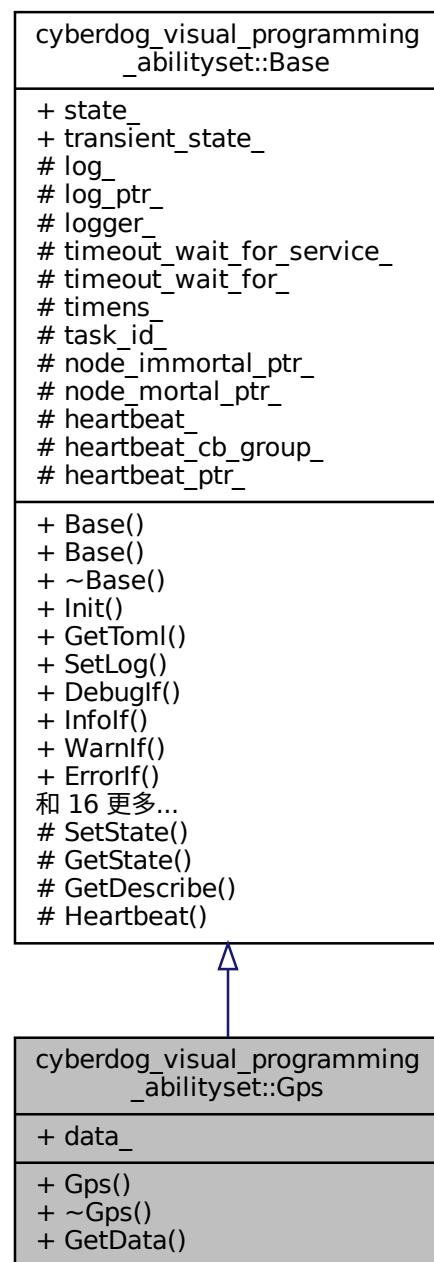
```
State cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse::state
```

状态

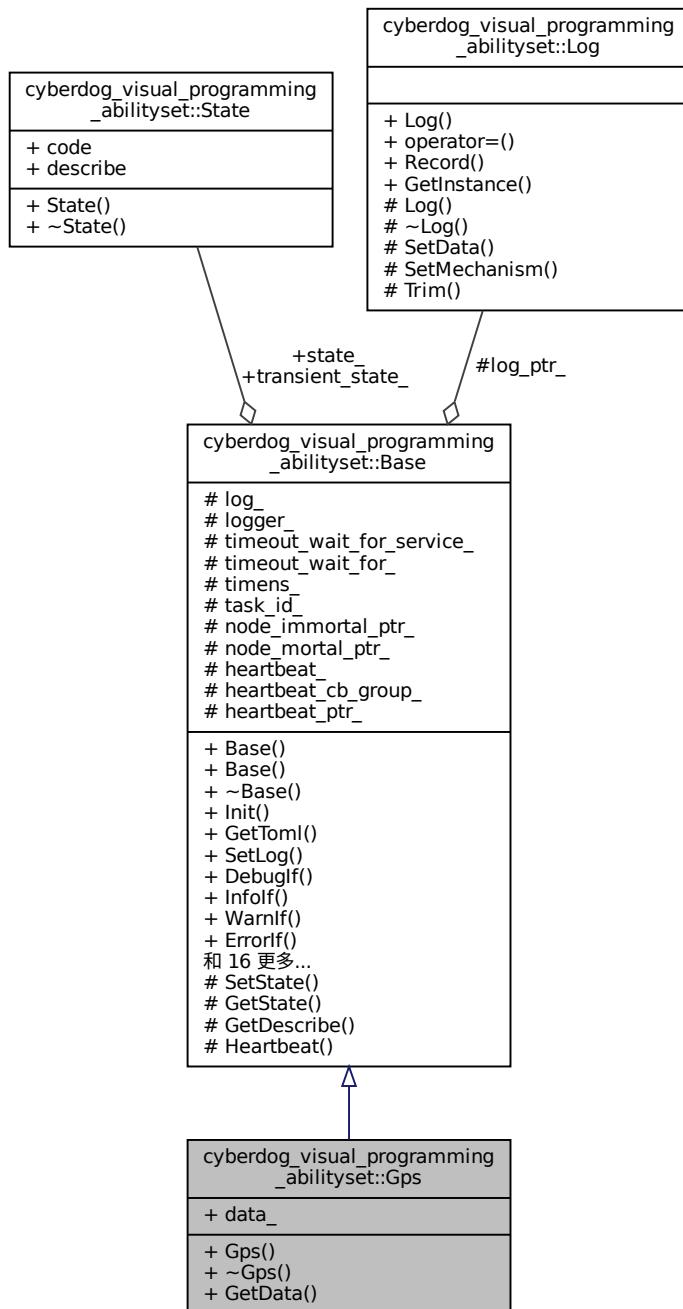
### 8.33 cyberdog\_visual\_programming\_abilityset::Gps类参考

```
#include <gps.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Gps 继承关系图:



cyberdog\_visual\_programming\_abilityset::Gps 的协作图:



## Public 成员函数

- [Gps \(\)](#)
- [~Gps \(\)](#)
- [MsgGpsPayload GetData \(const int \\_timeout=5\)](#)

## Public 属性

- `MsgGpsPayload data_`

额外继承的成员函数

### 8.33.1 构造及析构函数说明

#### 8.33.1.1 Gps()

```
cyberdog_visual_programming_abilityset::Gps::Gps ( ) [inline]
```

#### 8.33.1.2 ~Gps()

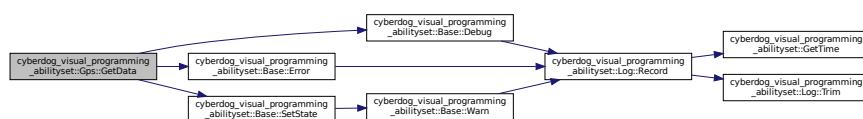
```
cyberdog_visual_programming_abilityset::Gps::~Gps ( ) [inline]
```

### 8.33.2 成员函数说明

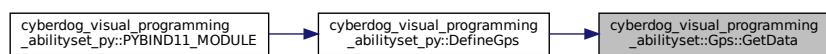
#### 8.33.2.1 GetData()

```
MsgGpsPayload cyberdog_visual_programming_abilityset::Gps::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.33.3 类成员变量说明

#### 8.33.3.1 data\_

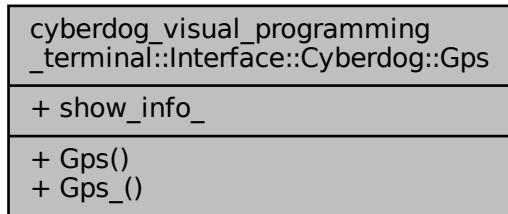
`MsgGpsPayload` `cyberdog_visual_programming_abilityset::Gps::data_`

数据

## 8.34 `cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps` 结构体 参考

#include <interface.hpp>

`cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps` 的协作图:



### Public 成员函数

- `Gps ()`
- `void Gps_ (const std::string _fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

### 8.34.1 构造及析构函数说明

### 8.34.1.1 Gps()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps::Gps ( ) [inline]
```

## 8.34.2 成员函数说明

### 8.34.2.1 Gps\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps::Gps_ (
    const std::string _fun ) [inline]
```

## 8.34.3 类成员变量说明

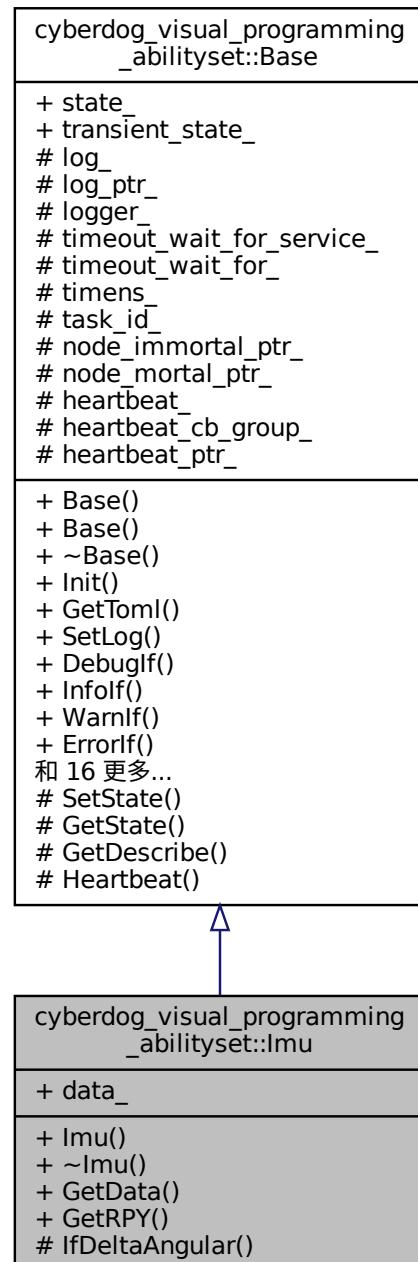
### 8.34.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Gps::show_info_
```

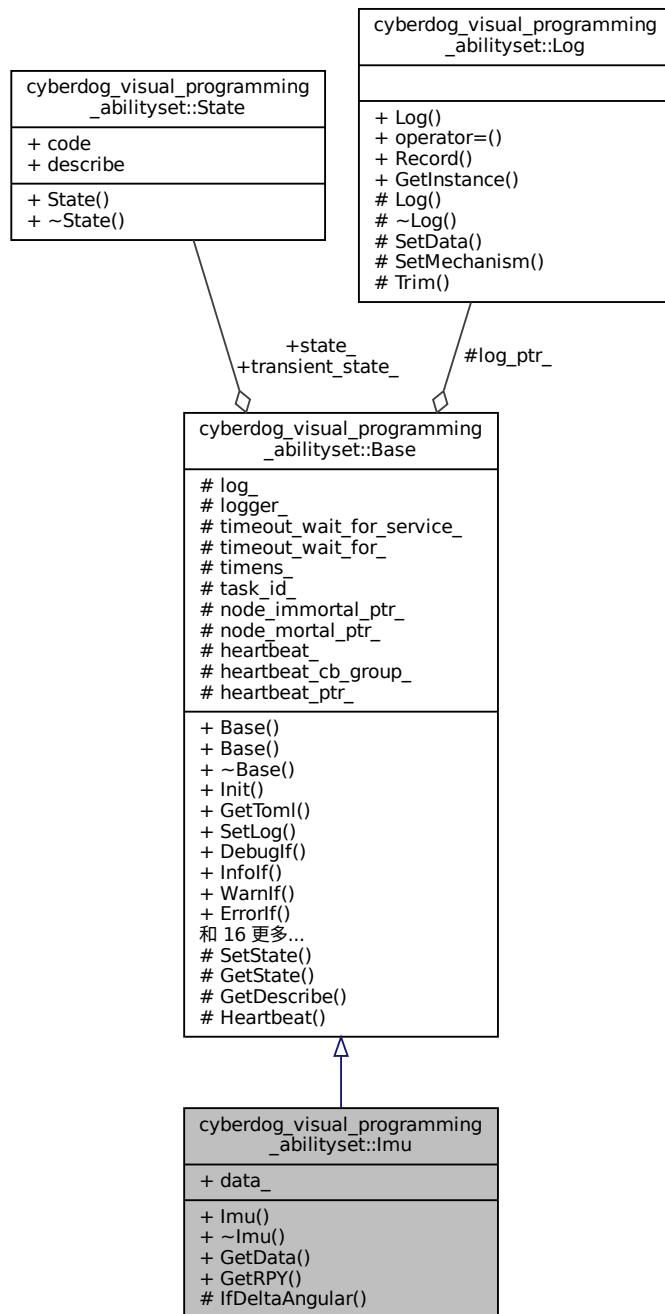
## 8.35 cyberdog\_visual\_programming\_abilityset::Imu类 参考

```
#include <imu.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Imu 继承关系图:



cyberdog\_visual\_programming\_abilityset::Imu 的协作图:



## Public 成员函数

- `Imu ()`
- `~Imu ()`
- `MsgImu GetData (const int _timeout=5)`
- `RPY GetRPY ()`

## Public 属性

- [MsgImu data\\_](#)

## Protected 成员函数

- bool [IfDeltaAngular \(const MsgImu &, const double, RPYType \\_type=RPYType::YAW\)](#)

额外继承的成员函数

### 8.35.1 构造及析构函数说明

#### 8.35.1.1 Imu()

```
cyberdog_visual_programming_abilityset::Imu::Imu ( ) [inline]
```

#### 8.35.1.2 ~Imu()

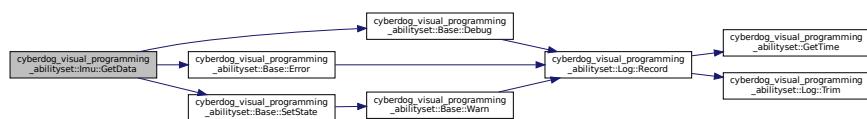
```
cyberdog_visual_programming_abilityset::Imu::~Imu ( ) [inline]
```

### 8.35.2 成员函数说明

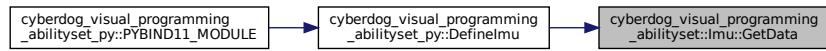
#### 8.35.2.1 GetData()

```
MsgImu cyberdog_visual_programming_abilityset::Imu::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



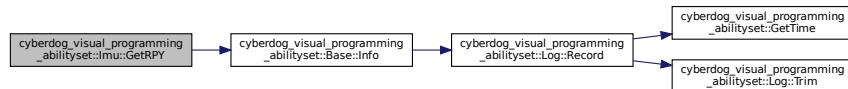
这是这个函数的调用关系图:



### 8.35.2.2 GetRPY()

```
RPY cyberdog_visual_programming_abilityset::Imu::GetRPY ( )
```

获取欧拉角 函数调用图:



### 8.35.2.3 IfDeltaAngular()

```
bool cyberdog_visual_programming_abilityset::Imu::IfDeltaAngular (
    const MsgImu & _imu,
    const double _angular,
    RPYType _type = RPYType::YAW ) [protected]
```

判断角度

## 8.35.3 类成员变量说明

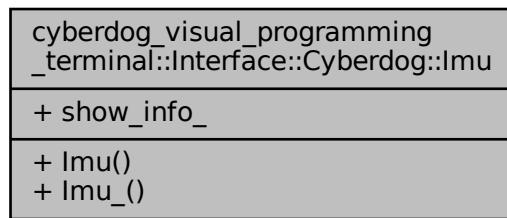
### 8.35.3.1 data\_

```
MsgImu cyberdog_visual_programming_abilityset::Imu::data_
IMU
```

## 8.36 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog<-- ::Imu结构体 参 考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Imu 的协作图:



## Public 成员函数

- `Imu()`
- `void Imu_(const std::string &fun)`

## Public 属性

- `std::function<void(std::string)> show_info_`

### 8.36.1 构造及析构函数说明

#### 8.36.1.1 `Imu()`

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu::Imu () [inline]
```

### 8.36.2 成员函数说明

#### 8.36.2.1 `Imu_()`

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu::Imu_ (
    const std::string &fun) [inline]
```

### 8.36.3 类成员变量说明

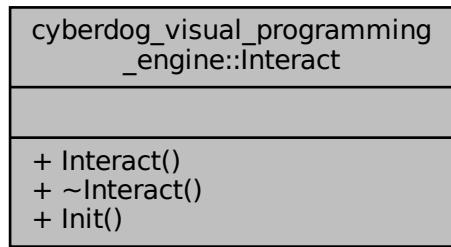
#### 8.36.3.1 `show_info_`

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu::show_info_
```

## 8.37 cyberdog\_visual\_programming\_engine::Interact类 参考

```
#include <interact.hpp>
```

cyberdog\_visual\_programming\_engine::Interact 的协作图:



### Public 成员函数

- `Interact ()`
- `~Interact ()`
- `bool Init ()`

#### 8.37.1 构造及析构函数说明

##### 8.37.1.1 Interact()

```
cyberdog_visual_programming_engine::Interact::Interact ( )
```

函数调用图:



### 8.37.1.2 ~Interact()

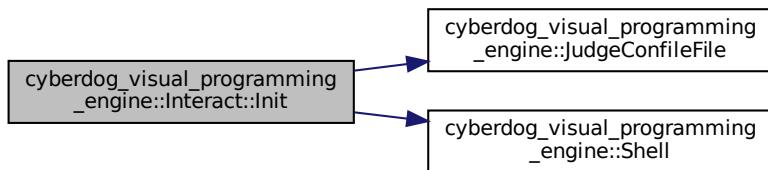
```
cyberdog_visual_programming_engine::Interact::~Interact ( )
```

## 8.37.2 成员函数说明

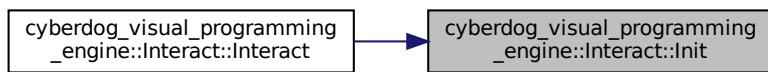
### 8.37.2.1 Init()

```
bool cyberdog_visual_programming_engine::Interact::Init ( )
```

初始化 函数调用图:



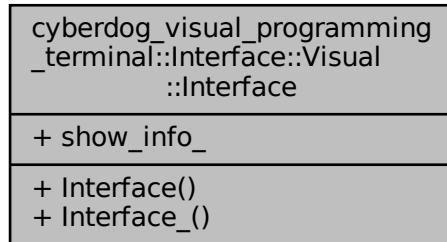
这是这个函数的调用关系图:



## 8.38 cyberdog\_visual\_programming\_terminal::Interface::Visual::Interface 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Visual::Interface 的协作图:



## Public 成员函数

- `Interface ()`
- `void Interface_(const std::string &fun)`

## Public 属性

- `std::function< void(std::string) > show_info_`

### 8.38.1 构造及析构函数说明

#### 8.38.1.1 Interface()

```
cyberdog_visual_programming_terminal::Interface::Visual::Interface () [inline]
```

### 8.38.2 成员函数说明

#### 8.38.2.1 Interface\_()

```
void cyberdog_visual_programming_terminal::Interface::Visual::Interface_ (
    const std::string &fun ) [inline]
```

### 8.38.3 类成员变量说明

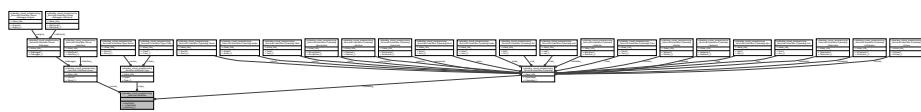
### 8.38.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Visual::show_info_
```

## 8.39 cyberdog\_visual\_programming\_terminal::Interface类 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface 的协作图:



类

- struct [Cyberdog](#)
- struct [Type](#)
- struct [Visual](#)

### Public 成员函数

- [Interface \(\)](#)
- [~Interface \(\)](#)
- void [Interface\\_](#)(const std::string \_fun)

### Public 属性

- struct [cyberdog\\_visual\\_programming\\_terminal::Interface::Cyberdog](#) [cyberdog\\_](#)
- struct [cyberdog\\_visual\\_programming\\_terminal::Interface::Visual](#) [visual\\_](#)
- struct [cyberdog\\_visual\\_programming\\_terminal::Interface::Type](#) [type\\_](#)

### 8.39.1 构造及析构函数说明

#### 8.39.1.1 Interface()

```
cyberdog_visual_programming_terminal::Interface::Interface ( )
```

### 8.39.1.2 ~Interface()

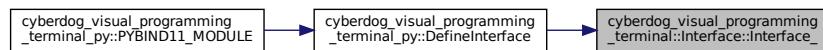
```
cyberdog_visual_programming_terminal::Interface::~Interface ( )
```

## 8.39.2 成员函数说明

### 8.39.2.1 Interface\_()

```
void cyberdog_visual_programming_terminal::Interface::Interface_ (
    const std::string &fun ) [inline]
```

这是这个函数的调用关系图:



## 8.39.3 类成员变量说明

### 8.39.3.1 cyberdog\_

```
struct cyberdog_visual_programming_terminal::Interface::Cyberdog cyberdog_visual_programming_terminal::Interface::cyberdog_
```

### 8.39.3.2 type\_

```
struct cyberdog_visual_programming_terminal::Interface::Type cyberdog_visual_programming_terminal::Interface::type_
```

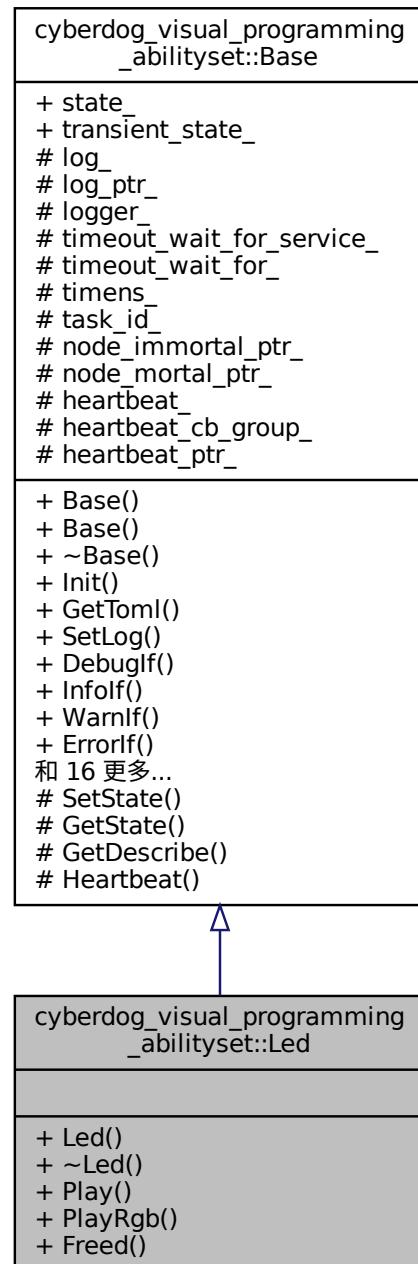
### 8.39.3.3 visual\_

```
struct cyberdog_visual_programming_terminal::Interface::Visual cyberdog_visual_programming_terminal::Interface::visual_
```

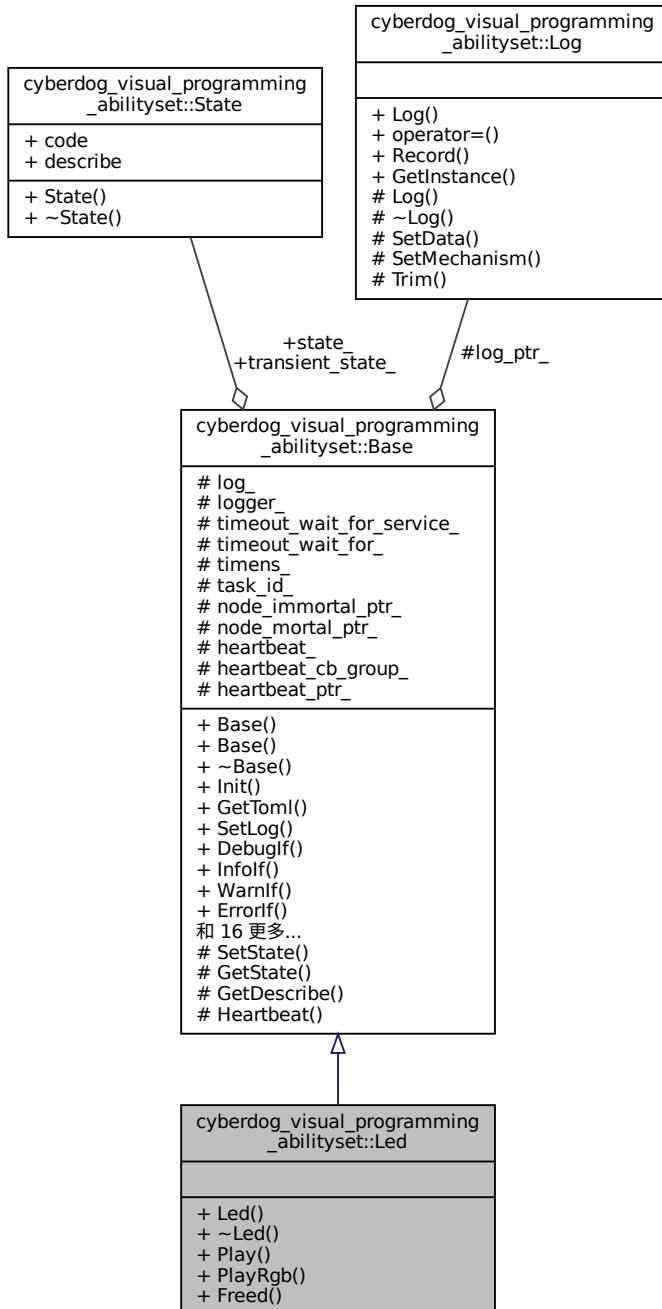
## 8.40 cyberdog\_visual\_programming\_abilityset::Led类 参考

```
#include <led.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Led 继承关系图:



cyberdog\_visual\_programming\_abilityset::Led 的协作图:



## Public 成员函数

- [Led \(\)](#)
- [~Led \(\)](#)
- [LedServiceResponse Play \(const int &\\_target=SrvLedExecute::Request::HEAD\\_LED, const int &\\_effect=SrvLedExecute::Request::RED\\_ON\)](#)
- [LedServiceResponse PlayRgb \(const int &, const int &, const int &, const int &, const int &\)](#)
- [LedServiceResponse Freed \(const int &\)](#)

额外继承的成员函数

### 8.40.1 构造及析构函数说明

#### 8.40.1.1 Led()

```
cyberdog_visual_programming_abilityset::Led::Led ( ) [inline]
```

#### 8.40.1.2 ~Led()

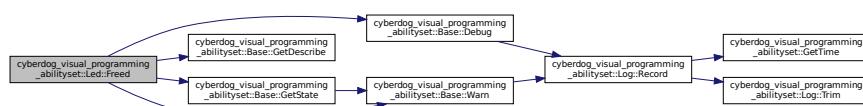
```
cyberdog_visual_programming_abilityset::Led::~Led ( ) [inline]
```

### 8.40.2 成员函数说明

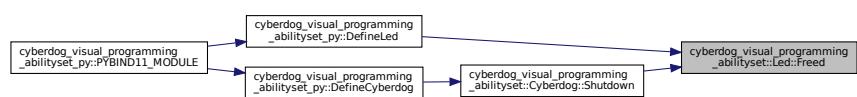
#### 8.40.2.1 Freed()

```
LedServiceResponse cyberdog_visual_programming_abilityset::Led::Freed (
    const int & _target )
```

释放灯控制权 函数调用图:



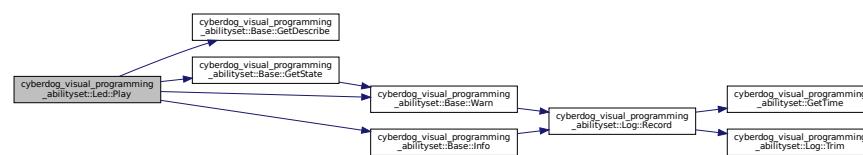
这是这个函数的调用关系图:



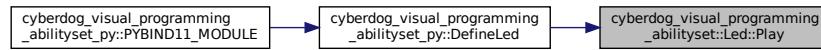
### 8.40.2.2 Play()

```
LedServiceResponse cyberdog_visual_programming_abilityset::Led::Play (
    const int & _target = SrvLedExecute::Request::HEAD_LED,
    const int & _effect = SrvLedExecute::Request::RED_ON )
```

设置灯效 函数调用图:



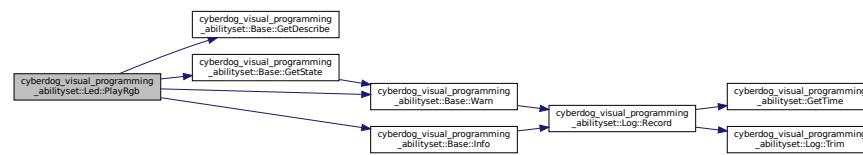
这是这个函数的调用关系图:



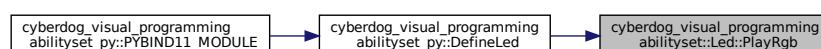
### 8.40.2.3 PlayRgb()

```
LedServiceResponse cyberdog_visual_programming_abilityset::Led::PlayRgb (
    const int & _target,
    const int & _effect,
    const int & _r,
    const int & _g,
    const int & _b )
```

设置灯效 函数调用图:



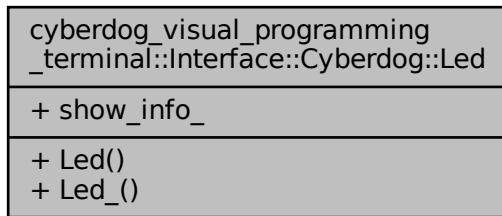
这是这个函数的调用关系图:



## 8.41 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Led 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Led 的协作图:



### Public 成员函数

- `Led()`
- `void Led_(const std::string &fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

#### 8.41.1 构造及析构函数说明

##### 8.41.1.1 Led()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Led::Led ( ) [inline]
```

#### 8.41.2 成员函数说明

##### 8.41.2.1 Led\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Led::Led_ (
    const std::string &fun ) [inline]
```

### 8.41.3 类成员变量说明

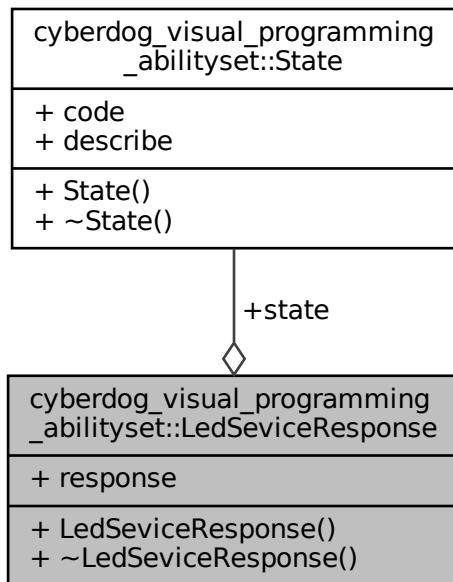
#### 8.41.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Led::show_info_
```

## 8.42 cyberdog\_visual\_programming\_abilityset::LedSeviceResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::LedSeviceResponse 的协作图:



### Public 成员函数

- `LedSeviceResponse ()`
- `~LedSeviceResponse ()`

### Public 属性

- `State state`
- `SrvLedExecute::Response response`

### 8.42.1 详细描述

led服务反馈

### 8.42.2 构造及析构函数说明

#### 8.42.2.1 LedSeviceResponse()

```
cyberdog_visual_programming_abilityset::LedSeviceResponse::LedSeviceResponse () [inline]
```

#### 8.42.2.2 ~LedSeviceResponse()

```
cyberdog_visual_programming_abilityset::LedSeviceResponse::~LedSeviceResponse () [inline]
```

### 8.42.3 类成员变量说明

#### 8.42.3.1 response

```
SrvLedExecute::Response cyberdog_visual_programming_abilityset::LedSeviceResponse::response
```

反馈

#### 8.42.3.2 state

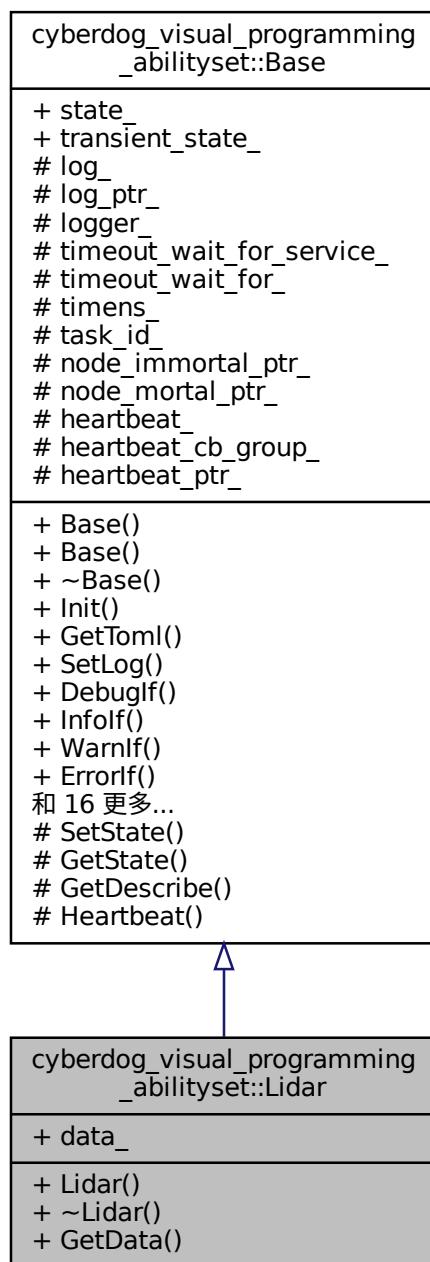
```
State cyberdog_visual_programming_abilityset::LedSeviceResponse::state
```

状态

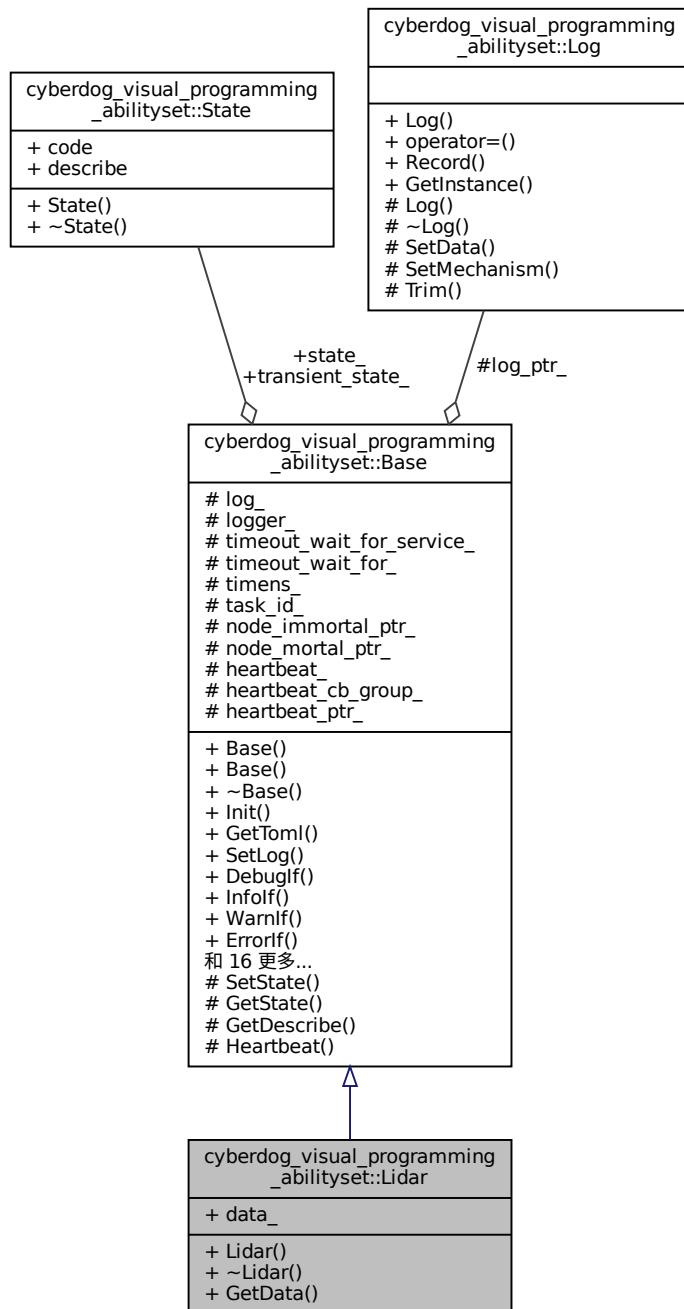
## 8.43 cyberdog\_visual\_programming\_abilityset::Lidar类参考

```
#include <lidar.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Lidar 继承关系图:



cyberdog\_visual\_programming\_abilityset::Lidar 的协作图:



## Public 成员函数

- [Lidar \(\)](#)
- [~Lidar \(\)](#)
- [MsgLaserScan GetData \(const int \\_timeout=5\)](#)

## Public 属性

- `MsgLaserScan data_`

额外继承的成员函数

### 8.43.1 构造及析构函数说明

#### 8.43.1.1 Lidar()

```
cyberdog_visual_programming_abilityset::Lidar::Lidar ( ) [inline]
```

#### 8.43.1.2 ~Lidar()

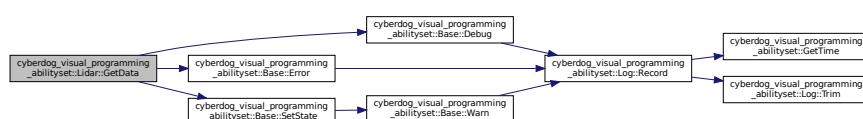
```
cyberdog_visual_programming_abilityset::Lidar::~Lidar ( ) [inline]
```

### 8.43.2 成员函数说明

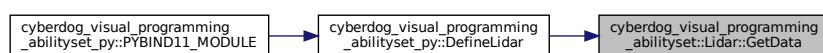
#### 8.43.2.1 GetData()

```
MsgLaserScan cyberdog_visual_programming_abilityset::Lidar::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.43.3 类成员变量说明

#### 8.43.3.1 data\_

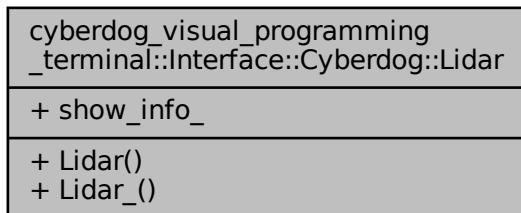
MsgLaserScan cyberdog\_visual\_programming\_abilityset::Lidar::data\_

数据

## 8.44 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Lidar 结构体 参考

#include <interface.hpp>

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Lidar 的协作图:



### Public 成员函数

- [Lidar \(\)](#)
- void [Lidar\\_ \(const std::string \\_fun\)](#)

### Public 属性

- std::function< void(std::string)> [show\\_info\\_](#)

### 8.44.1 构造及析构函数说明

#### 8.44.1.1 Lidar()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar::Lidar() [inline]
```

### 8.44.2 成员函数说明

#### 8.44.2.1 Lidar\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar::Lidar_(
    const std::string &fun) [inline]
```

### 8.44.3 类成员变量说明

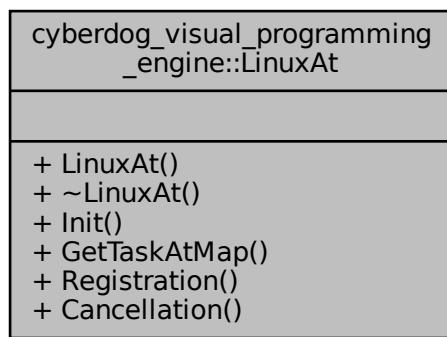
#### 8.44.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_
```

## 8.45 cyberdog\_visual\_programming\_engine::LinuxAt类 参考

```
#include <linux_at.hpp>
```

cyberdog\_visual\_programming\_engine::LinuxAt 的协作图:



## Public 成员函数

- `LinuxAt ()`
- `~LinuxAt ()`
- `bool Init ()`
- `bool GetTaskAtMap (TaskAtMap &)`
- `bool Registration (const std::string &, const std::string &)`
- `bool Cancellation (const std::string &)`

### 8.45.1 构造及析构函数说明

#### 8.45.1.1 LinuxAt()

```
cyberdog_visual_programming_engine::LinuxAt::LinuxAt ( )
```

#### 8.45.1.2 ~LinuxAt()

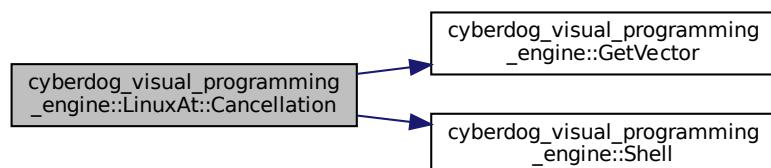
```
cyberdog_visual_programming_engine::LinuxAt::~LinuxAt ( ) [inline]
```

## 8.45.2 成员函数说明

#### 8.45.2.1 Cancellation()

```
bool cyberdog_visual_programming_engine::LinuxAt::Cancellation (
    const std::string & _id )
```

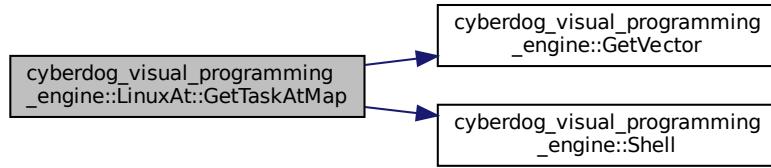
注销 函数调用图:



#### 8.45.2.2 GetTaskAtMap()

```
bool cyberdog_visual_programming_engine::LinuxAt::GetTaskAtMap ( TaskAtMap & _task_at )
```

获取任务列表 函数调用图:



#### 8.45.2.3 Init()

```
bool cyberdog_visual_programming_engine::LinuxAt::Init ( )
```

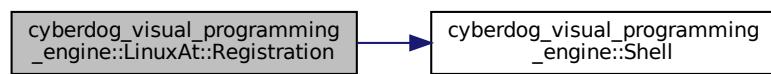
初始化 函数调用图:



#### 8.45.2.4 Registration()

```
bool cyberdog_visual_programming_engine::LinuxAt::Registration ( const std::string & _condition, const std::string & _file )
```

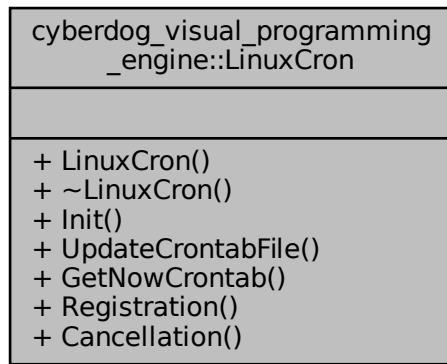
注册 函数调用图:



## 8.46 cyberdog\_visual\_programming\_engine::LinuxCron类 参考

```
#include <linux_cron.hpp>
```

cyberdog\_visual\_programming\_engine::LinuxCron 的协作图:



### Public 成员函数

- [LinuxCron \(\)](#)
- [~LinuxCron \(\)](#)
- [bool Init \(\)](#)
- [bool UpdateCrontabFile \(const std::vector< std::string > &, const bool \\_create=false\)](#)
- [bool GetNowCrontab \(std::vector< std::string > &\)](#)
- [bool Registration \(const std::string &, const std::string &\)](#)
- [bool Cancellation \(const std::string &\)](#)

#### 8.46.1 构造及析构函数说明

##### 8.46.1.1 LinuxCron()

```
cyberdog_visual_programming_engine::LinuxCron::LinuxCron ( )
```

##### 8.46.1.2 ~LinuxCron()

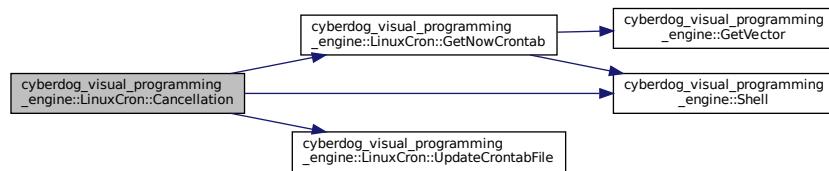
```
cyberdog_visual_programming_engine::LinuxCron::~LinuxCron ( ) [inline]
```

## 8.46.2 成员函数说明

### 8.46.2.1 Cancellation()

```
bool cyberdog_visual_programming_engine::LinuxCron::Cancellation (
    const std::string & _id )
```

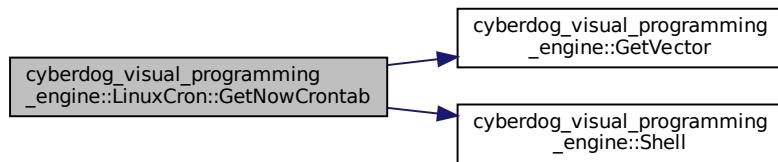
注销 函数调用图:



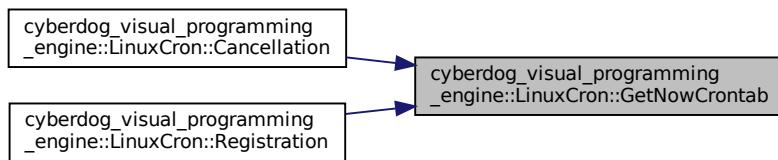
### 8.46.2.2 GetNowCrontab()

```
bool cyberdog_visual_programming_engine::LinuxCron::GetNowCrontab (
    std::vector< std::string > & _now_crontab )
```

获取当前 Crontab 任务列表 函数调用图:



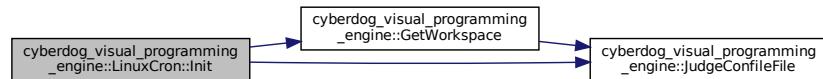
这是这个函数的调用关系图:



### 8.46.2.3 Init()

```
bool cyberdog_visual_programming_engine::LinuxCron::Init ( )
```

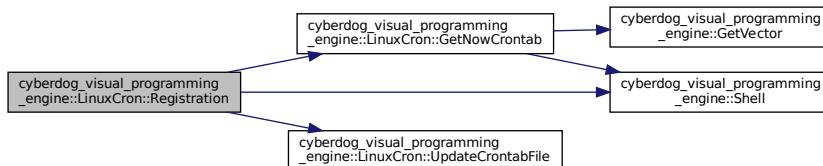
初始化 函数调用图:



### 8.46.2.4 Registration()

```
bool cyberdog_visual_programming_engine::LinuxCron::Registration (
    const std::string & _condition,
    const std::string & _file )
```

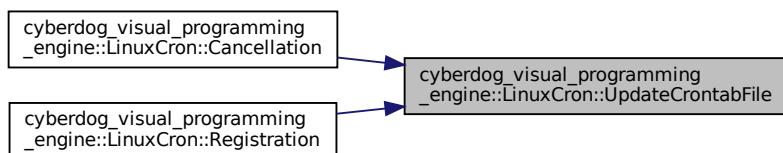
注册 函数调用图:



### 8.46.2.5 UpdateCrontabFile()

```
bool cyberdog_visual_programming_engine::LinuxCron::UpdateCrontabFile (
    const std::vector< std::string > & _new_crontab,
    const bool _create = false )
```

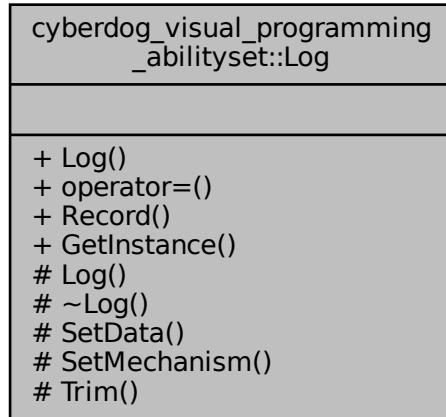
更新 Crontab 文件 这是这个函数的调用关系图:



## 8.47 cyberdog\_visual\_programming\_abilityset::Log类 参考

```
#include <log.hpp>
```

cyberdog\_visual\_programming\_abilityset::Log 的协作图:



### Public 类型

- enum Rank {
 debug = 0, info, warn, error,
 fatal
 }

### Public 成员函数

- Log (Log &other)=delete
- void operator= (const Log &)=delete
- template<typename ... Args>  
void Record (Rank rank, bool condition, std::string role, const char \*format, const Args &... args)

### 静态 Public 成员函数

- static Log \* GetInstance (rclcpp::Node::SharedPtr \_node\_ptr, const std::string &\_task\_id, const toml::value &\_params\_toml)

### Protected 成员函数

- Log (rclcpp::Node::SharedPtr \_node\_ptr, const std::string \_task\_id, const toml::value &\_params\_toml)
- ~Log ()
- bool SetData (const toml::value &)
- bool SetMechanism (const toml::value &)
- char \* Trim (char \*str)

## 8.47.1 成员枚举类型说明

### 8.47.1.1 Rank

```
enum cyberdog_visual_programming_abilityset::Log::Rank
```

日志级别

枚举值

debug	调试
info	信息
warn	警告
error	错误
fatal	致命

## 8.47.2 构造及析构函数说明

### 8.47.2.1 Log() [1/2]

```
cyberdog_visual_programming_abilityset::Log::Log (
    Log & other ) [delete]
```

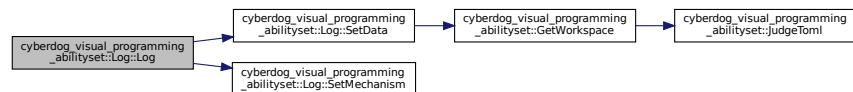
这是这个函数的调用关系图:



### 8.47.2.2 Log() [2/2]

```
cyberdog_visual_programming_abilityset::Log::Log (
    rclcpp::Node::SharedPtr node_ptr,
    const std::string task_id,
    const toml::value & params_toml ) [inline], [protected]
```

函数调用图:



#### 8.47.2.3 ~Log()

`cyberdog_visual_programming_abilityset::Log::~Log () [inline], [protected]`

### 8.47.3 成员函数说明

#### 8.47.3.1 GetInstance()

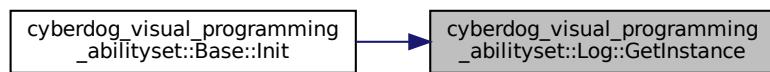
```

Log * cyberdog_visual_programming_abilityset::Log::GetInstance (
    rclcpp::Node::SharedPtr _node_ptr,
    const std::string & _task_id,
    const toml::value & _params_toml ) [static]
  
```

获取接口 函数调用图:



这是这个函数的调用关系图:



### 8.47.3.2 operator=( )

```
void cyberdog_visual_programming_abilityset::Log::operator= (
    const Log & ) [delete]
```

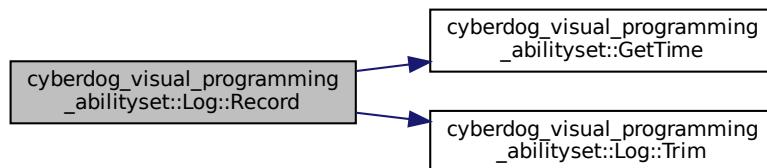
### 8.47.3.3 Record()

```
template<typename ... Args>
void cyberdog_visual_programming_abilityset::Log::Record (
    Rank rank,
    bool condition,
    std::string role,
    const char * format,
    const Args &... args ) [inline]
```

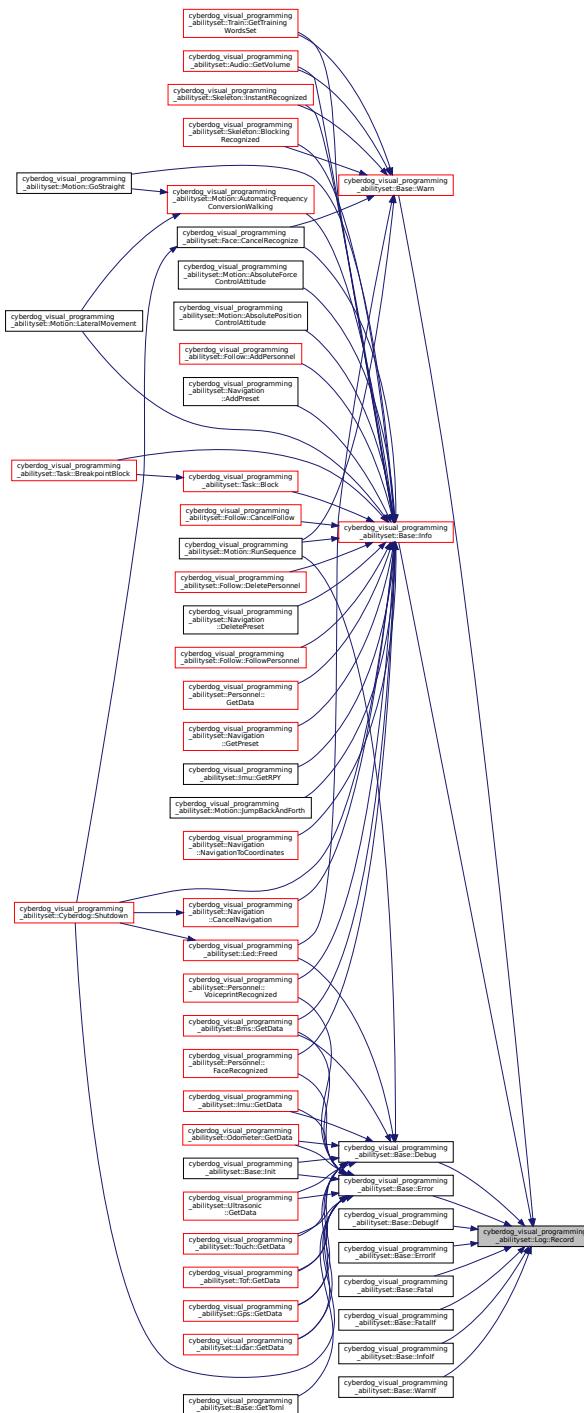
参数

args	记录日志
------	------

函数调用图:



这是这个函数的调用关系图:



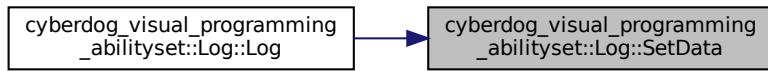
#### 8.47.3.4 SetData()

```
bool cyberdog_visual_programming_abilityset::Log::SetData (
    const toml::value & _params_toml ) [protected]
```

设置数据 函数调用图:



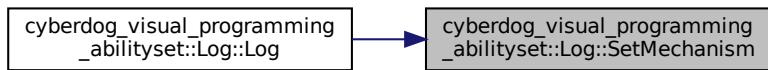
这是这个函数的调用关系图:



#### 8.47.3.5 SetMechanism()

```
bool cyberdog_visual_programming_abilityset::Log::SetMechanism ( const toml::value & _params_toml ) [protected]
```

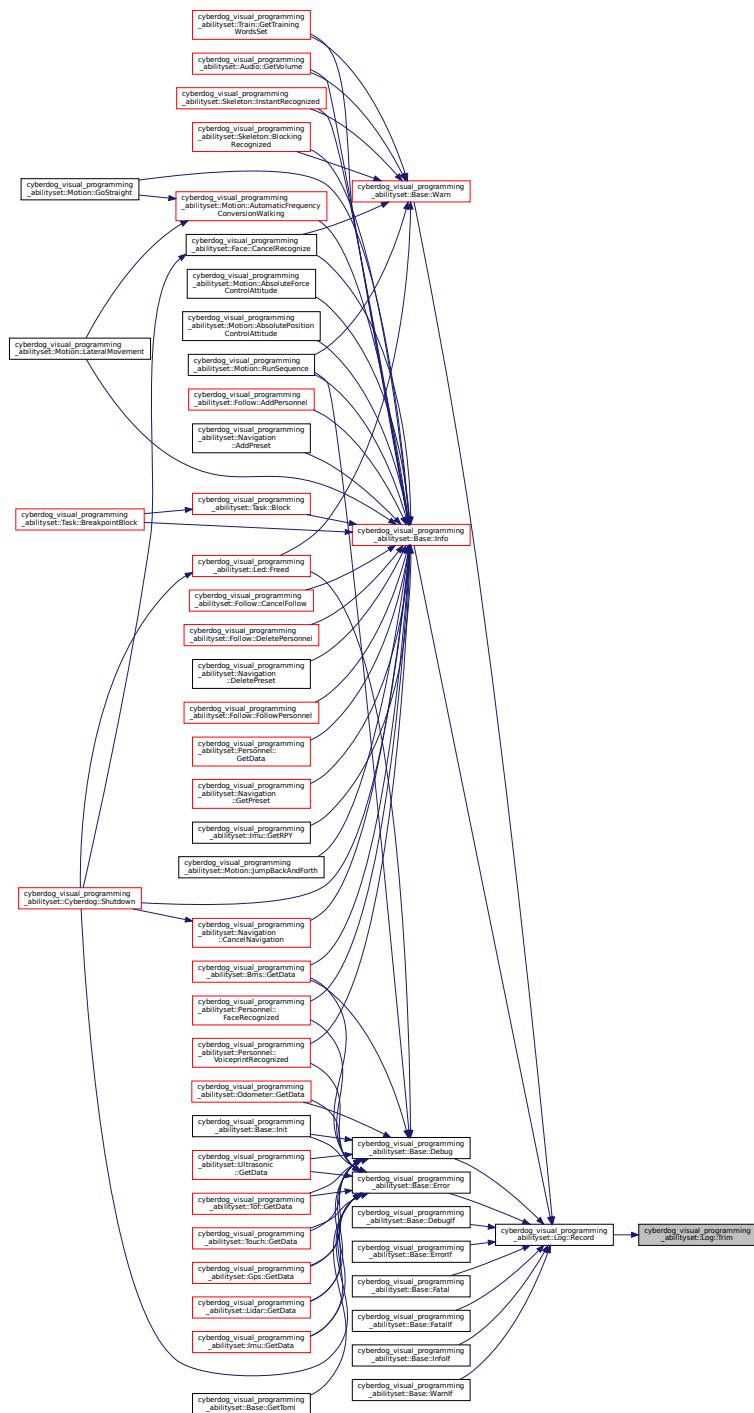
设置机制 这是这个函数的调用关系图:



#### 8.47.3.6 Trim()

```
char * cyberdog_visual_programming_abilityset::Log::Trim ( char * str ) [protected]
```

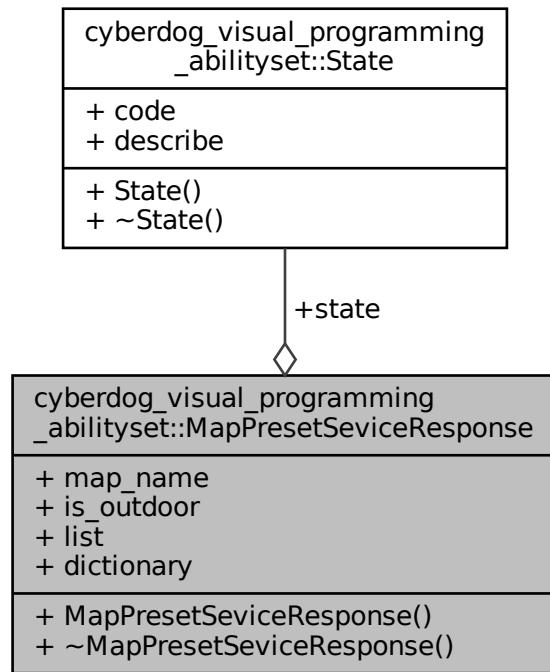
裁剪字符串收尾空白字符 这是这个函数的调用关系图:



## 8.48 cyberdog\_visual\_programming\_abilityset::MapPresetSevice<-Response类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::MapPresetSeviceResponse 的协作图:



## Public 成员函数

- `MapPresetSeviceResponse ()`
- `~MapPresetSeviceResponse ()`

## Public 属性

- `State state`
- `std::string map_name`
- `bool is_outdoor`
- `std::vector< MsgPreset > list`
- `std::map< std::string, MsgPreset > dictionary`

### 8.48.1 详细描述

预置点信息

### 8.48.2 构造及析构函数说明

### 8.48.2.1 MapPresetSeviceResponse()

```
cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::MapPresetSeviceResponse ( )
[inline]
```

### 8.48.2.2 ~MapPresetSeviceResponse()

```
cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::~MapPresetSeviceResponse ( )
[inline]
```

## 8.48.3 类成员变量说明

### 8.48.3.1 dictionary

```
std::map<std::string, MsgPreset> cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::dictionary
```

字典

### 8.48.3.2 is\_outdoor

```
bool cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::is_outdoor
```

是否为室外地图

### 8.48.3.3 list

```
std::vector<MsgPreset> cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::list
```

列表

### 8.48.3.4 map\_name

```
std::string cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::map_name
```

地图名

### 8.48.3.5 state

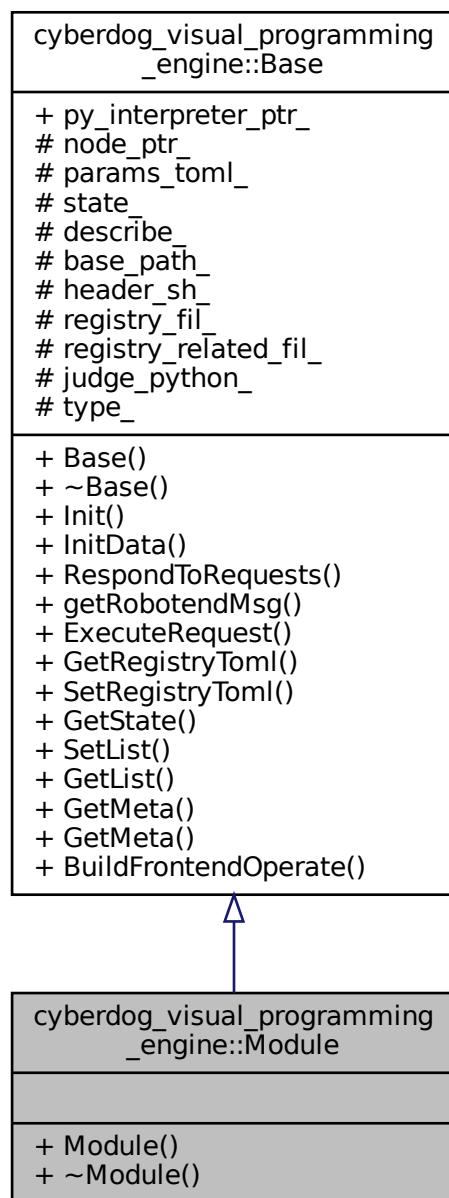
```
State cyberdog_visual_programming_abilityset::MapPresetSeviceResponse::state
```

状态

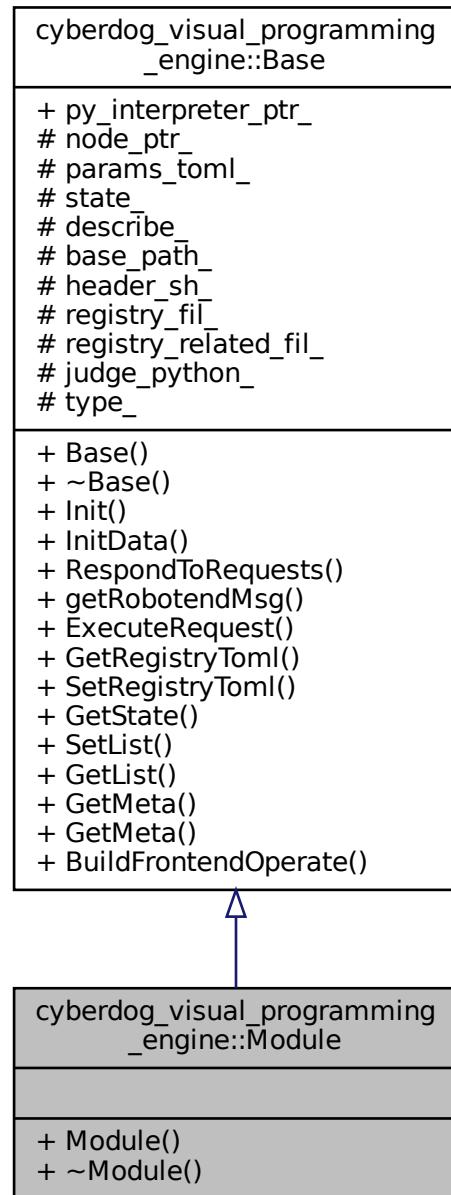
## 8.49 cyberdog\_visual\_programming\_engine::Module类 参考

```
#include <module.hpp>
```

类 cyberdog\_visual\_programming\_engine::Module 继承关系图:



cyberdog\_visual\_programming\_engine::Module 的协作图:



## Public 成员函数

- [Module \(\)](#)
- [~Module \(\)](#)

额外继承的成员函数

### 8.49.1 构造及析构函数说明

#### 8.49.1.1 Module()

```
cyberdog_visual_programming_engine::Module::Module ( )
```

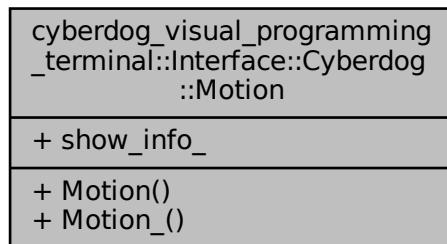
#### 8.49.1.2 ~Module()

```
cyberdog_visual_programming_engine::Module::~Module ( )
```

## 8.50 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Motion 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Motion 的协作图:



### Public 成员函数

- `Motion ()`
- `void Motion_(const std::string &fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

## 8.50.1 构造及析构函数说明

### 8.50.1.1 Motion()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion::Motion () [inline]
```

## 8.50.2 成员函数说明

### 8.50.2.1 Motion\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion::Motion_ (
    const std::string &fun ) [inline]
```

## 8.50.3 类成员变量说明

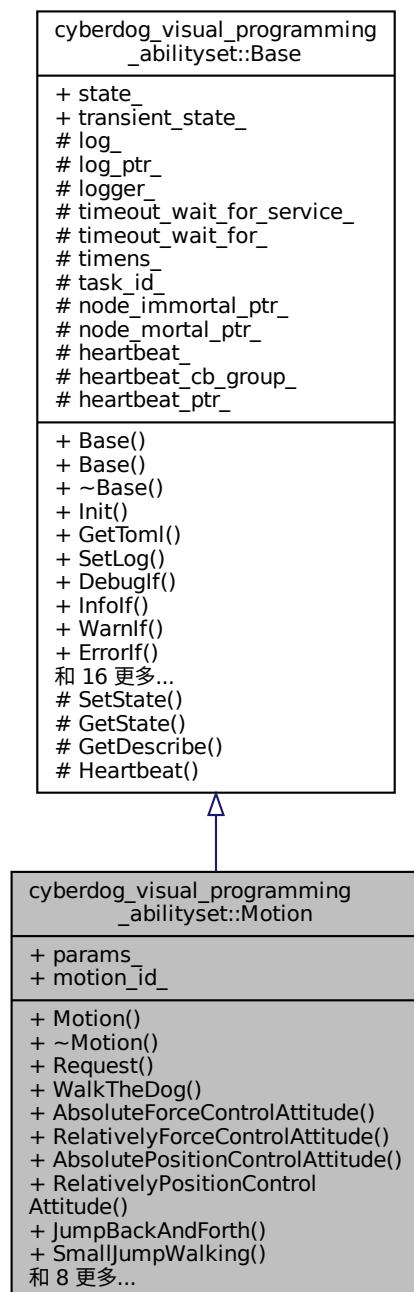
### 8.50.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Motion::show_info_
```

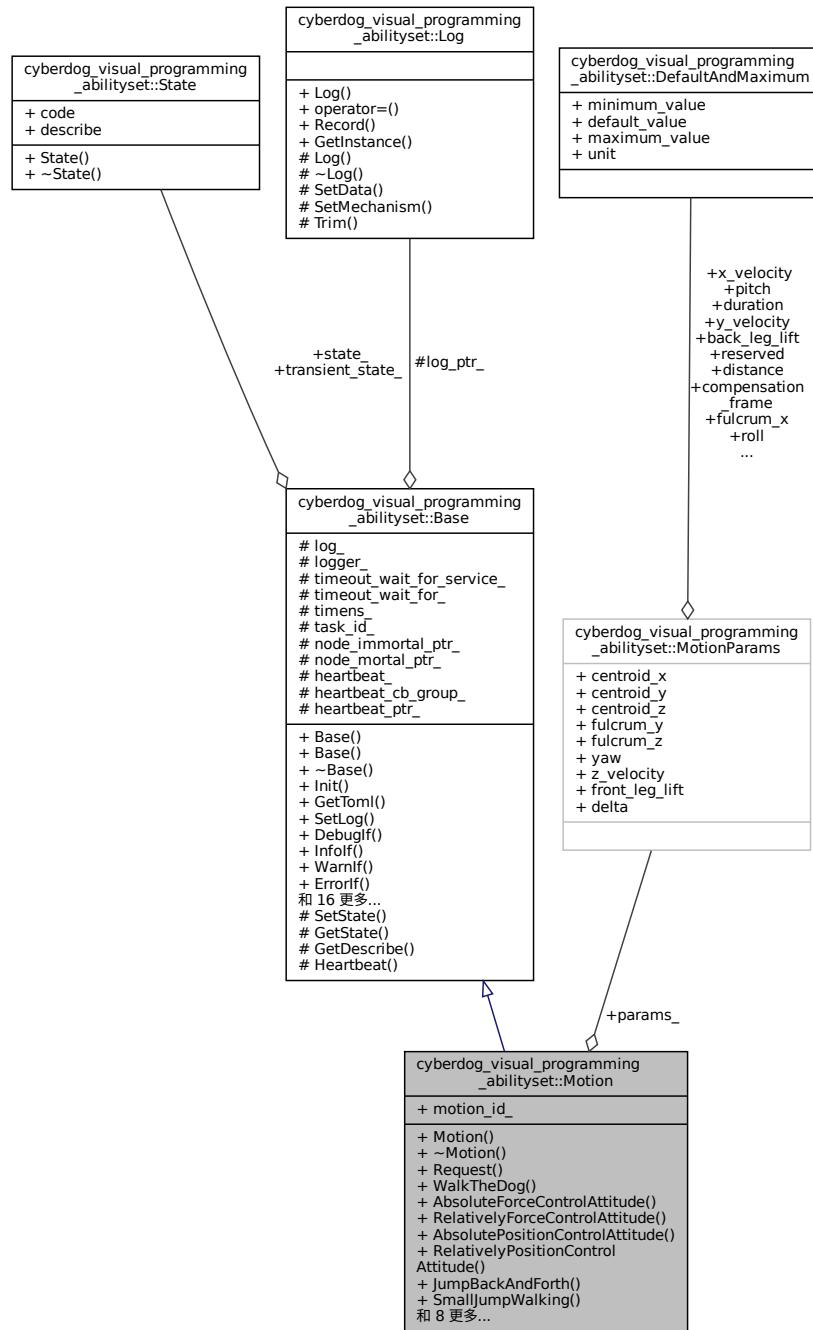
## 8.51 cyberdog\_visual\_programming\_abilityset::Motion类 参考

```
#include <motion.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Motion 继承关系图:



cyberdog\_visual\_programming\_abilityset::Motion 的协作图:



## Public 成员函数

- `Motion ()`
- `~Motion ()`
- `MotionResultServiceResponse Request (const int32_t motion_id=0, const int32_t duration=0)`
- `MotionResultServiceResponse WalkTheDog (const double, const double)`
- `MotionResultServiceResponse AbsoluteForceControlAttitude (const double, const double, const double, const double, const double)`

- `MotionResultServiceResponse RelativelyForceControlAttitude` (const double, const double, const double, const double, const double, const double)
- `MotionResultServiceResponse AbsolutePositionControlAttitude` (const double, const double)
- `MotionResultServiceResponse RelativelyPositionControlAttitude` (const double, const double)
- `MotionServoCmdResponse JumpBackAndForth` (const double, const double, const double, const double, const double, const double \_distance=0, const double \_duration=0, const uint \_compensation\_frame\_size=0)
- `MotionServoCmdResponse SmallJumpWalking` (const double, const double, const double, const double, const double, const double, const double \_distance=0, const double \_duration=0, const uint \_compensation\_frame\_size=0)
- `MotionServoCmdResponse AutomaticFrequencyConversionWalking` (const double, const double, const double, const double, const double, const double, const double \_distance=0, const double \_duration=0, const uint \_compensation\_frame\_size=0)
- `MotionServoCmdResponse TrotWalking` (const double, const double, const double, const double, const double, const double \_distance=0, const double \_duration=0, const uint \_compensation\_frame\_size=0)
- `MotionServoCmdResponse RunFastWalking` (const double, const double, const double, const double, const double, const double \_distance=0, const double \_duration=0, const uint \_compensation\_frame\_size=0)
- `MotionServoCmdResponse Turn` (const double, double)
- `MotionServoCmdResponse GoStraight` (const double, const double, const double)
- `MotionServoCmdResponse LateralMovement` (const double, const double, const double)
- `MotionSequenceServiceResponse RunSequence` (const `MotionSequence` &)
- void `InitDependent` (const std::function< `AudioPlaySeviceResponse`(const std::string, const int8\_t)> &, const std::function< `AudioPlaySeviceResponse`(const uint16\_t, const int8\_t)> &, const std::function< bool(`MsgOdometry` &, double &, const double)> &, const std::function< bool(`Processor`)> &)

## Public 属性

- MotionParams `params_`
- int `motion_id_`

额外继承的成员函数

### 8.51.1 构造及析构函数说明

#### 8.51.1.1 Motion()

```
cyberdog_visual_programming_abilityset::Motion::Motion () [inline]
```

#### 8.51.1.2 ~Motion()

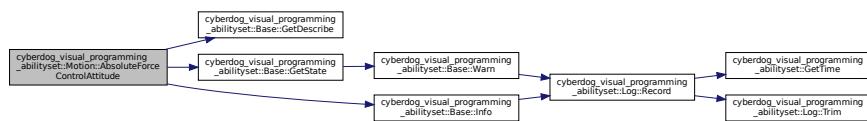
```
cyberdog_visual_programming_abilityset::Motion::~Motion () [inline]
```

### 8.51.2 成员函数说明

### 8.51.2.1 AbsoluteForceControlAttitude()

```
MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::AbsoluteForce←
ControlAttitude (
    const double _centroidZ,
    const double _roll,
    const double _pitch,
    const double _yaw,
    const double _duration )
```

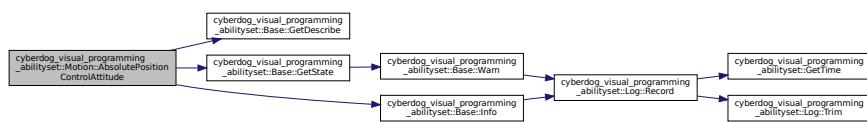
[结果指令]绝对力控姿态 函数调用图:



### 8.51.2.2 AbsolutePositionControlAttitude()

```
MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::AbsolutePosition←
ControlAttitude (
    const double _centroidZ,
    const double _duration )
```

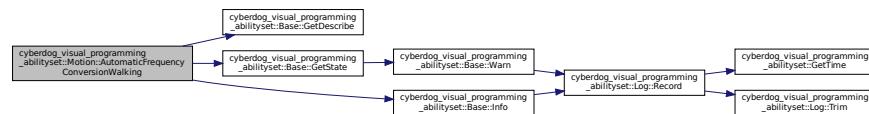
[结果指令]绝对位控姿态 函数调用图:



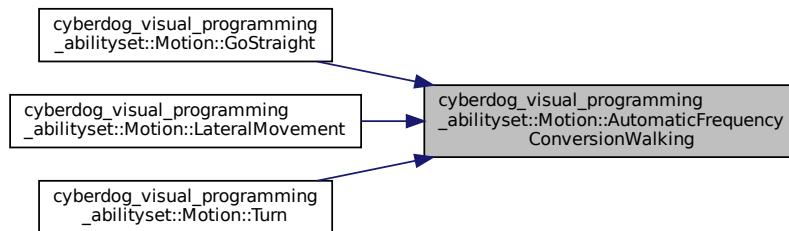
### 8.51.2.3 AutomaticFrequencyConversionWalking()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::AutomaticFrequency←
ConversionWalking (
    const double _x_velocity,
    const double _y_velocity,
    const double _z_velocity,
    const double _front_leg_lift,
    const double _back_leg_lift,
    const double _distance = 0,
    const double _duration = 0,
    const uint _compensation_frame_size = 0 )
```

[伺服指令]自动变频行走 函数调用图:



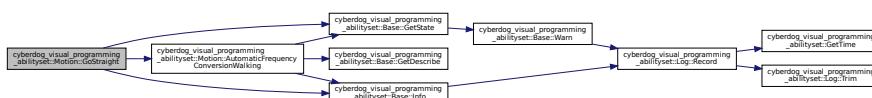
这是这个函数的调用关系图:



#### 8.51.2.4 GoStraight()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::GoStraight (
    const double _x_velocity,
    const double _distance,
    const double _duration )
```

直行 函数调用图:



#### 8.51.2.5 InitDependent()

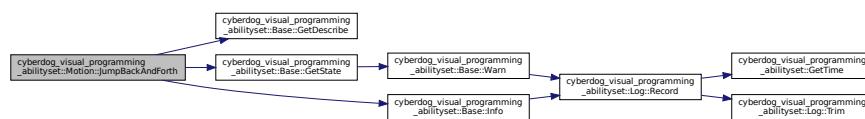
```
void cyberdog_visual_programming_abilityset::Motion::InitDependent (
    const std::function< AudioPlayServiceResponse(const std::string, const int8_t)> &_FOnlinePlay,
    const std::function< AudioPlayServiceResponse(const uint16_t, const int8_t)> &_FOfflinePlay,
    const std::function< bool(MsgOdometry &, double &, const double)> &_OdomIfCumulativeDistance,
    const std::function< bool(Processor)> &_FGetProcessor )
```

初始化依赖

### 8.51.2.6 JumpBackAndForth()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::JumpBackAndForth (
    const double _x_velocity,
    const double _y_velocity,
    const double _z_velocity,
    const double _front_leg_lift,
    const double _back_leg_lift,
    const double _distance = 0,
    const double _duration = 0,
    const uint _compensation_frame_size = 0 )
```

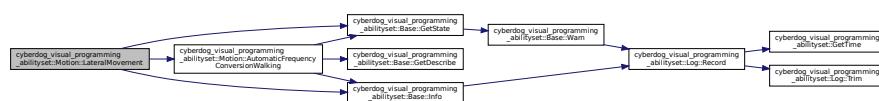
[伺服指令]前后跳 函数调用图:



### 8.51.2.7 LateralMovement()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::LateralMovement (
    const double y_velocity,
    const double _distance,
    const double _duration )
```

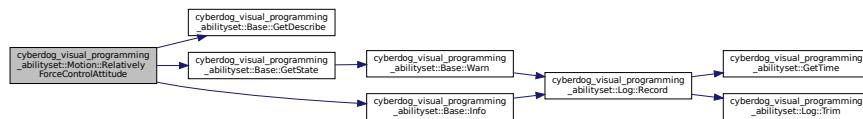
横移 函数调用图:



### 8.51.2.8 RelativelyForceControlAttitude()

```
MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::RelativelyForceControlAttitude (
    const double _centroid_x,
    const double _centroid_y,
    const double _centroid_z,
    const double _roll,
    const double _pitch,
    const double _yaw,
    const double _duration )
```

[结果指令]相对力控姿态 函数调用图:

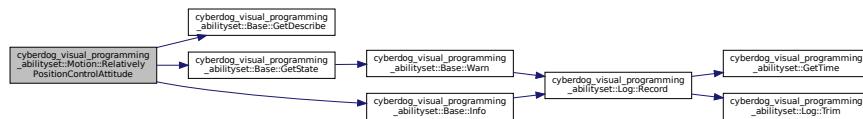


### 8.51.2.9 RelativelyPositionControlAttitude()

```

MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::RelativelyPositionControlAttitude (
    const double _centroid_x,
    const double _centroid_y,
    const double _centroid_z,
    const double _roll,
    const double _pitch,
    const double _yaw,
    const double _fulcrum_x,
    const double _fulcrum_y,
    const double _fulcrum_z,
    const double _duration )
  
```

[结果指令]相对位控姿态 函数调用图:

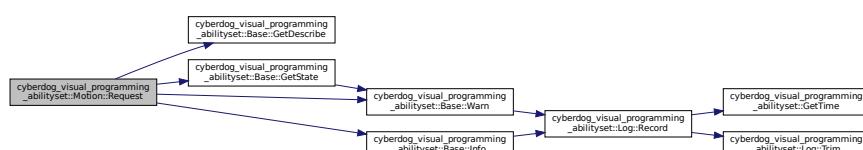


### 8.51.2.10 Request()

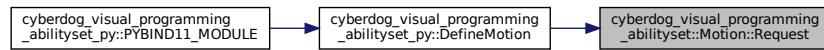
```

MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::Request (
    const int32_t motion_id = 0,
    const int32_t duration = 0 )
  
```

[结果指令]请求接口 函数调用图:



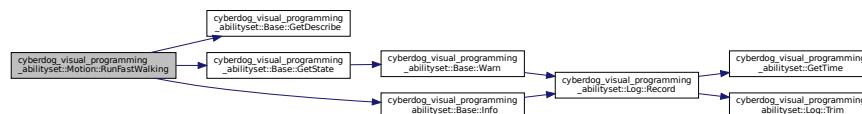
这是这个函数的调用关系图:



### 8.51.2.11 RunFastWalking()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::RunFastWalking (
    const double _x_velocity,
    const double _y_velocity,
    const double _z_velocity,
    const double _front_leg_lift,
    const double _back_leg_lift,
    const double _distance = 0,
    const double _duration = 0,
    const uint _compensation_frame_size = 0 )
```

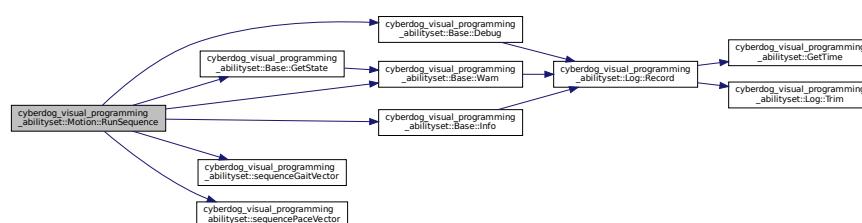
[伺服指令]快跑行走 函数调用图:



### 8.51.2.12 RunSequence()

```
MotionSequenceServiceResponse cyberdog_visual_programming_abilityset::Motion::RunSequence (
    const MotionSequence & _sequence )
```

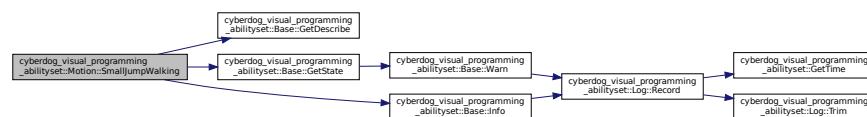
运行序列动作 函数调用图:



### 8.51.2.13 SmallJumpWalking()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::SmallJumpWalking (
    const double _x_velocity,
    const double _y_velocity,
    const double _z_velocity,
    const double _front_leg_lift,
    const double _back_leg_lift,
    const double _distance = 0,
    const double _duration = 0,
    const uint _compensation_frame_size = 0 )
```

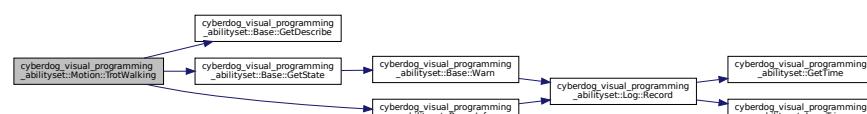
[伺服指令]小跳行走 函数调用图:



### 8.51.2.14 TrotWalking()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::TrotWalking (
    const double _x_velocity,
    const double _y_velocity,
    const double _z_velocity,
    const double _front_leg_lift,
    const double _back_leg_lift,
    const double _distance = 0,
    const double _duration = 0,
    const uint _compensation_frame_size = 0 )
```

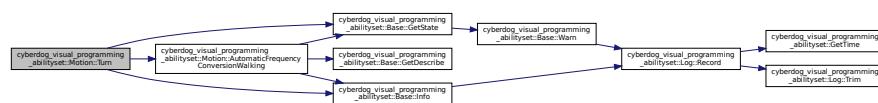
[伺服指令]小跑行走 函数调用图:



### 8.51.2.15 Turn()

```
MotionServoCmdResponse cyberdog_visual_programming_abilityset::Motion::Turn (
    const double _angle,
    double _duration )
```

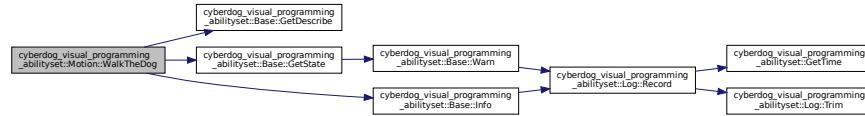
转向 函数调用图:



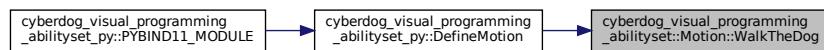
### 8.51.2.16 WalkTheDog()

```
MotionResultServiceResponse cyberdog_visual_programming_abilityset::Motion::WalkTheDog (
    const double _front_leg_lift,
    const double _back_leg_lift )
```

[结果指令]遛狗 函数调用图:



这是这个函数的调用关系图:



## 8.51.3 类成员变量说明

### 8.51.3.1 motion\_id\_

```
int cyberdog_visual_programming_abilityset::Motion::motion_id_
```

运动ID

### 8.51.3.2 params\_

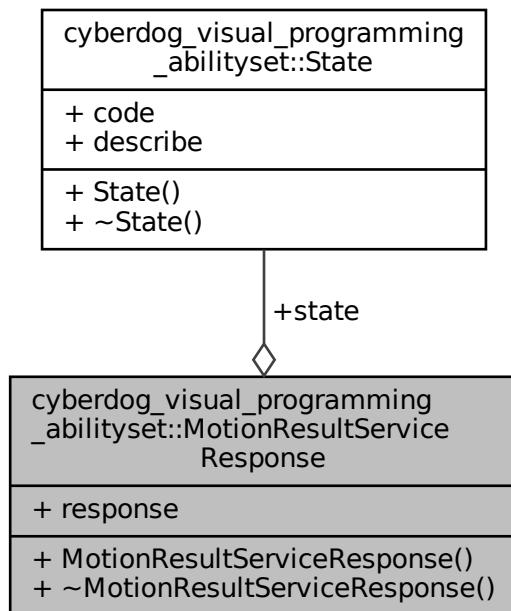
MotionParams cyberdog\_visual\_programming\_abilityset::Motion::params\_

参数

## 8.52 cyberdog\_visual\_programming\_abilityset::MotionResultServiceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::MotionResultServiceResponse 的协作图:



### Public 成员函数

- `MotionResultServiceResponse ()`
- `~MotionResultServiceResponse ()`

### Public 属性

- `State state`
- `SrvMotionResultCmd::Response response`

### 8.52.1 详细描述

运动结果指令服务反馈

### 8.52.2 构造及析构函数说明

#### 8.52.2.1 MotionResultServiceResponse()

```
cyberdog_visual_programming_abilityset::MotionResultServiceResponse::MotionResultServiceResponse
( ) [inline]
```

#### 8.52.2.2 ~MotionResultServiceResponse()

```
cyberdog_visual_programming_abilityset::MotionResultServiceResponse::~MotionResultServiceResponse
( ) [inline]
```

### 8.52.3 类成员变量说明

#### 8.52.3.1 response

```
SrvMotionResultCmd::Response cyberdog_visual_programming_abilityset::MotionResultServiceResponse::
::response
```

反馈

#### 8.52.3.2 state

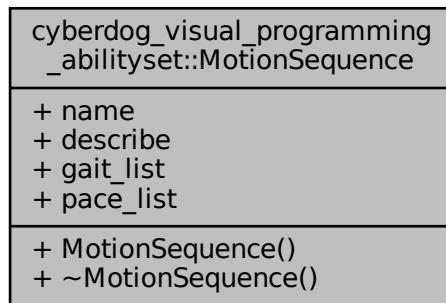
```
State cyberdog_visual_programming_abilityset::MotionResultServiceResponse::state
```

状态

## 8.53 cyberdog\_visual\_programming\_abilityset::MotionSequence类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::MotionSequence 的协作图:



### Public 成员函数

- `MotionSequence ()`
- `~MotionSequence ()`

### Public 属性

- `std::string name`
- `std::string describe`
- `std::vector<MsgMotionSequenceGait> gait_list`
- `std::vector<MsgMotionSequencePace> pace_list`

#### 8.53.1 详细描述

运动序列

#### 8.53.2 构造及析构函数说明

##### 8.53.2.1 MotionSequence()

```
cyberdog_visual_programming_abilityset::MotionSequence::MotionSequence ( ) [inline]
```

### 8.53.2.2 ~MotionSequence()

```
cyberdog_visual_programming_abilityset::MotionSequence::~MotionSequence ( ) [inline]
```

## 8.53.3 类成员变量说明

### 8.53.3.1 describe

```
std::string cyberdog_visual_programming_abilityset::MotionSequence::describe
```

描述

### 8.53.3.2 gait\_list

```
std::vector<MsgMotionSequenceGait> cyberdog_visual_programming_abilityset::MotionSequence::gait_list
```

步态列表

### 8.53.3.3 name

```
std::string cyberdog_visual_programming_abilityset::MotionSequence::name
```

名称

### 8.53.3.4 pace\_list

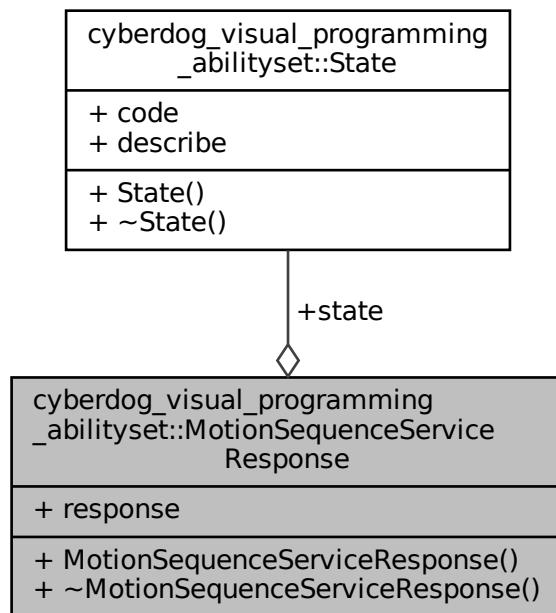
```
std::vector<MsgMotionSequencePace> cyberdog_visual_programming_abilityset::MotionSequence::pace_list
```

步伐列表

## 8.54 cyberdog\_visual\_programming\_abilityset::MotionSequenceServiceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::MotionSequenceServiceResponse 的协作图:



### Public 成员函数

- `MotionSequenceServiceResponse ()`
- `~MotionSequenceServiceResponse ()`

### Public 属性

- `State state`
- `SrvMotionSequenceShow::Response response`

#### 8.54.1 详细描述

运动序列动作服务反馈

## 8.54.2 构造及析构函数说明

### 8.54.2.1 MotionSequenceServiceResponse()

```
cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse::MotionSequenceServiceResponse ( ) [inline]
```

### 8.54.2.2 ~MotionSequenceServiceResponse()

```
cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse::~MotionSequenceServiceResponse ( ) [inline]
```

## 8.54.3 类成员变量说明

### 8.54.3.1 response

```
SrvMotionSequenceShow::Response cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse::response
```

反馈

### 8.54.3.2 state

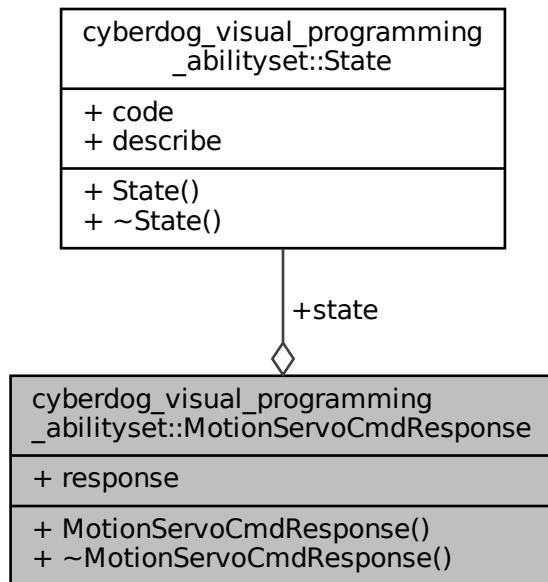
```
State cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse::state
```

状态

## 8.55 cyberdog\_visual\_programming\_abilityset::MotionServoCmdResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::MotionServoCmdResponse 的协作图:



### Public 成员函数

- `MotionServoCmdResponse ()`
- `~MotionServoCmdResponse ()`

### Public 属性

- `State state`
- `MsgMotionServoResponse response`

#### 8.55.1 详细描述

运动伺服反馈

#### 8.55.2 构造及析构函数说明

### 8.55.2.1 MotionServoCmdResponse()

```
cyberdog_visual_programming_abilityset::MotionServoCmdResponse::MotionServoCmdResponse ( )
[inline]
```

### 8.55.2.2 ~MotionServoCmdResponse()

```
cyberdog_visual_programming_abilityset::MotionServoCmdResponse::~MotionServoCmdResponse ( )
[inline]
```

## 8.55.3 类成员变量说明

### 8.55.3.1 response

```
MsgMotionServoResponse cyberdog_visual_programming_abilityset::MotionServoCmdResponse::response
```

反馈

### 8.55.3.2 state

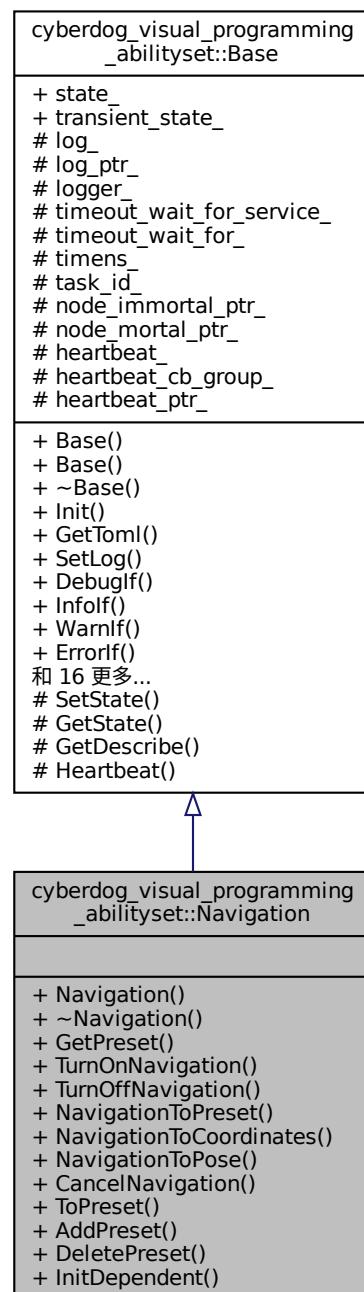
```
State cyberdog_visual_programming_abilityset::MotionServoCmdResponse::state
```

状态

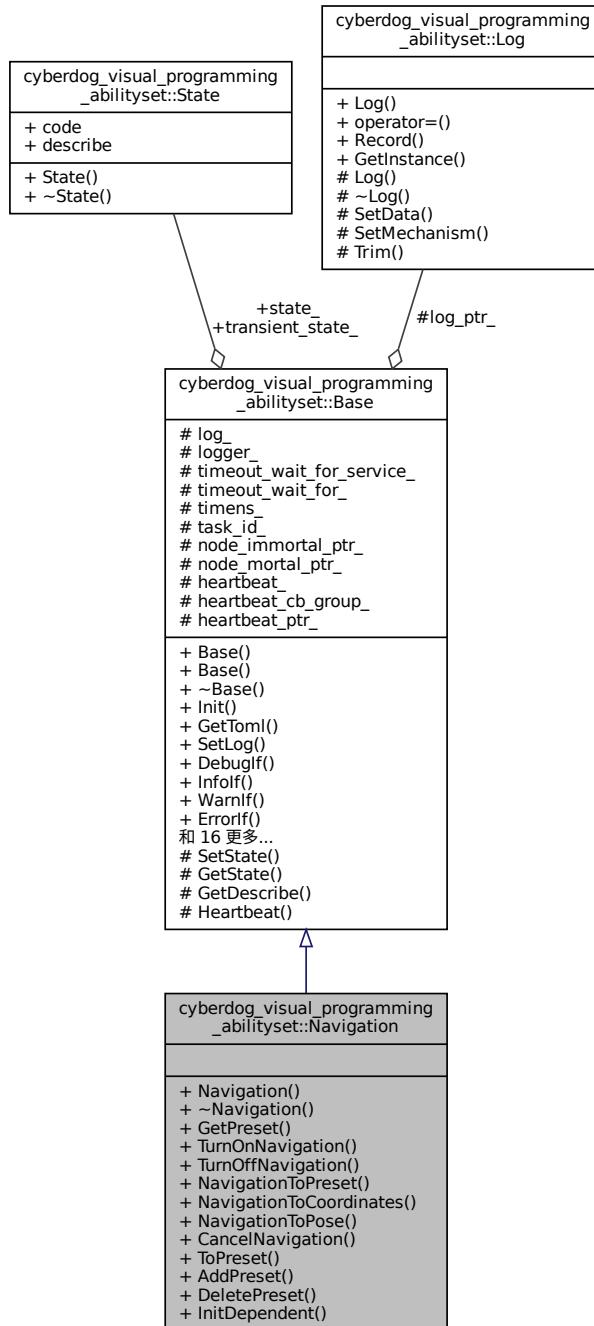
## 8.56 cyberdog\_visual\_programming\_abilityset::Navigation类 参考

```
#include <navigation.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Navigation 继承关系图:



cyberdog\_visual\_programming\_abilityset::Navigation 的协作图:



## Public 成员函数

- `Navigation ()`
- `~Navigation ()`
- `MapPresetSeviceResponse GetPreset (const int _timeout=3)`
- `NavigationActionResponse TurnOnNavigation (const bool outdoor=false, const bool assisted_relocation=false, const bool interact=false, const int volume=50)`

- `NavigationActionResponse TurnOffNavigation ()`
- `NavigationActionResponse NavigationToPreset (const std::string)`
- `NavigationActionResponse NavigationToCoordinates (const bool is_outdoor=false, const double x=0, const double y=0, const double z=0, const double roll=0, const double pitch=0, const double yaw=0)`
- `NavigationActionResponse NavigationToPose (const MsgPose &)`
- `NavigationActionResponse CancelNavigation (const int _timeout=3)`
- `NavigationActionResponse ToPreset (const std::string, const bool assisted_relocation=false, const bool interact=false, const int volume=50)`
- `State AddPreset (const std::string)`
- `State DeletePreset (const std::string)`
- `void InitDependent (const std::function< MotionServoCmdResponse(const double, double)> &, const std::function< AudioPlaySeviceResponse(const std::string, const int8_t)> &, const std::function< State(const std::string, const int8_t)> &)`

额外继承的成员函数

### 8.56.1 构造及析构函数说明

#### 8.56.1.1 Navigation()

```
cyberdog_visual_programming_abilityset::Navigation::Navigation ( ) [inline]
```

#### 8.56.1.2 ~Navigation()

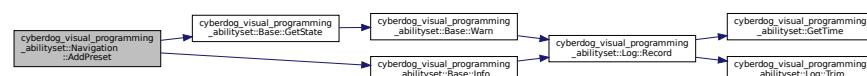
```
cyberdog_visual_programming_abilityset::Navigation::~Navigation ( ) [inline]
```

### 8.56.2 成员函数说明

#### 8.56.2.1 AddPreset()

```
State cyberdog_visual_programming_abilityset::Navigation::AddPreset (
    const std::string _preset_name )
```

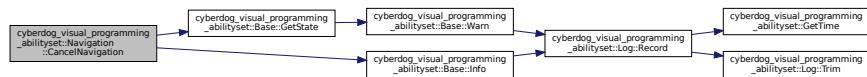
添加预置点 函数调用图:



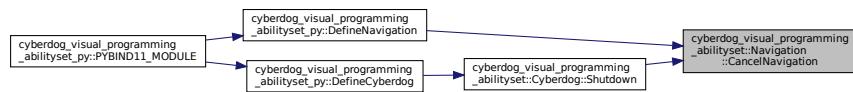
### 8.56.2.2 CancelNavigation()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::CancelNavigation (
    const int _timeout = 3 )
```

取消导航 函数调用图:



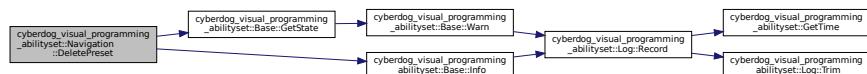
这是这个函数的调用关系图:



### 8.56.2.3 DeletePreset()

```
State cyberdog_visual_programming_abilityset::Navigation::DeletePreset (
    const std::string _preset_name )
```

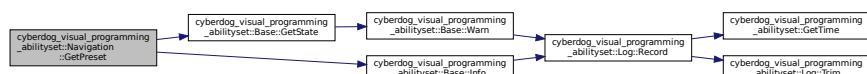
删除预置点 函数调用图:



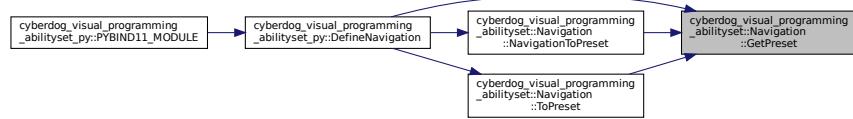
### 8.56.2.4 GetPreset()

```
MapPresetServiceResponse cyberdog_visual_programming_abilityset::Navigation::GetPreset (
    const int _timeout = 3 )
```

获取预置点 函数调用图:



这是这个函数的调用关系图:



### 8.56.2.5 InitDependent()

```

void cyberdog_visual_programming_abilityset::Navigation::InitDependent (
    const std::function< MotionServoCmdResponse(const double, double)> & _Turn,
    const std::function< AudioPlaySeviceResponse(const std::string, const int8_t)> & _Play,
    const std::function< State(const std::string, const int8_t)> & _InstantlyPlay )
  
```

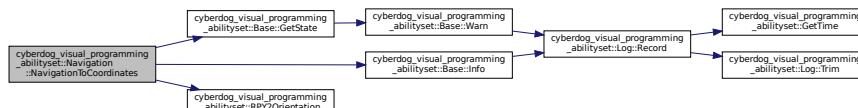
初始化依赖

### 8.56.2.6 NavigationToCoordinates()

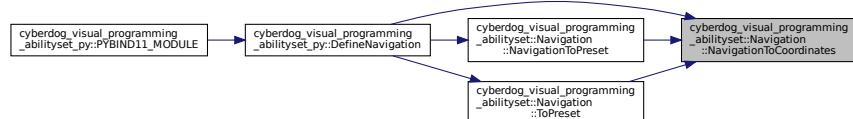
```

NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::NavigationTo←
Coordinates (
    const bool is_outdoor = false,
    const double x = 0,
    const double y = 0,
    const double z = 0,
    const double roll = 0,
    const double pitch = 0,
    const double yaw = 0 )
  
```

导航到坐标点 函数调用图:



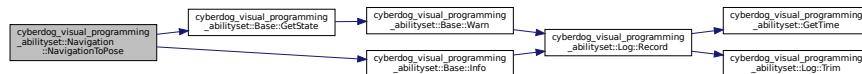
这是这个函数的调用关系图:



### 8.56.2.7 NavigationToPose()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::NavigationToPose (
    const MsgPose & _pose )
```

导航到位姿 函数调用图:



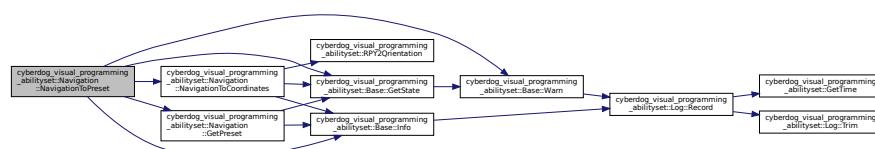
这是这个函数的调用关系图:



### 8.56.2.8 NavigationToPreset()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::NavigationToPreset
(
    const std::string & _preset_name )
```

导航到预置点 函数调用图:



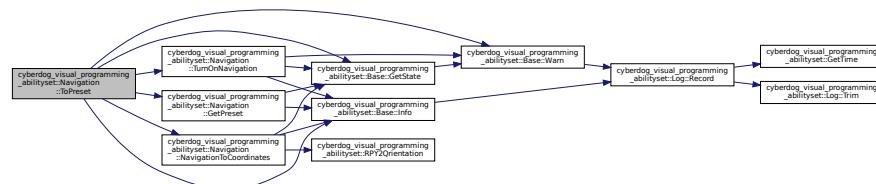
这是这个函数的调用关系图:



### 8.56.2.9 ToPreset()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::ToPreset (
    const std::string & preset_name,
    const bool assisted_relocation = false,
    const bool interact = false,
    const int volume = 50 )
```

去预置点 函数调用图:



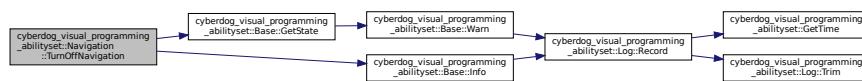
这是这个函数的调用关系图:



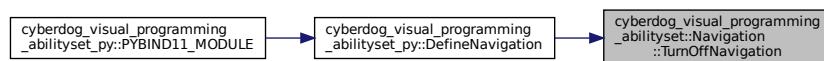
### 8.56.2.10 TurnOffNavigation()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::TurnOffNavigation ()
```

请求关闭重定位 函数调用图:



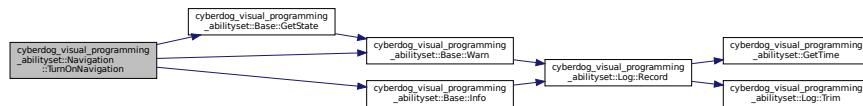
这是这个函数的调用关系图:



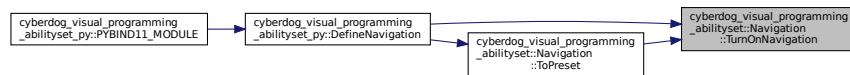
### 8.56.2.11 TurnOnNavigation()

```
NavigationActionResponse cyberdog_visual_programming_abilityset::Navigation::TurnOnNavigation (
    const bool outdoor = false,
    const bool assisted_relocation = false,
    const bool interact = false,
    const int volume = 50 )
```

请求开启重定位 函数调用图:



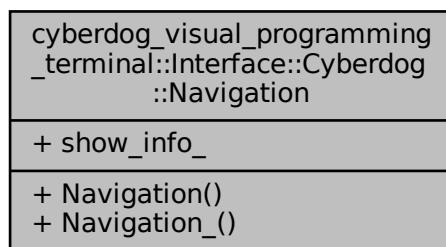
这是这个函数的调用关系图:



## 8.57 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Navigation 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Navigation 的协作图:



### Public 成员函数

- `Navigation ()`
- `void Navigation_(const std::string _fun)`

## Public 属性

- std::function< void(std::string) > [show\\_info\\_](#)

### 8.57.1 构造及析构函数说明

#### 8.57.1.1 Navigation()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation::Navigation ( ) [inline]
```

### 8.57.2 成员函数说明

#### 8.57.2.1 Navigation\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation::Navigation_ (
    const std::string _fun ) [inline]
```

### 8.57.3 类成员变量说明

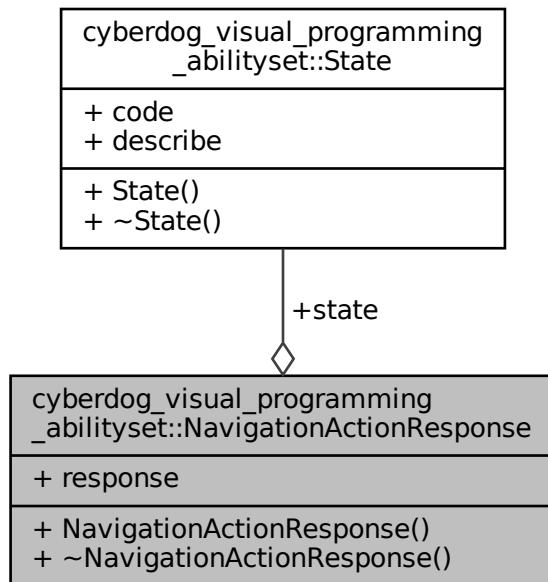
#### 8.57.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation::show_info_
```

## 8.58 cyberdog\_visual\_programming\_abilityset::NavigationActionResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::NavigationActionResponse 的协作图:



### Public 成员函数

- `NavigationActionResponse ()`
- `~NavigationActionResponse ()`

### Public 属性

- `State state`
- `ActNavigation::Result response`

#### 8.58.1 详细描述

导航反馈信息

#### 8.58.2 构造及析构函数说明

### 8.58.2.1 NavigationActionResponse()

```
cyberdog_visual_programming_abilityset::NavigationActionResponse::NavigationActionResponse ( )  
[inline]
```

### 8.58.2.2 ~NavigationActionResponse()

```
cyberdog_visual_programming_abilityset::NavigationActionResponse::~NavigationActionResponse ( )  
[inline]
```

## 8.58.3 类成员变量说明

### 8.58.3.1 response

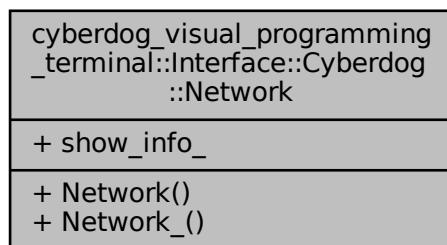
ActNavigation::Result cyberdog\_visual\_programming\_abilityset::NavigationActionResponse::response  
反馈

### 8.58.3.2 state

State cyberdog\_visual\_programming\_abilityset::NavigationActionResponse::state  
状态

## 8.59 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Network 结构体 参考

```
#include <interface.hpp>  
  
cyberdog_visual_programming_terminal::Interface::Cyberdog::Network 的协作图:
```



## Public 成员函数

- `Network()`
- `void Network_(const std::string &fun)`

## Public 属性

- `std::function<void(std::string)> show_info_`

### 8.59.1 构造及析构函数说明

#### 8.59.1.1 Network()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Network::Network() [inline]
```

### 8.59.2 成员函数说明

#### 8.59.2.1 Network\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Network::Network_(
    const std::string &fun) [inline]
```

### 8.59.3 类成员变量说明

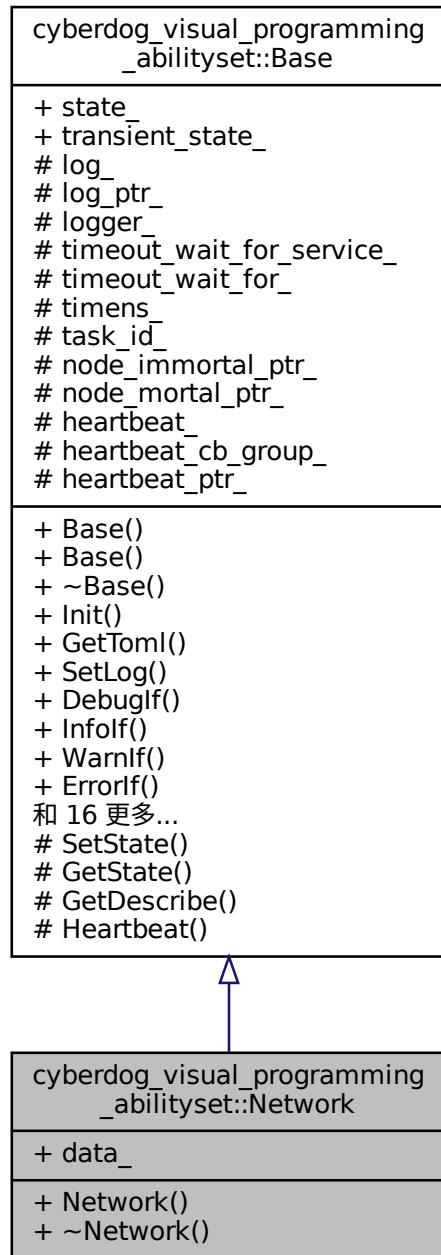
#### 8.59.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_
```

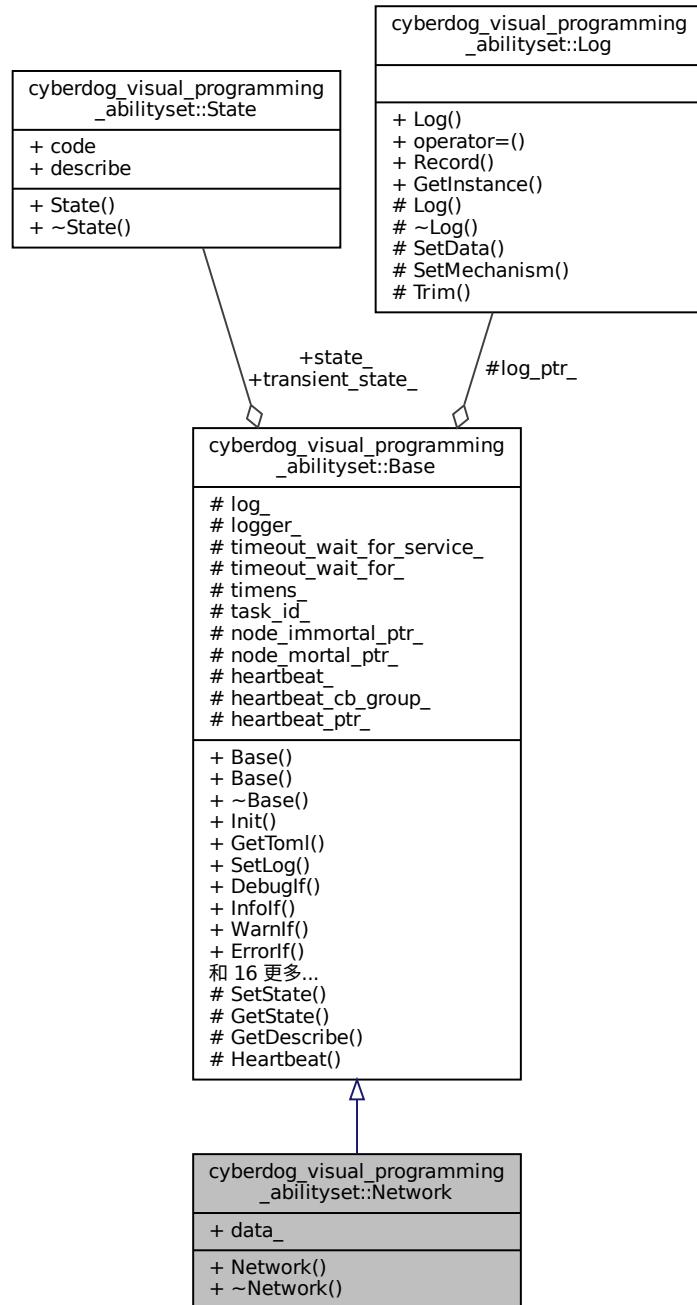
## 8.60 cyberdog\_visual\_programming\_abilityset::Network类 参考

```
#include <network.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Network 继承关系图:



cyberdog\_visual\_programming\_abilityset::Network 的协作图:



## Public 成员函数

- `Network ()`
- `~Network ()`

## Public 属性

- `MsgConnectorStatus data_`

额外继承的成员函数

### 8.60.1 构造及析构函数说明

#### 8.60.1.1 Network()

```
cyberdog_visual_programming_abilityset::Network::Network () [inline]
```

#### 8.60.1.2 ~Network()

```
cyberdog_visual_programming_abilityset::Network::~Network () [inline]
```

### 8.60.2 类成员变量说明

#### 8.60.2.1 data\_

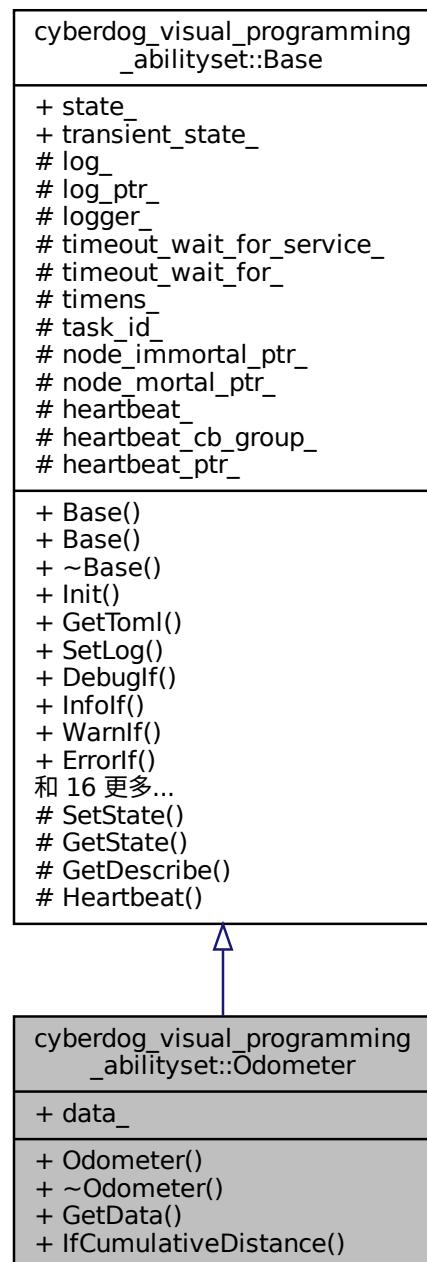
```
MsgConnectorStatus cyberdog_visual_programming_abilityset::Network::data_
```

网络

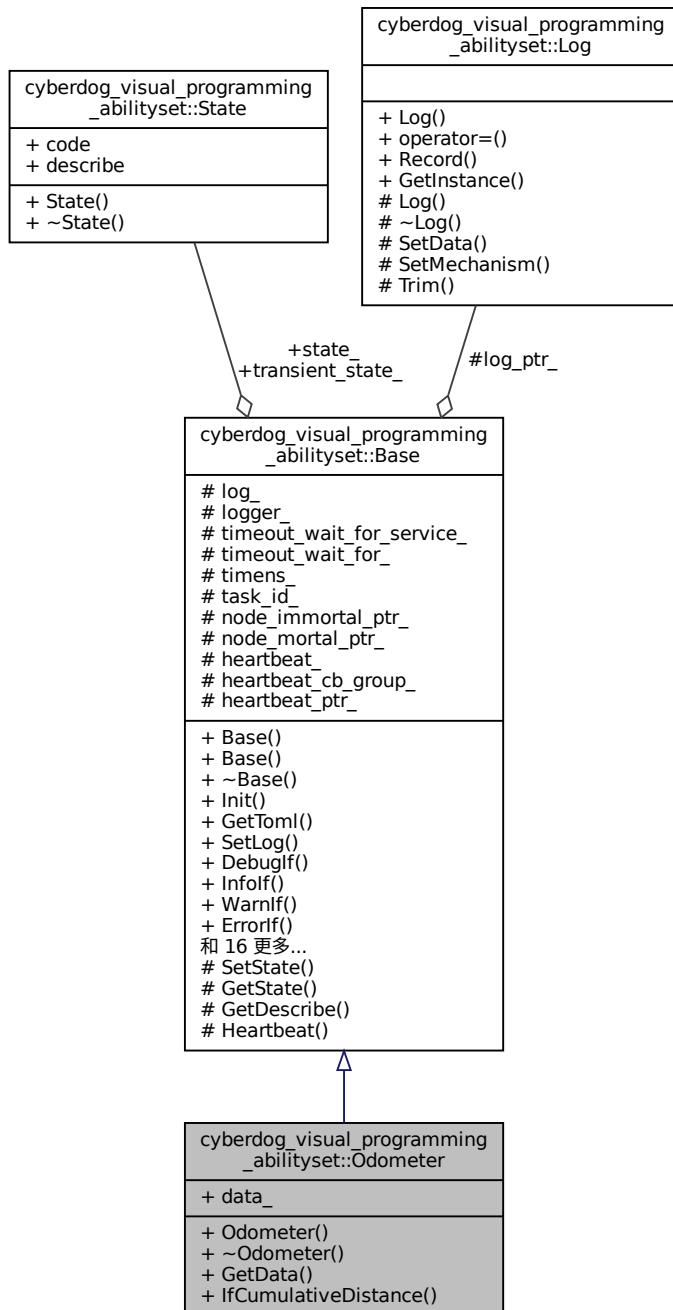
## 8.61 **cyberdog\_visual\_programming\_abilityset::Odometer**类 参考

```
#include <odometer.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Odometer 继承关系图:



cyberdog\_visual\_programming\_abilityset::Odometer 的协作图:



## Public 成员函数

- [Odometer \(\)](#)
- [~Odometer \(\)](#)
- [MsgOdometry GetData \(const int \\_timeout=5\)](#)
- [bool IfCumulativeDistance \(MsgOdometry &, double &, const double\)](#)

## Public 属性

- [MsgOdometry data\\_](#)

额外继承的成员函数

### 8.61.1 构造及析构函数说明

#### 8.61.1.1 Odometer()

```
cyberdog_visual_programming_abilityset::Odometer::Odometer ( ) [inline]
```

#### 8.61.1.2 ~Odometer()

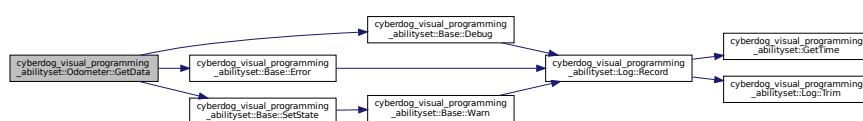
```
cyberdog_visual_programming_abilityset::Odometer::~Odometer ( ) [inline]
```

### 8.61.2 成员函数说明

#### 8.61.2.1 GetData()

```
MsgOdometry cyberdog_visual_programming_abilityset::Odometer::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.61.2.2 IfCumulativeDistance()

```
bool cyberdog_visual_programming_abilityset::Odometer::IfCumulativeDistance (
    MsgOdometry & _odom,
    double & _cumulative,
    const double _distance )
```

判断累计变化距离

### 8.61.3 类成员变量说明

#### 8.61.3.1 data\_

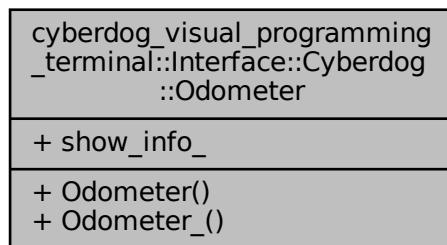
```
MsgOdometry cyberdog_visual_programming_abilityset::Odometer::data_
```

里程计

## 8.62 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Odometer结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Odometer 的协作图:



### Public 成员函数

- `Odometer ()`
- `void Odometer_ (const std::string _fun)`

## Public 属性

- std::function< void(std::string)> [show\\_info\\_](#)

### 8.62.1 构造及析构函数说明

#### 8.62.1.1 Odometer()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer::Odometer ( ) [inline]
```

### 8.62.2 成员函数说明

#### 8.62.2.1 Odometer\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer::Odometer_(
    const std::string &fun ) [inline]
```

### 8.62.3 类成员变量说明

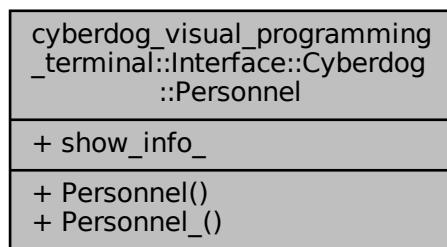
#### 8.62.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Odometer::show_info_
```

## 8.63 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog← ::Personnel结构体 参 考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Personnel 的协作图:



## Public 成员函数

- Personnel ()
- void Personnel\_ (const std::string &fun)

## Public 属性

- std::function< void(std::string)> show\_info\_

### 8.63.1 构造及析构函数说明

#### 8.63.1.1 Personnel()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel::Personnel ( ) [inline]
```

### 8.63.2 成员函数说明

#### 8.63.2.1 Personnel\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel::Personnel_ (
    const std::string &fun ) [inline]
```

### 8.63.3 类成员变量说明

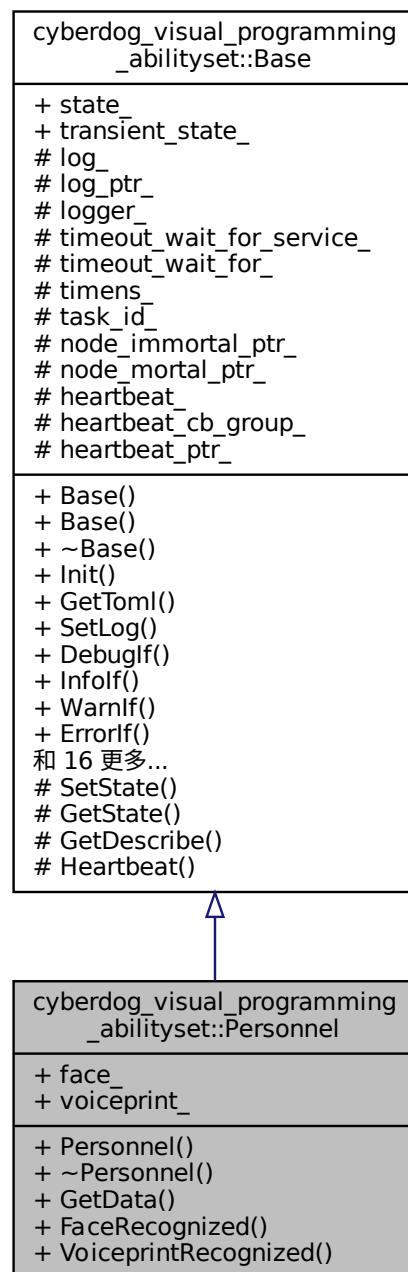
#### 8.63.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel::show_info_
```

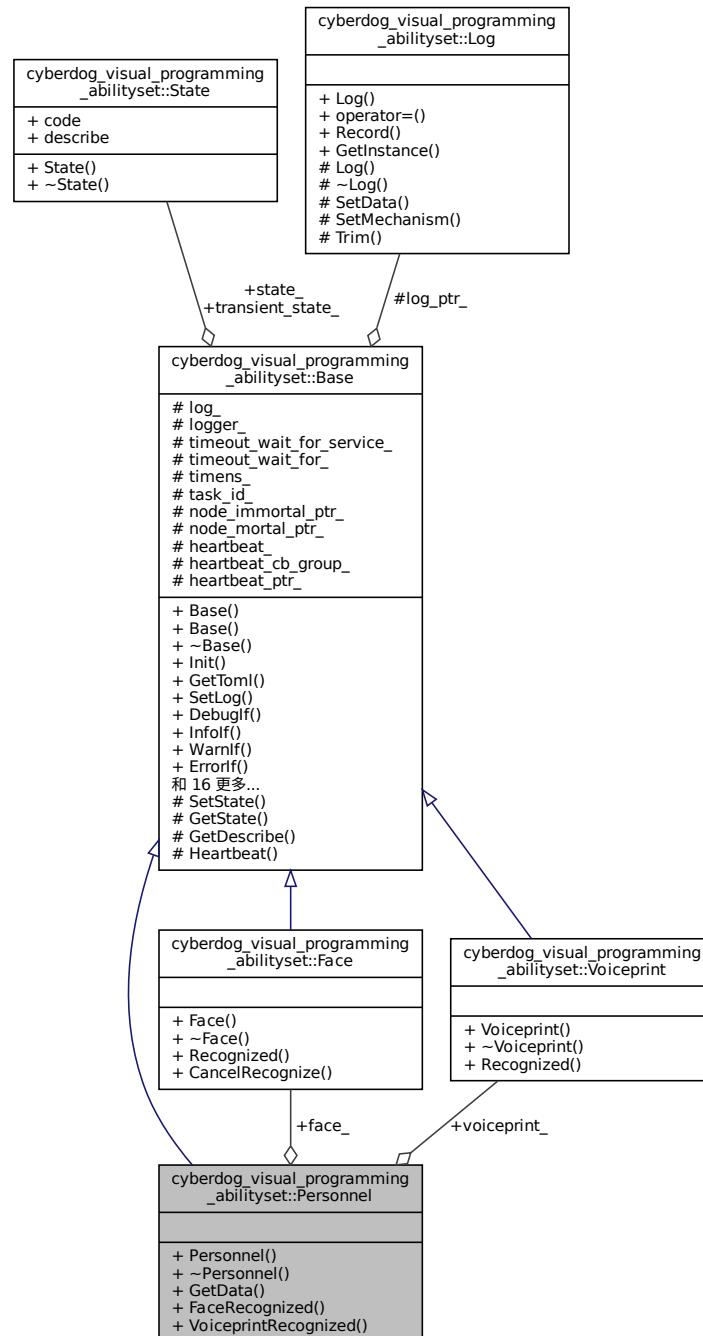
## 8.64 cyberdog\_visual\_programming\_abilityset::Personnel类 参考

```
#include <personnel.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Personnel 继承关系图:



cyberdog\_visual\_programming\_abilityset::Personnel 的协作图:



## Public 成员函数

- `Personnel ()`
- `~Personnel ()`
- `SrvPersonnel::Response GetData (const int _timeout=3)`
- `FaceRecognizedSeviceResponse FaceRecognized (const std::vector< std::string > &, const bool _and← operation=false, const int _duration=-1)`

- `VoiceprintRecognizedResponse VoiceprintRecognized (const std::vector< std::string > &, bool _and_<operation=false, const int _duration=-1, const int _sensitivity=1)`

## Public 属性

- `Face face_`
- `Voiceprint voiceprint_`

额外继承的成员函数

### 8.64.1 构造及析构函数说明

#### 8.64.1.1 Personnel()

```
cyberdog_visual_programming_abilityset::Personnel () [inline]
```

#### 8.64.1.2 ~Personnel()

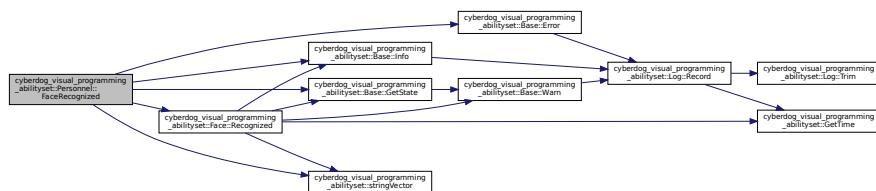
```
cyberdog_visual_programming_abilityset::Personnel::~Personnel () [inline]
```

### 8.64.2 成员函数说明

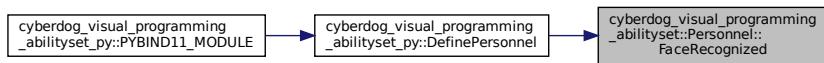
#### 8.64.2.1 FaceRecognized()

```
FaceRecognizedServiceResponse cyberdog_visual_programming_abilityset::Personnel::FaceRecognized (
    const std::vector< std::string > & _target,
    const bool _and_operation = false,
    const int _duration = -1 )
```

识别到目标人员人脸 函数调用图:



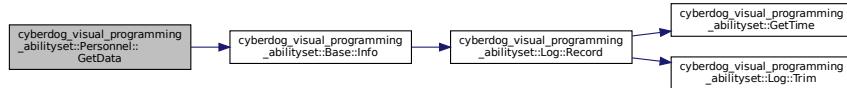
这是这个函数的调用关系图:



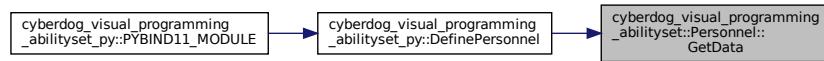
### 8.64.2.2 GetData()

```
SrvPersonnel::Response cyberdog_visual_programming_abilityset::Personnel::GetData (
    const int _timeout = 3 )
```

获取人员信息 函数调用图:



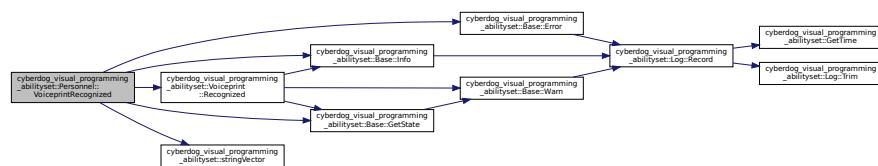
这是这个函数的调用关系图:



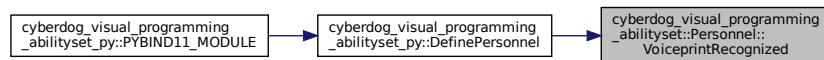
### 8.64.2.3 VoiceprintRecognized()

```
VoiceprintRecognizedResponse cyberdog_visual_programming_abilityset::Personnel::VoiceprintRecognized (
    const std::vector< std::string > &_target,
    bool _and_operation = false,
    const int _duration = -1,
    const int _sensitivity = 1 )
```

识别到目标人员声纹 函数调用图:



这是这个函数的调用关系图:



### 8.64.3 类成员变量说明

#### 8.64.3.1 face\_

`Face` `cyberdog_visual_programming_abilityset::Personnel::face_`

人脸识别句柄

#### 8.64.3.2 voiceprint\_

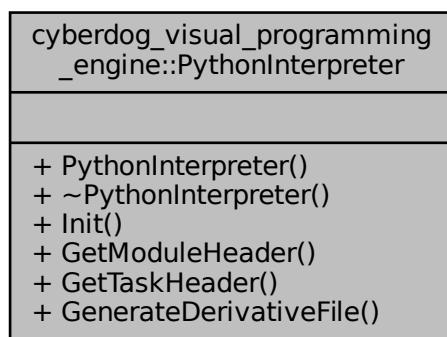
`Voiceprint` `cyberdog_visual_programming_abilityset::Personnel::voiceprint_`

声纹识别句柄

## 8.65 `cyberdog_visual_programming_engine::PythonInterpreter`类 参考

#include <python\_interpreter.hpp>

`cyberdog_visual_programming_engine::PythonInterpreter` 的协作图:



### Public 成员函数

- `PythonInterpreter ()`
- `~PythonInterpreter ()`
- `bool Init ()`
- `bool GetModuleHeader (const std::string &, const std::string &, const std::string &, std::vector< std::string > &, std::vector< std::string > &)`
- `bool GetTaskHeader (const std::string &, const std::string &, std::vector< std::string > &, std::vector< std::string > &)`
- `bool GenerateDerivativeFile (const std::string &)`

## 8.65.1 构造及析构函数说明

### 8.65.1.1 PythonInterpreter()

```
cyberdog_visual_programming_engine::PythonInterpreter::PythonInterpreter ( ) [inline]
```

### 8.65.1.2 ~PythonInterpreter()

```
cyberdog_visual_programming_engine::PythonInterpreter::~PythonInterpreter ( ) [inline]
```

## 8.65.2 成员函数说明

### 8.65.2.1 GenerateDerivativeFile()

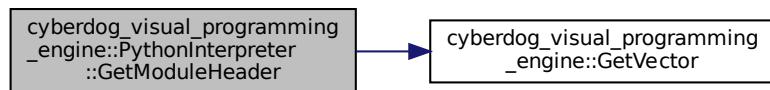
```
bool cyberdog_visual_programming_engine::PythonInterpreter::GenerateDerivativeFile ( const std::string & _source_file )
```

生成衍生文件

### 8.65.2.2 GetModuleHeader()

```
bool cyberdog_visual_programming_engine::PythonInterpreter::GetModuleHeader ( const std::string & _interface, const std::string & _describe, const std::string & _body, std::vector< std::string > & _header, std::vector< std::string > & _import_module )
```

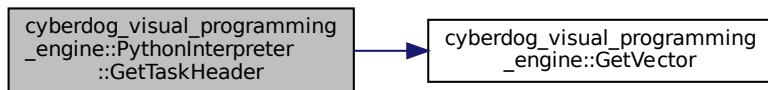
获取模块头 函数调用图:



### 8.65.2.3 GetTaskHeader()

```
bool cyberdog_visual_programming_engine::PythonInterpreter::GetTaskHeader ( const std::string & _id, const std::string & _body, std::vector< std::string > & _header, std::vector< std::string > & _import_module )
```

获取任务头 函数调用图:



### 8.65.2.4 Init()

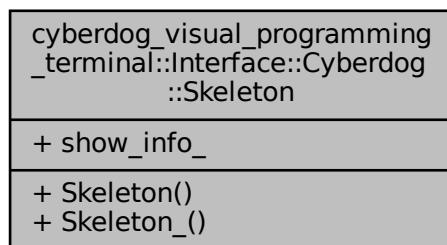
```
bool cyberdog_visual_programming_engine::PythonInterpreter::Init ( )
```

初始化

## 8.66 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog<--::Skeleton 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Skeleton 的协作图:



## Public 成员函数

- [Skeleton \(\)](#)
- [void Skeleton\\_\(const std::string &fun\)](#)

## Public 属性

- [std::function< void\(std::string\) > show\\_info\\_](#)

### 8.66.1 构造及析构函数说明

#### 8.66.1.1 Skeleton()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton () [inline]
```

### 8.66.2 成员函数说明

#### 8.66.2.1 Skeleton\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton_ (
    const std::string &fun ) [inline]
```

### 8.66.3 类成员变量说明

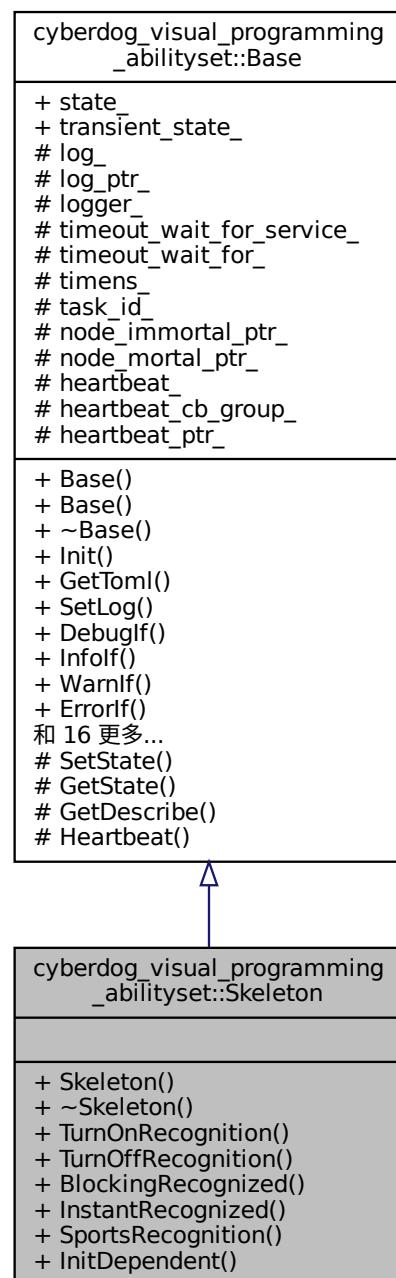
#### 8.66.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_
```

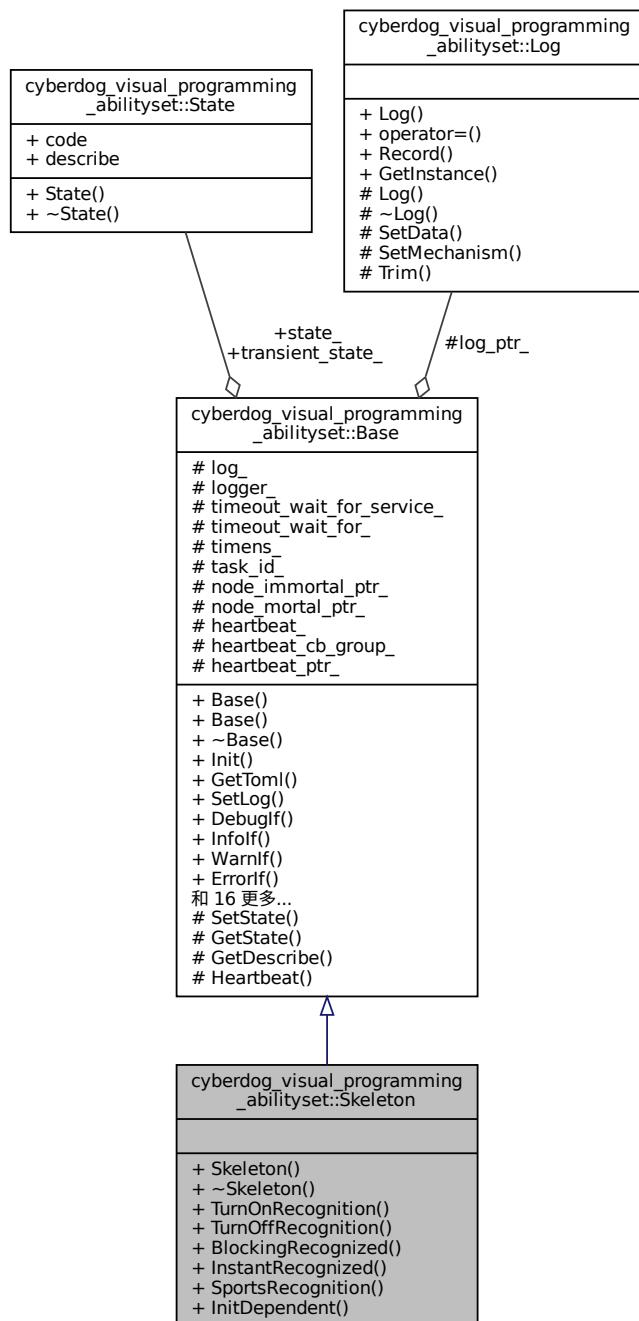
## 8.67 cyberdog\_visual\_programming\_abilityset::Skeleton类 参考

```
#include <skelton.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Skeleton 继承关系图:



cyberdog\_visual\_programming\_abilityset::Skeleton 的协作图:



## Public 成员函数

- `Skeleton ()`
- `~Skeleton ()`
- `SkeletonRecognizedSeviceResponse TurnOnRecognition (const uint, const int, const int)`
- `SkeletonRecognizedSeviceResponse TurnOffRecognition ()`
- `SkeletonRecognizedMessageResponse BlockingRecognized (const int timeout=-1)`

- `SkeletonRecognizedMessageResponse InstantRecognized ()`
- `SkeletonRecognizedMessageResponse SportsRecognition (const uint, const int, const int, const bool interact=false, const bool instantly=false, const int volume=50)`
- `void InitDependent (const std::function< AudioPlayServiceResponse>(const std::string, const int8_t)> &, const std::function< State>(const std::string, const int8_t)> &)`

额外继承的成员函数

### 8.67.1 构造及析构函数说明

#### 8.67.1.1 `Skeleton()`

```
cyberdog_visual_programming_abilityset::Skeleton::Skeleton ( ) [inline]
```

#### 8.67.1.2 `~Skeleton()`

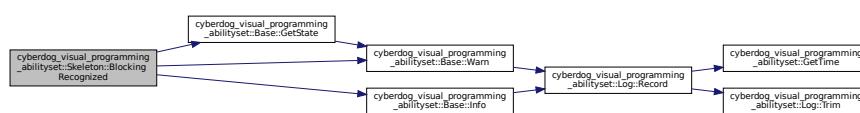
```
cyberdog_visual_programming_abilityset::Skeleton::~Skeleton ( ) [inline]
```

### 8.67.2 成员函数说明

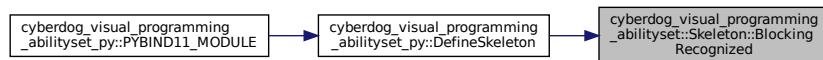
#### 8.67.2.1 `BlockingRecognized()`

```
SkeletonRecognizedMessageResponse cyberdog_visual_programming_abilityset::Skeleton::BlockingRecognized (
    const int timeout = -1 )
```

阻塞式识别到: 状态改变才返回 函数调用图:



这是这个函数的调用关系图:



### 8.67.2.2 InitDependent()

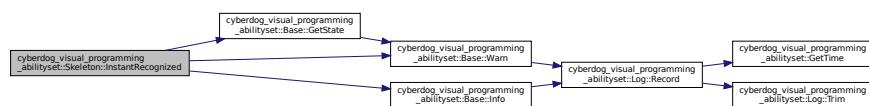
```
void cyberdog_visual_programming_abilityset::Skeleton::InitDependent (
    const std::function< AudioPlayServiceResponse(const std::string, const int8_t)> &_Play,
    const std::function< State(const std::string, const int8_t)> &_InstantlyPlay )
```

初始化依赖

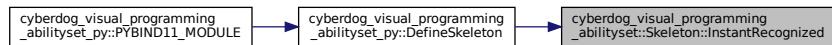
### 8.67.2.3 InstantRecognized()

```
SkeletonRecognizedMessageResponse cyberdog_visual_programming_abilityset::Skeleton::InstantRecognized ()
```

瞬时式识别到：返回最近的状态 函数调用图:



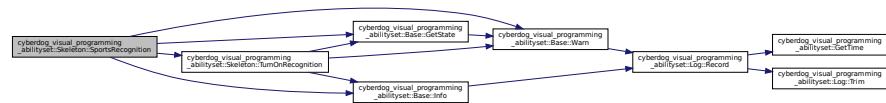
这是这个函数的调用关系图:



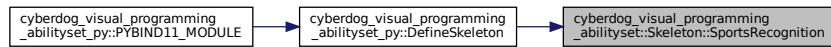
### 8.67.2.4 SportsRecognition()

```
SkeletonRecognizedMessageResponse cyberdog_visual_programming_abilityset::Skeleton::SportsRecognition (
    const uint _sport_type,
    const int _counts,
    const int _timeout,
    const bool interact = false,
    const bool instantly = false,
    const int volume = 50 )
```

运动识别 函数调用图:



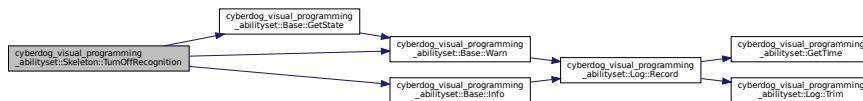
这是这个函数的调用关系图:



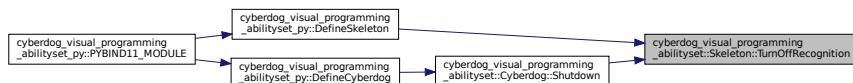
### 8.67.2.5 TurnOffRecognition()

```
SkeletonRecognizedServiceResponse cyberdog_visual_programming_abilityset::Skeleton::TurnOffRecognition ()
```

关闭识别骨骼（点）功能 函数调用图:



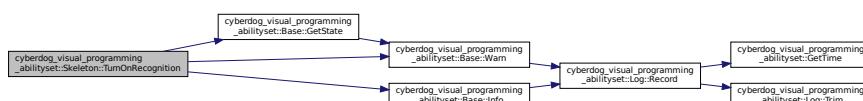
这是这个函数的调用关系图:



### 8.67.2.6 TurnOnRecognition()

```
SkeletonRecognizedServiceResponse cyberdog_visual_programming_abilityset::Skeleton::TurnOnRecognition (
    const uint _sport_type,
    const int _counts,
    const int _timeout )
```

打开识别骨骼（点）功能 函数调用图:



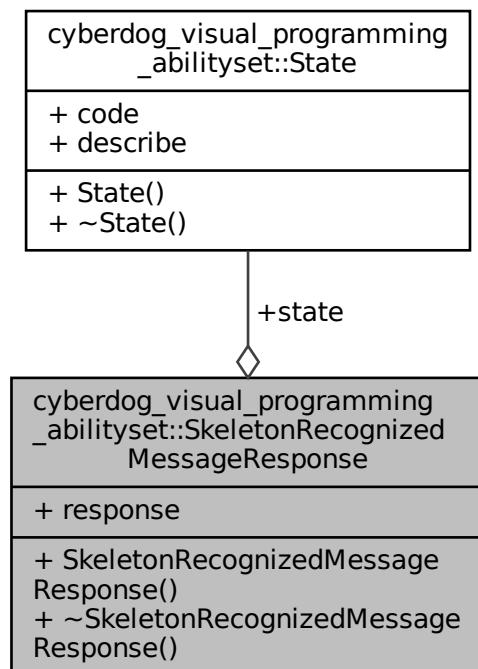
这是这个函数的调用关系图:



## 8.68 cyberdog\_visual\_programming\_abilityset::SkeletonRecognizedMessageResponse类 参考

#include <common.hpp>

cyberdog\_visual\_programming\_abilityset::SkeletonRecognizedMessageResponse 的协作图:



### Public 成员函数

- `SkeletonRecognizedMessageResponse ()`
- `~SkeletonRecognizedMessageResponse ()`

### Public 属性

- `State state`
- `MsgSport response`

### 8.68.1 详细描述

骨骼点识别消息反馈

### 8.68.2 构造及析构函数说明

#### 8.68.2.1 **SkeletonRecognizedMessageResponse()**

```
cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse::SkeletonRecognized<→
MessageResponse ( ) [inline]
```

#### 8.68.2.2 **~SkeletonRecognizedMessageResponse()**

```
cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse::~SkeletonRecognized<→
MessageResponse ( ) [inline]
```

### 8.68.3 类成员变量说明

#### 8.68.3.1 **response**

```
MsgSport cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse::response
```

反馈

#### 8.68.3.2 **state**

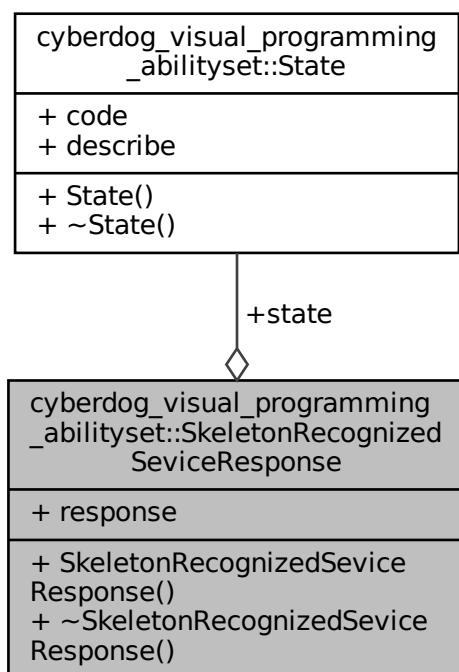
```
State cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse::state
```

状态

## 8.69 cyberdog\_visual\_programming\_abilityset::SkeletonRecognizedSeviceResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::SkeletonRecognizedSeviceResponse 的协作图:



### Public 成员函数

- `SkeletonRecognizedSeviceResponse ()`
- `~SkeletonRecognizedSeviceResponse ()`

### Public 属性

- `State state`
- `SrvSport::Response response`

#### 8.69.1 详细描述

骨骼点识别服务反馈

## 8.69.2 构造及析构函数说明

### 8.69.2.1 SkeletonRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse::SkeletonRecognized<-
SeviceResponse ( ) [inline]
```

### 8.69.2.2 ~SkeletonRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse::~SkeletonRecognized<-
SeviceResponse ( ) [inline]
```

## 8.69.3 类成员变量说明

### 8.69.3.1 response

```
SrvSport::Response cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse<-
::response
```

反馈

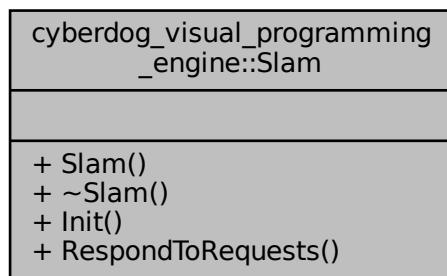
### 8.69.3.2 state

```
State cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse::state
状态
```

## 8.70 cyberdog\_visual\_programming\_engine::Slam类参考

```
#include <slam.hpp>
```

cyberdog\_visual\_programming\_engine::Slam 的协作图:



## Public 成员函数

- [Slam \(\)](#)
- [~Slam \(\)](#)
- [bool Init \(const rclcpp::Node::SharedPtr &, const toml::value &\)](#)
- [bool RespondToRequests \(const OperateMsg &, GRPCMsg &\)](#)

### 8.70.1 构造及析构函数说明

#### 8.70.1.1 Slam()

```
cyberdog_visual_programming_engine::Slam::Slam ( )
```

#### 8.70.1.2 ~Slam()

```
cyberdog_visual_programming_engine::Slam::~Slam ( )
```

### 8.70.2 成员函数说明

#### 8.70.2.1 Init()

```
bool cyberdog_visual_programming_engine::Slam::Init (
    const rclcpp::Node::SharedPtr & _node_ptr,
    const toml::value & _params_toml )
```

初始化

#### 8.70.2.2 RespondToRequests()

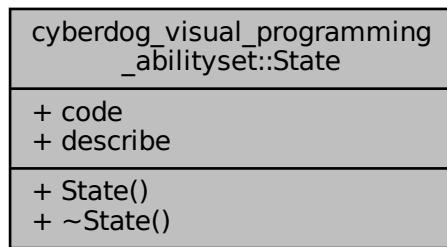
```
bool cyberdog_visual_programming_engine::Slam::RespondToRequests (
    const OperateMsg & _msg,
    GRPCMsg & msg_ )
```

执行请求

## 8.71 cyberdog\_visual\_programming\_abilityset::State类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::State 的协作图:



### Public 成员函数

- [State \(\)](#)
- [~State \(\)](#)

### Public 属性

- [StateCode code](#)
- [std::string describe](#)

#### 8.71.1 详细描述

参数约束:默认值和最大值

#### 8.71.2 构造及析构函数说明

##### 8.71.2.1 State()

```
cyberdog_visual_programming_abilityset::State::State ( ) [inline]
```

### 8.71.2.2 ~State()

```
cyberdog_visual_programming_abilityset::State::~State ( ) [inline]
```

## 8.71.3 类成员变量说明

### 8.71.3.1 code

```
StateCode cyberdog_visual_programming_abilityset::State::code
```

状态

### 8.71.3.2 describe

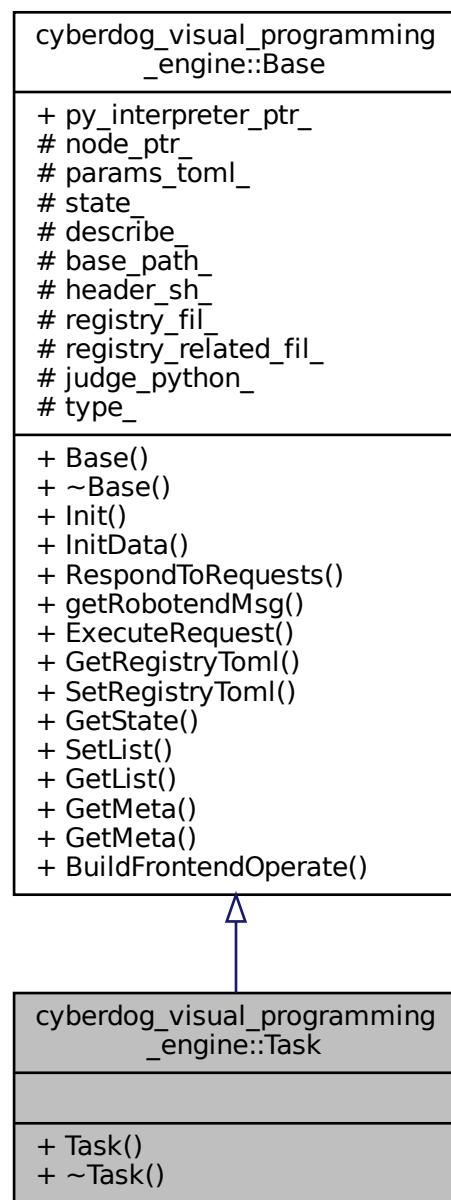
```
std::string cyberdog_visual_programming_abilityset::State::describe
```

描述

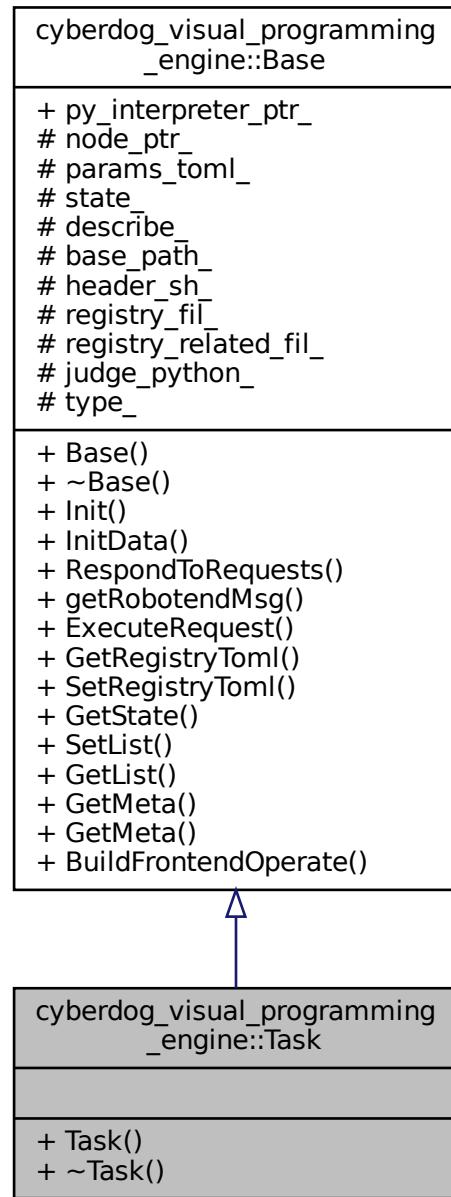
## 8.72 cyberdog\_visual\_programming\_engine::Task类 参考

```
#include <task.hpp>
```

类 `cyberdog_visual_programming_engine::Task` 继承关系图:



cyberdog\_visual\_programming\_engine::Task 的协作图:



## Public 成员函数

- [Task \(\)](#)
- [~Task \(\)](#)

额外继承的成员函数

### 8.72.1 构造及析构函数说明

#### 8.72.1.1 Task()

```
cyberdog_visual_programming_engine::Task::Task ( )
```

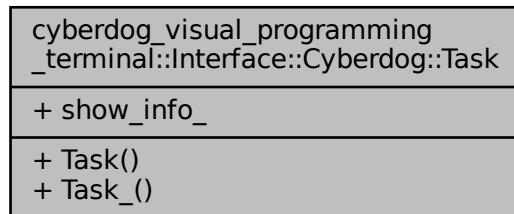
#### 8.72.1.2 ~Task()

```
cyberdog_visual_programming_engine::Task::~Task ( )
```

## 8.73 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog← ::Task结构体 参 考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Task 的协作图:



### Public 成员函数

- [Task \(\)](#)
- void [Task\\_](#)(const std::string \_fun)

### Public 属性

- std::function< void(std::string)> [show\\_info\\_](#)

### 8.73.1 构造及析构函数说明

#### 8.73.1.1 Task()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Task::Task ( ) [inline]
```

### 8.73.2 成员函数说明

#### 8.73.2.1 Task\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Task::Task_ (
    const std::string &fun ) [inline]
```

### 8.73.3 类成员变量说明

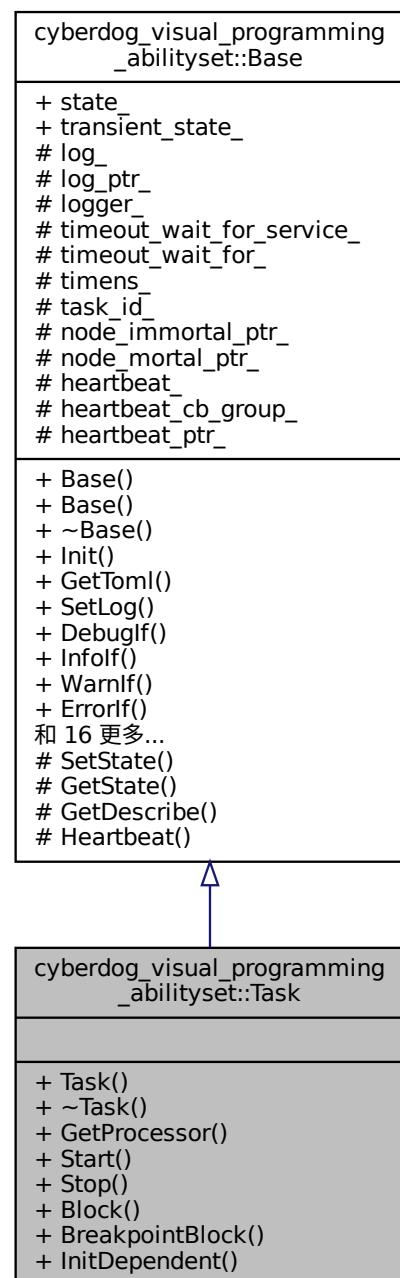
#### 8.73.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Task::show_info_
```

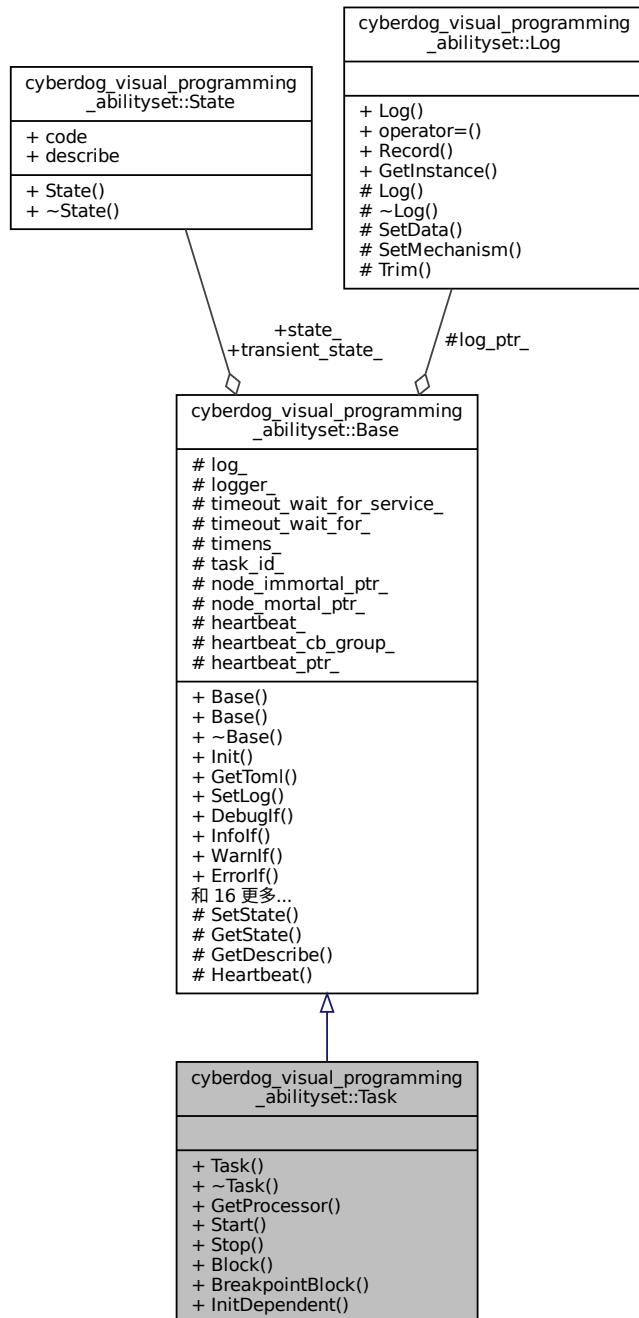
## 8.74 **cyberdog\_visual\_programming\_abilityset::Task**类 参考

```
#include <task.hpp>
```

类 `cyberdog_visual_programming_abilityset::Task` 继承关系图:



cyberdog\_visual\_programming\_abilityset::Task 的协作图:



## Public 成员函数

- `Task ()`
- `~Task ()`
- `bool GetProcessor (Processor)`
- `State Start ()`
- `State Stop ()`

- `State Block` (`const std::string id`)
- `State BreakpointBlock` (`const std::string id`)
- `void InitDependent` (`const std::function< void(bool)> &, const std::function< AudioPlaySeviceResponse(const std::string, const int8_t)> &`)

额外继承的成员函数

### 8.74.1 构造及析构函数说明

#### 8.74.1.1 Task()

```
cyberdog_visual_programming_abilityset::Task::Task ( ) [inline]
```

#### 8.74.1.2 ~Task()

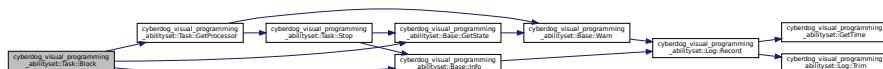
```
cyberdog_visual_programming_abilityset::Task::~Task ( ) [inline]
```

### 8.74.2 成员函数说明

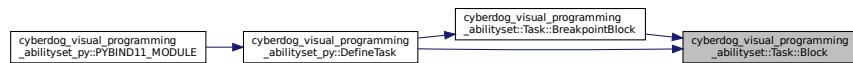
#### 8.74.2.1 Block()

```
State cyberdog_visual_programming_abilityset::Task::Block (
    const std::string id )
```

设置块 函数调用图:



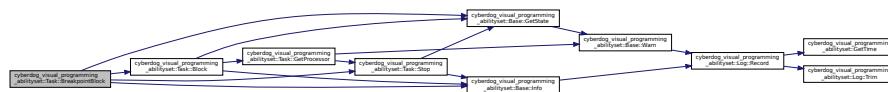
这是这个函数的调用关系图:



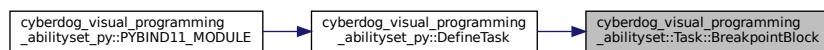
### 8.74.2.2 BreakpointBlock()

```
State cyberdog_visual_programming_abilityset::Task::BreakpointBlock (
    const std::string id )
```

设置断点块 函数调用图:



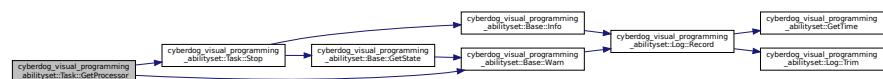
这是这个函数的调用关系图:



### 8.74.2.3 GetProcessor()

```
bool cyberdog_visual_programming_abilityset::Task::GetProcessor (
    Processor &processor )
```

获取处理器 函数调用图:



这是这个函数的调用关系图:



### 8.74.2.4 InitDependent()

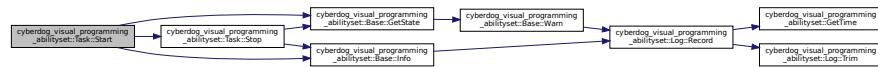
```
void cyberdog_visual_programming_abilityset::Task::InitDependent (
    const std::function< void(bool) > & _FShutdown,
    const std::function< AudioPlaySeviceResponse(const std::string, const int8_t) > &
    _FAudioPlay )
```

初始化依赖

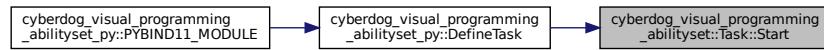
### 8.74.2.5 Start()

```
State cyberdog_visual_programming_abilityset::Task::Start ( )
```

开始任务 函数调用图:



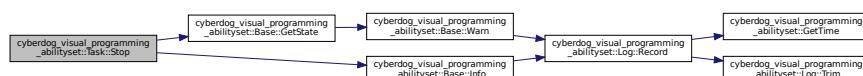
这是这个函数的调用关系图:



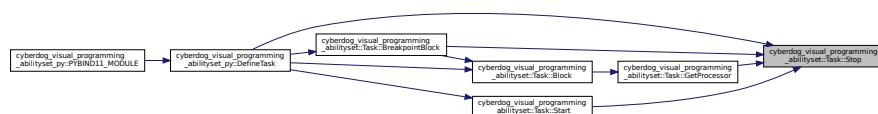
### 8.74.2.6 Stop()

```
State cyberdog_visual_programming_abilityset::Task::Stop ( )
```

结束任务 函数调用图:



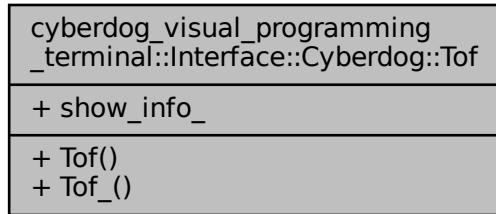
这是这个函数的调用关系图:



## 8.75 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Tof 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Tof 的协作图:



### Public 成员函数

- [Tof\(\)](#)
- void [Tof\\_\(const std::string &fun\)](#)

### Public 属性

- std::function< void(std::string)> [show\\_info\\_\(\)](#)

#### 8.75.1 构造及析构函数说明

##### 8.75.1.1 Tof()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof::Tof ( ) [inline]
```

#### 8.75.2 成员函数说明

##### 8.75.2.1 Tof\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof::Tof_ (
    const std::string &fun ) [inline]
```

### 8.75.3 类成员变量说明

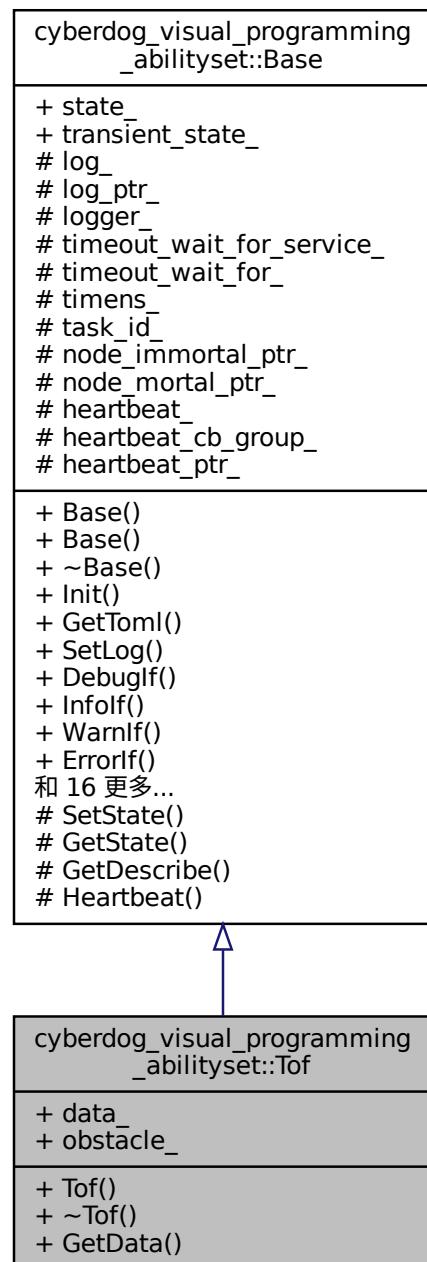
#### 8.75.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::←
Tof::show_info_
```

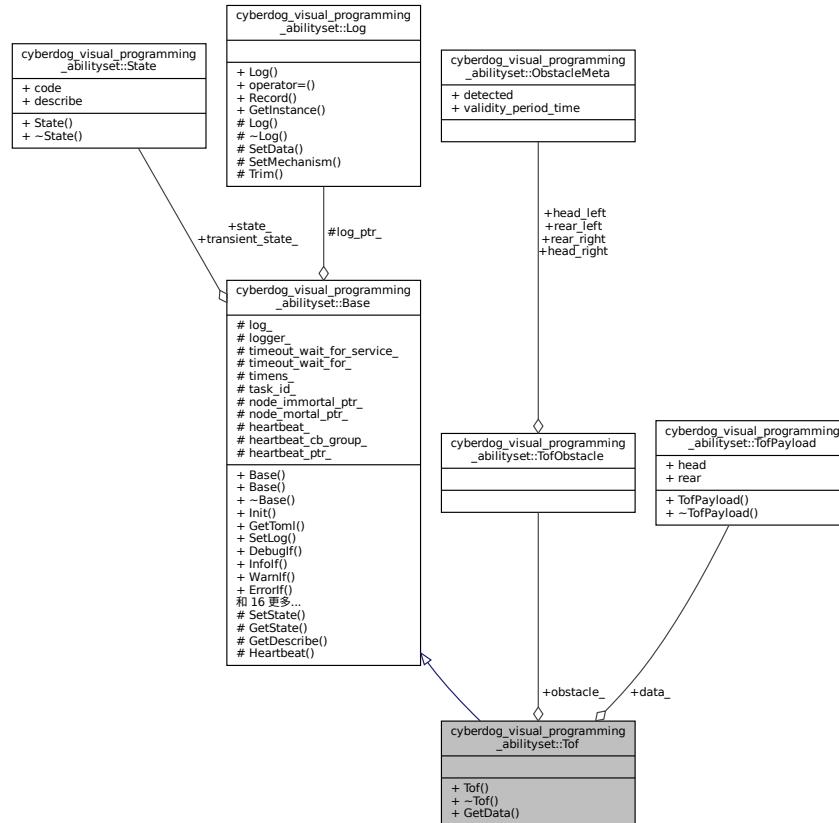
## 8.76 cyberdog\_visual\_programming\_abilityset::Tof类 参考

```
#include <tof.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Tof 继承关系图:



cyberdog\_visual\_programming\_abilityset::Tof 的协作图:



## Public 成员函数

- `Tof ()`
- `~Tof ()`
- `TofPayload GetData (const int _timeout=5)`

## Public 属性

- `TofPayload data_`
- `TofObstacle obstacle_`

额外继承的成员函数

### 8.76.1 构造及析构函数说明

### 8.76.1.1 Tof()

```
cyberdog_visual_programming_abilityset::Tof::Tof ( ) [inline]
```

### 8.76.1.2 ~Tof()

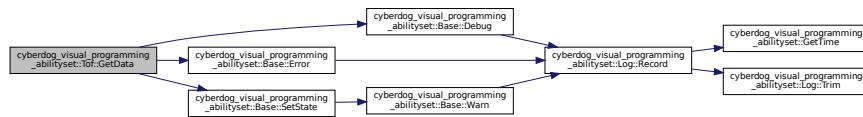
```
cyberdog_visual_programming_abilityset::Tof::~Tof ( ) [inline]
```

## 8.76.2 成员函数说明

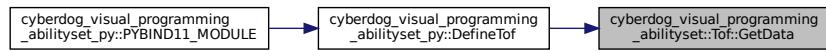
### 8.76.2.1 GetData()

```
TofPayload cyberdog_visual_programming_abilityset::Tof::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



## 8.76.3 类成员变量说明

### 8.76.3.1 data\_

```
TofPayload cyberdog_visual_programming_abilityset::Tof::data_
```

数据

### 8.76.3.2 obstacle\_

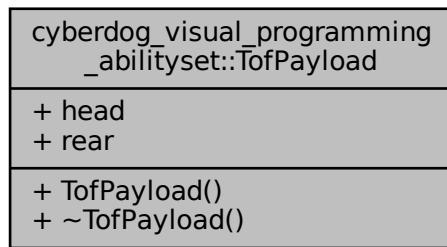
`TofObstacle` `cyberdog_visual_programming_abilityset::Tof::obstacle_`

障碍物

## 8.77 `cyberdog_visual_programming_abilityset::TofPayload`类 参考

`#include <common.hpp>`

`cyberdog_visual_programming_abilityset::TofPayload` 的协作图:



### Public 成员函数

- `TofPayload ()`
- `~TofPayload ()`

### Public 属性

- `MsgHeadTofPayload head`
- `MsgRearTofPayload rear`

### 8.77.1 详细描述

参数约束:tof数据

### 8.77.2 构造及析构函数说明

### 8.77.2.1 TofPayload()

```
cyberdog_visual_programming_abilityset::TofPayload::TofPayload () [inline]
```

### 8.77.2.2 ~TofPayload()

```
cyberdog_visual_programming_abilityset::TofPayload::~TofPayload () [inline]
```

## 8.77.3 类成员变量说明

### 8.77.3.1 head

```
MsgHeadTofPayload cyberdog_visual_programming_abilityset::TofPayload::head
```

头部数据

### 8.77.3.2 rear

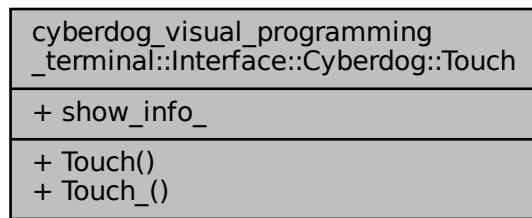
```
MsgRearTofPayload cyberdog_visual_programming_abilityset::TofPayload::rear
```

尾部数据

## 8.78 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Touch结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Touch 的协作图:



## Public 成员函数

- `Touch()`
- `void Touch_(const std::string &fun)`

## Public 属性

- `std::function<void(std::string)> show_info_`

### 8.78.1 构造及析构函数说明

#### 8.78.1.1 `Touch()`

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch::Touch () [inline]
```

### 8.78.2 成员函数说明

#### 8.78.2.1 `Touch_()`

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch::Touch_ (
    const std::string &fun ) [inline]
```

### 8.78.3 类成员变量说明

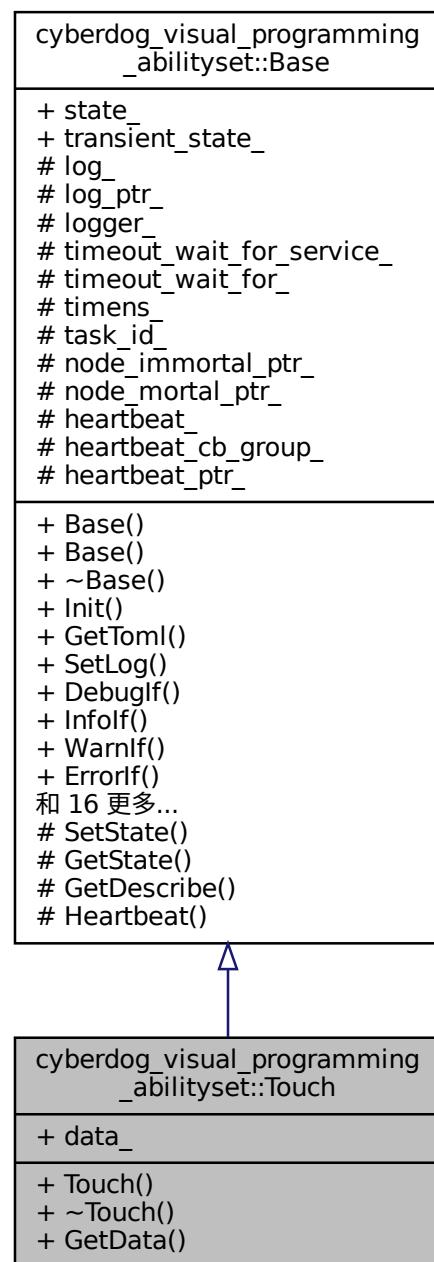
#### 8.78.3.1 `show_info_`

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch::show_info_
```

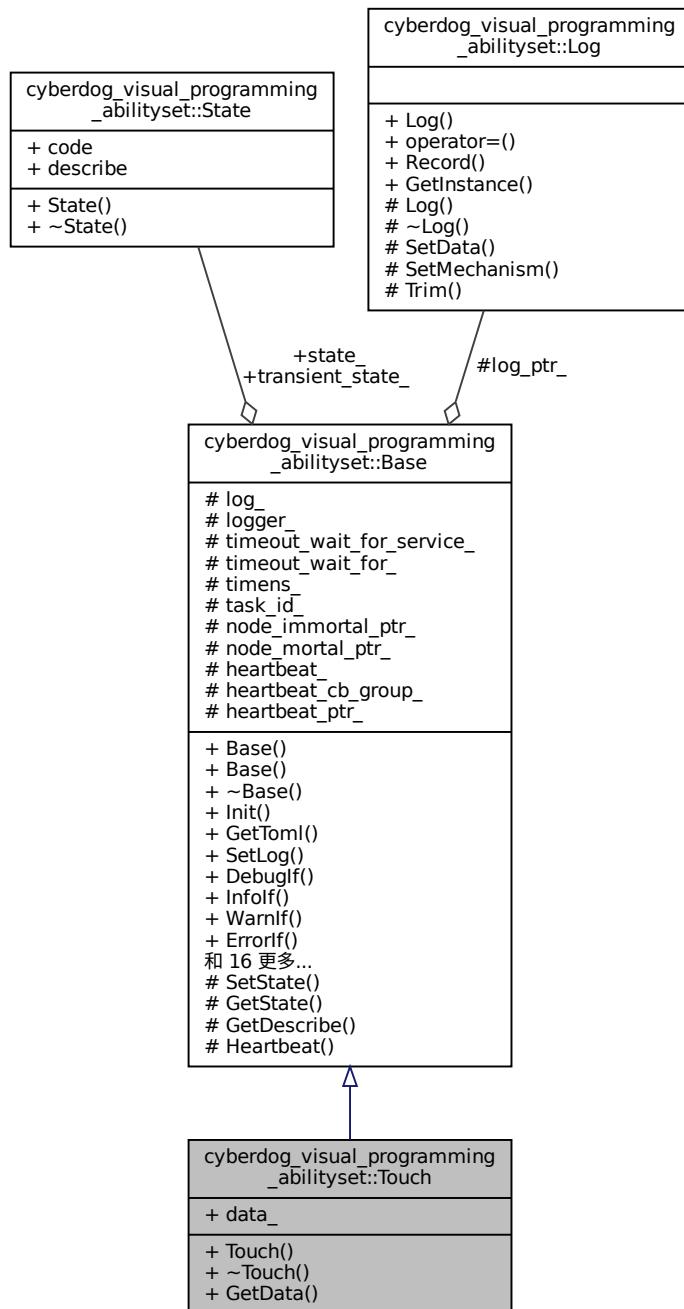
## 8.79 cyberdog\_visual\_programming\_abilityset::Touch类 参考

```
#include <touch.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Touch 继承关系图:



cyberdog\_visual\_programming\_abilityset::Touch 的协作图:



## Public 成员函数

- [Touch \(\)](#)
- [~Touch \(\)](#)
- [MsgTouchStatus GetData \(const int \\_timeout=5\)](#)

## Public 属性

- `MsgTouchStatus data_`

额外继承的成员函数

### 8.79.1 构造及析构函数说明

#### 8.79.1.1 Touch()

```
cyberdog_visual_programming_abilityset::Touch::Touch ( ) [inline]
```

#### 8.79.1.2 ~Touch()

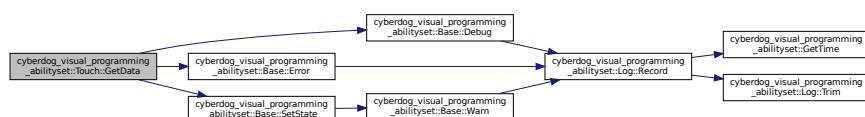
```
cyberdog_visual_programming_abilityset::Touch::~Touch ( ) [inline]
```

### 8.79.2 成员函数说明

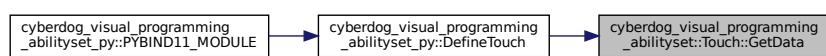
#### 8.79.2.1 GetData()

```
MsgTouchStatus cyberdog_visual_programming_abilityset::Touch::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.79.3 类成员变量说明

#### 8.79.3.1 data\_-

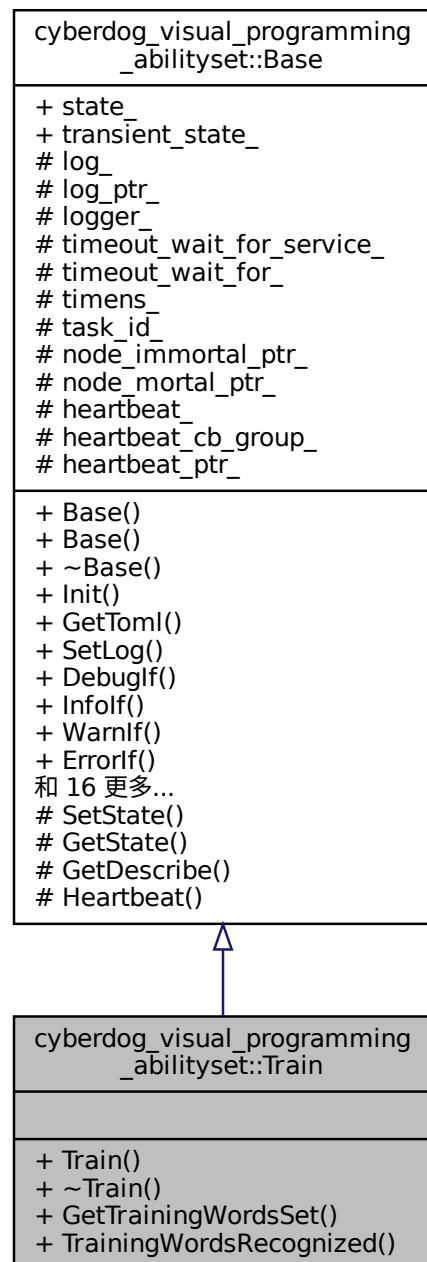
```
MsgTouchStatus cyberdog_visual_programming_abilityset::Touch::data_-
```

数据

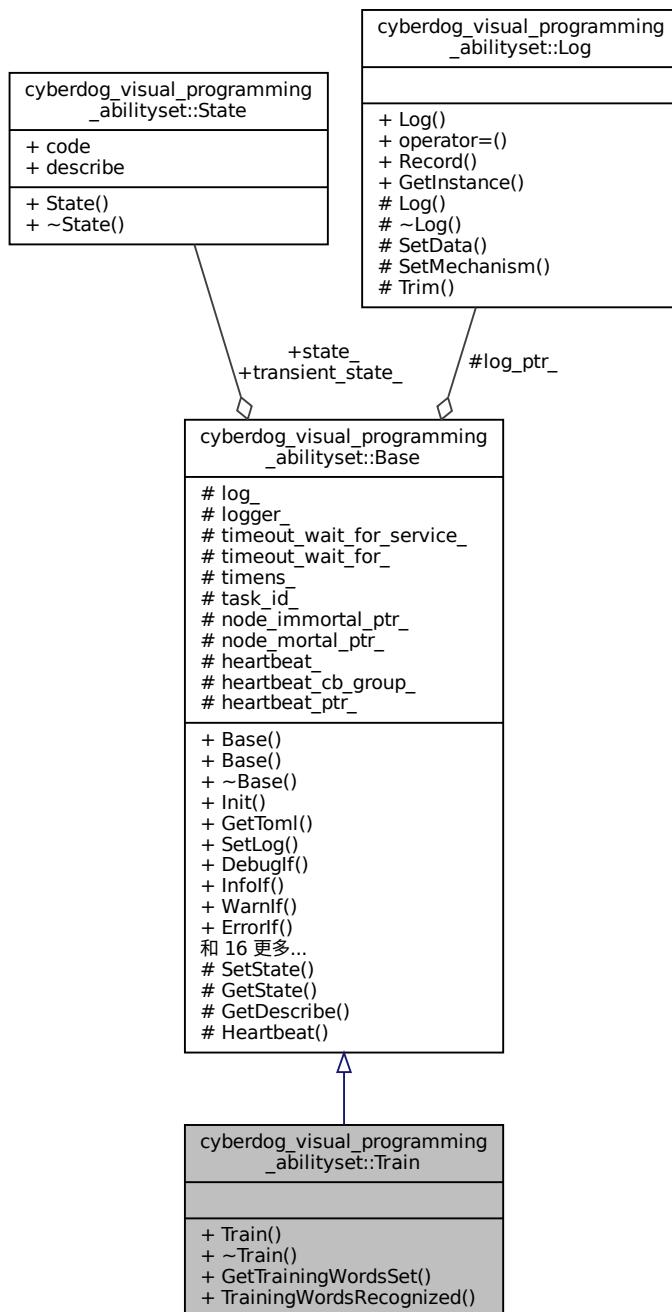
## 8.80 cyberdog\_visual\_programming\_abilityset::Train类 参考

```
#include <train.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Train 继承关系图:



cyberdog\_visual\_programming\_abilityset::Train 的协作图:



## Public 成员函数

- `Train ()`
- `~Train ()`
- `TrainingWordsRecognizedSeviceResponse GetTrainingWordsSet ()`
- `TrainingWordsRecognizedMessageResponse TrainingWordsRecognized (const int timeout=-1)`

额外继承的成员函数

### 8.80.1 构造及析构函数说明

#### 8.80.1.1 Train()

```
cyberdog_visual_programming_abilityset::Train::Train ( ) [inline]
```

#### 8.80.1.2 ~Train()

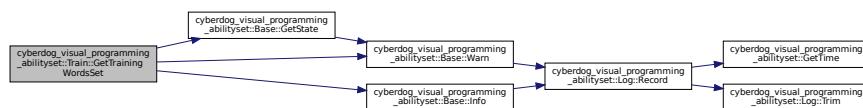
```
cyberdog_visual_programming_abilityset::Train::~Train ( ) [inline]
```

### 8.80.2 成员函数说明

#### 8.80.2.1 GetTrainingWordsSet()

```
TrainingWordsRecognizedServiceResponse cyberdog_visual_programming_abilityset::Train::GetTrainingWordsSet ( )
```

获取训练词集合 函数调用图:



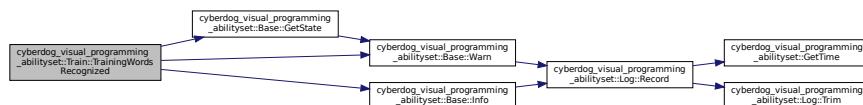
这是这个函数的调用关系图:



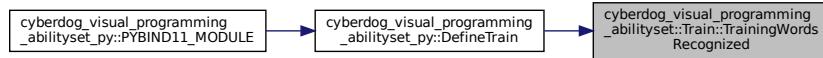
### 8.80.2.2 TrainingWordsRecognized()

```
TrainingWordsRecognizedMessageResponse cyberdog_visual_programming_abilityset::Train::Training<-
WordsRecognized (
    const int timeout = -1 )
```

识别到训练词 函数调用图:



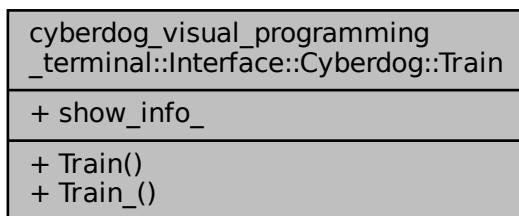
这是这个函数的调用关系图:



## 8.81 cyberdog\_visual\_programming\_terminal::Interface::Cyberdog<- ::Train 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Cyberdog::Train 的协作图:



### Public 成员函数

- `Train ()`
- `void Train_(const std::string _fun)`

## Public 属性

- std::function< void(std::string) > [show\\_info\\_](#)

### 8.81.1 构造及析构函数说明

#### 8.81.1.1 Train()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Train::Train ( ) [inline]
```

### 8.81.2 成员函数说明

#### 8.81.2.1 Train\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Train::Train_ (
    const std::string _fun ) [inline]
```

### 8.81.3 类成员变量说明

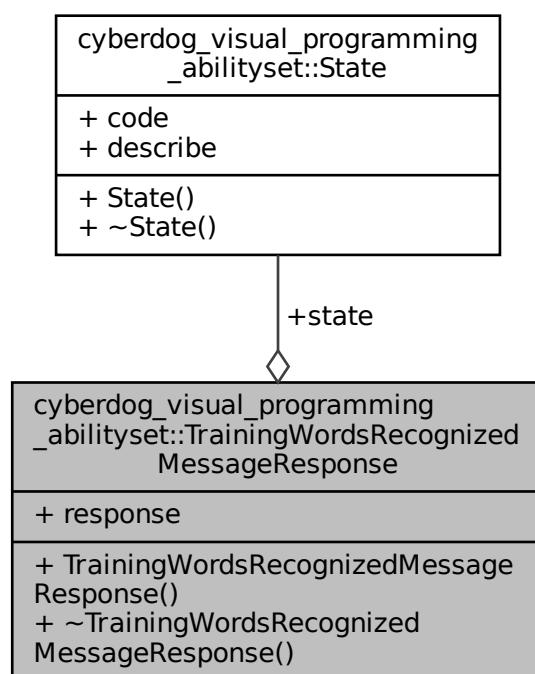
#### 8.81.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::Train::show_info_
```

## 8.82 cyberdog\_visual\_programming\_abilityset::TrainingWordsRecognizedMessageResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::TrainingWordsRecognizedMessageResponse 的协作图:



### Public 成员函数

- `TrainingWordsRecognizedMessageResponse ()`
- `~TrainingWordsRecognizedMessageResponse ()`

### Public 属性

- `State state`
- `MsgTrainingWords response`

#### 8.82.1 详细描述

训练词识别消息反馈

## 8.82.2 构造及析构函数说明

### 8.82.2.1 TrainingWordsRecognizedMessageResponse()

```
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse::TrainingWordsRecognizedMessageResponse ( ) [inline]
```

### 8.82.2.2 ~TrainingWordsRecognizedMessageResponse()

```
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse::~TrainingWordsRecognizedMessageResponse ( ) [inline]
```

## 8.82.3 类成员变量说明

### 8.82.3.1 response

```
MsgTrainingWords cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse::response
```

反馈

### 8.82.3.2 state

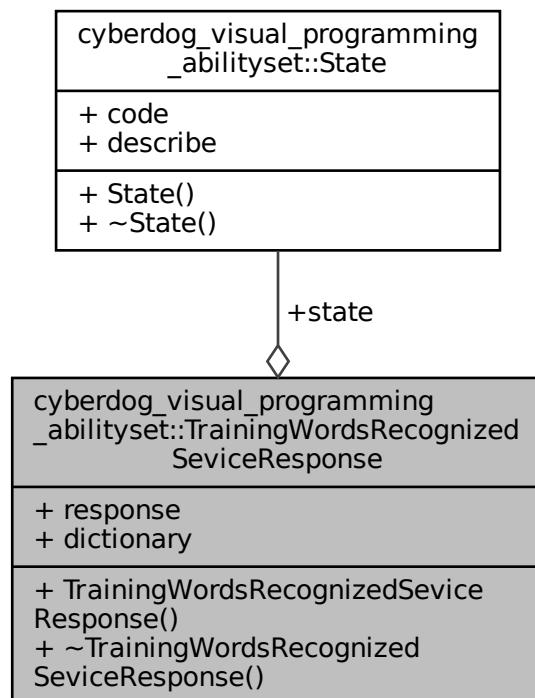
```
State cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse::state
```

状态

## 8.83 cyberdog\_visual\_programming\_abilityset::TrainingWordsRecognizedSeviceResponse类参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::TrainingWordsRecognizedSeviceResponse 的协作图:



### Public 成员函数

- `TrainingWordsRecognizedSeviceResponse ()`
- `~TrainingWordsRecognizedSeviceResponse ()`

### Public 属性

- `State state`
- `SrvTrainingWords::Response response`
- `std::map< std::string, MsgTrainingWords > dictionary`

#### 8.83.1 详细描述

训练词识别服务反馈

## 8.83.2 构造及析构函数说明

### 8.83.2.1 TrainingWordsRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse::TrainingWordsRecognizedSeviceResponse ( ) [inline]
```

### 8.83.2.2 ~TrainingWordsRecognizedSeviceResponse()

```
cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse::~TrainingWordsRecognizedSeviceResponse ( ) [inline]
```

## 8.83.3 类成员变量说明

### 8.83.3.1 dictionary

```
std::map<std::string, MsgTrainingWords> cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse::dictionary
```

字典

### 8.83.3.2 response

```
SrvTrainingWords::Response cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse::response
```

反馈

### 8.83.3.3 state

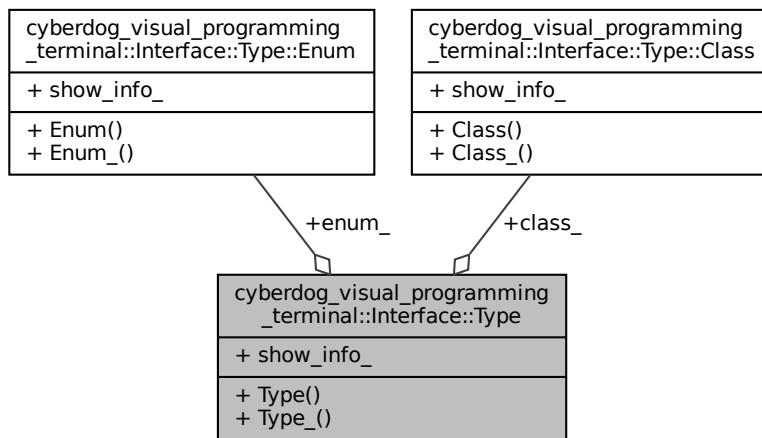
```
State cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse::state
```

状态

## 8.84 cyberdog\_visual\_programming\_terminal::Interface::Type 结构体 参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Type 的协作图:



### 类

- struct [Class](#)
- struct [Enum](#)

### Public 成员函数

- [Type \(\)](#)
- void [Type\\_](#)(const std::string \_fun)

### Public 属性

- std::function< void(std::string)> [show\\_info\\_](#)
- struct [cyberdog\\_visual\\_programming\\_terminal::Interface::Type::Enum](#) [enum\\_](#)
- struct [cyberdog\\_visual\\_programming\\_terminal::Interface::Type::Class](#) [class\\_](#)

#### 8.84.1 构造及析构函数说明

### 8.84.1.1 Type()

```
cyberdog_visual_programming_terminal::Interface::Type::Type ( ) [inline]
```

## 8.84.2 成员函数说明

### 8.84.2.1 Type\_()

```
void cyberdog_visual_programming_terminal::Interface::Type::Type_ (
    const std::string &fun ) [inline]
```

## 8.84.3 类成员变量说明

### 8.84.3.1 class\_

```
struct cyberdog_visual_programming_terminal::Interface::Type::Class cyberdog_visual_programming←
terminal::Interface::Type::class_
```

### 8.84.3.2 enum\_

```
struct cyberdog_visual_programming_terminal::Interface::Type::Enum cyberdog_visual_programming←
terminal::Interface::Type::enum_
```

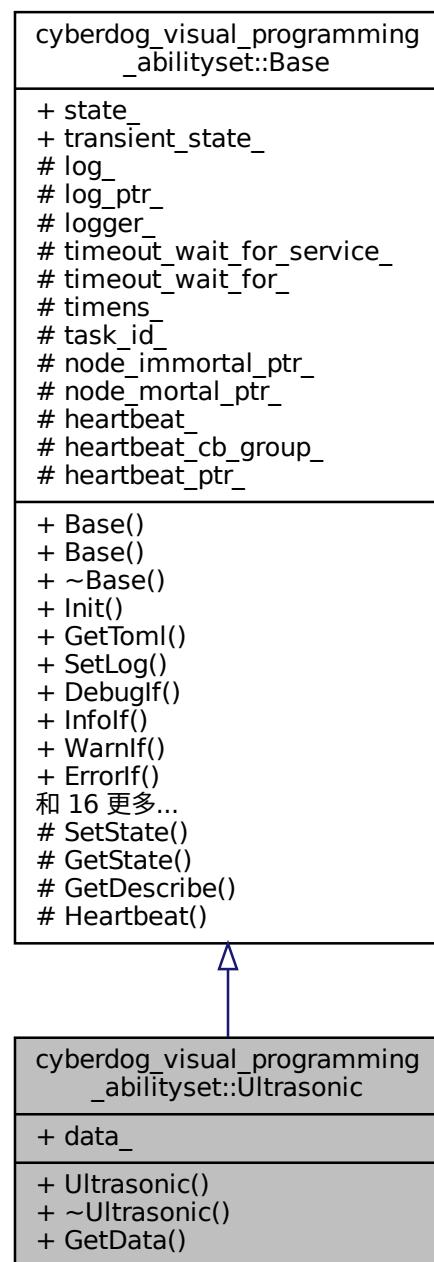
### 8.84.3.3 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Type::show←
info_
```

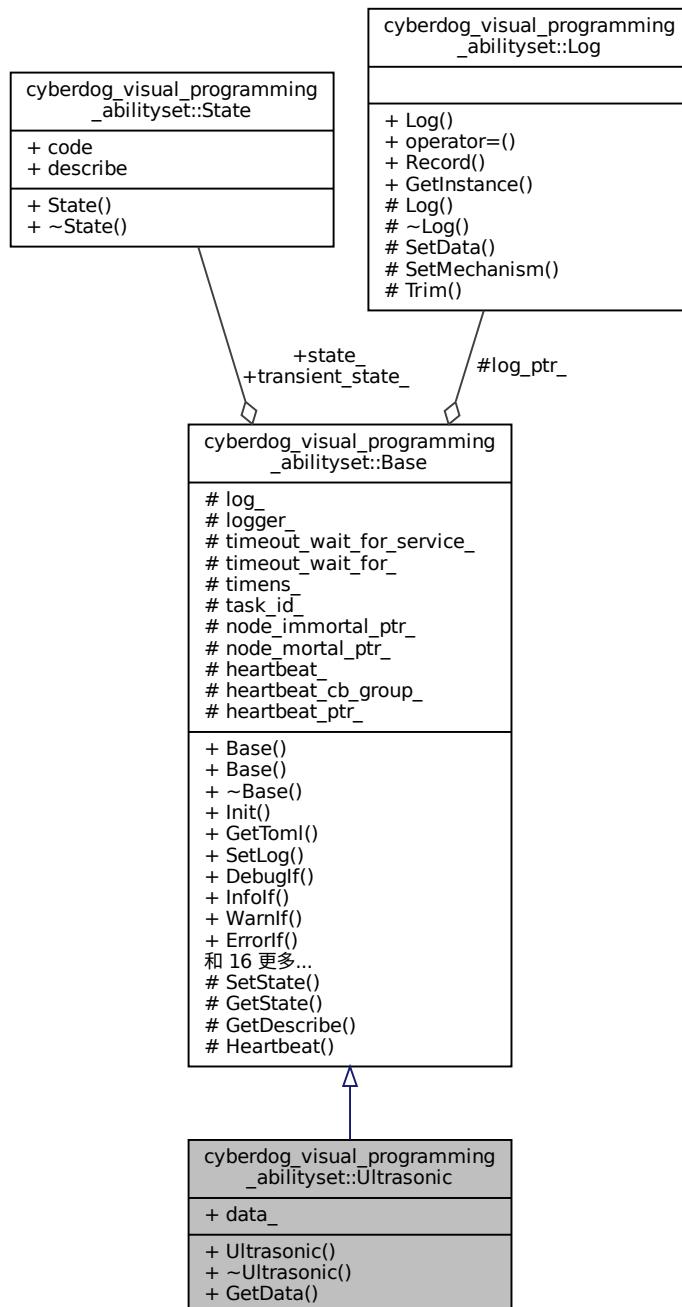
## 8.85 cyberdog\_visual\_programming\_abilityset::Ultrasonic类 参考

```
#include <ultrasonic.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Ultrasonic 继承关系图:



cyberdog\_visual\_programming\_abilityset::Ultrasonic 的协作图:



## Public 成员函数

- `Ultrasonic ()`
- `~Ultrasonic ()`
- `MsgRange GetData (const int _timeout=5)`

## Public 属性

- [MsgRange data\\_](#)

额外继承的成员函数

### 8.85.1 构造及析构函数说明

#### 8.85.1.1 Ultrasonic()

```
cyberdog_visual_programming_abilityset::Ultrasonic::Ultrasonic ( ) [inline]
```

#### 8.85.1.2 ~Ultrasonic()

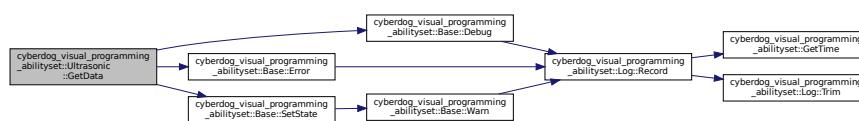
```
cyberdog_visual_programming_abilityset::Ultrasonic::~Ultrasonic ( ) [inline]
```

### 8.85.2 成员函数说明

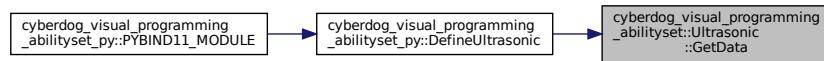
#### 8.85.2.1 GetData()

```
MsgRange cyberdog_visual_programming_abilityset::Ultrasonic::GetData (
    const int _timeout = 5 )
```

获取最新数据 函数调用图:



这是这个函数的调用关系图:



### 8.85.3 类成员变量说明

#### 8.85.3.1 `data_`

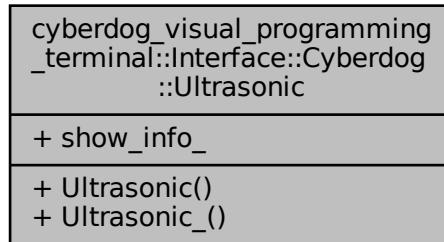
`MsgRange cyberdog_visual_programming_abilityset::Ultrasonic::data_`

数据

## 8.86 `cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic` 结构体 参考

```
#include <interface.hpp>
```

`cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic` 的协作图:



### Public 成员函数

- `Ultrasonic ()`
- `void Ultrasonic_(const std::string _fun)`

### Public 属性

- `std::function< void(std::string)> show_info_`

### 8.86.1 构造及析构函数说明

### 8.86.1.1 Ultrasonic()

```
cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic::Ultrasonic() [inline]
```

## 8.86.2 成员函数说明

### 8.86.2.1 Ultrasonic\_()

```
void cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic::Ultrasonic_(
    const std::string &fun) [inline]
```

## 8.86.3 类成员变量说明

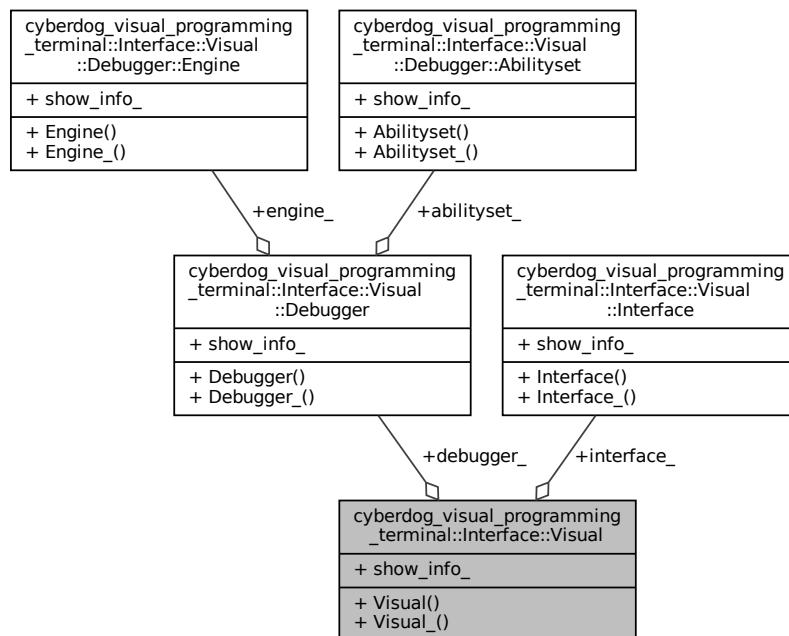
### 8.86.3.1 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Cyberdog::show_info_;
```

## 8.87 cyberdog\_visual\_programming\_terminal::Interface::Visual结构体参考

```
#include <interface.hpp>
```

cyberdog\_visual\_programming\_terminal::Interface::Visual 的协作图:



## 类

- struct `Debugger`
- struct `Interface`

## Public 成员函数

- `Visual()`
- void `Visual_ (const std::string &fun)`

## Public 属性

- std::function< void(std::string) > `show_info_`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Interface interface_`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Debugger debugger_`

### 8.87.1 构造及析构函数说明

#### 8.87.1.1 `Visual()`

```
cyberdog_visual_programming_terminal::Interface::Visual::Visual () [inline]
```

### 8.87.2 成员函数说明

#### 8.87.2.1 `Visual_()`

```
void cyberdog_visual_programming_terminal::Interface::Visual::Visual_ (
    const std::string &fun ) [inline]
```

### 8.87.3 类成员变量说明

#### 8.87.3.1 `debugger_`

```
struct cyberdog_visual_programming_terminal::Interface::Visual::Debugger cyberdog_visual_programming_terminal::Interface::Visual::debugger_
```

### 8.87.3.2 interface\_

```
struct cyberdog_visual_programming_terminal::Interface::Visual::Interface cyberdog_visual_<-
programming_terminal::Interface::Visual::interface_
```

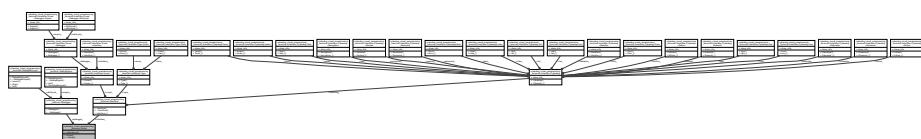
### 8.87.3.3 show\_info\_

```
std::function<void(std::string)> cyberdog_visual_programming_terminal::Interface::Visual_<-
::show_info_
```

## 8.88 cyberdog\_visual\_programming\_terminal::Visual类 参考

```
#include <visual.hpp>
```

cyberdog\_visual\_programming\_terminal::Visual 的协作图:



### Public 成员函数

- `Visual (std::string _task="visual", std::string _namespace="", bool _ros=false, std::string _parameters "")`
- `~Visual ()`

### Public 属性

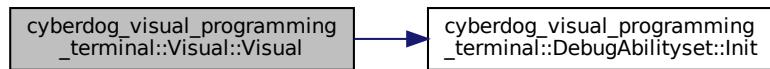
- `std::shared_ptr< VPA::Cyberdog > cyberdog_ptr_ {nullptr}`
- `Interface interface_`
- `Debugger debugger_`

### 8.88.1 构造及析构函数说明

### 8.88.1.1 Visual()

```
cyberdog_visual_programming_terminal::Visual::Visual (
    std::string _task = "visual",
    std::string _namespace = "",
    bool _ros = false,
    std::string _parameters = "" )
```

函数调用图:



### 8.88.1.2 ~Visual()

```
cyberdog_visual_programming_terminal::Visual::~Visual ( )
```

## 8.88.2 类成员变量说明

### 8.88.2.1 cyberdog\_ptr\_

```
std::shared_ptr<VPA::Cyberdog> cyberdog_visual_programming_terminal::Visual::cyberdog_ptr_<->
{nullptr}
```

铁蛋

### 8.88.2.2 debugger\_

```
Debugger cyberdog_visual_programming_terminal::Visual::debugger_
```

调试模块

### 8.88.2.3 interface\_

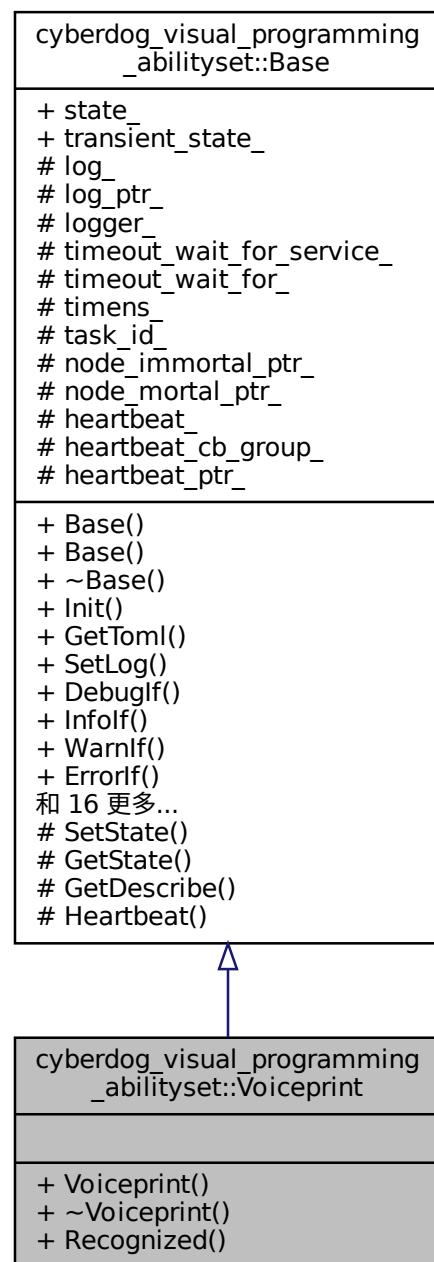
```
Interface cyberdog_visual_programming_terminal::Visual::interface_
```

接口模块

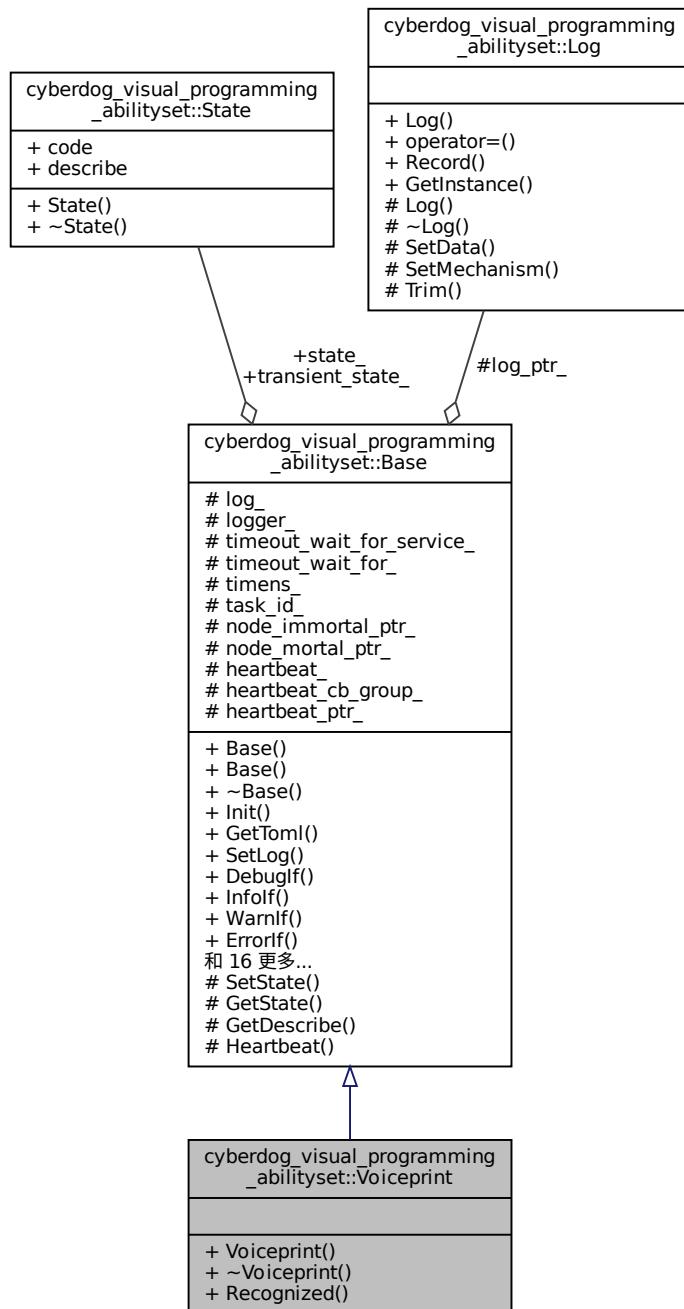
## 8.89 cyberdog\_visual\_programming\_abilityset::Voiceprint类 参考

```
#include <voiceprint.hpp>
```

类 cyberdog\_visual\_programming\_abilityset::Voiceprint 继承关系图:



cyberdog\_visual\_programming\_abilityset::Voiceprint 的协作图:



## Public 成员函数

- [Voiceprint \(\)](#)
- [~Voiceprint \(\)](#)
- [VoiceprintRecognizedResponse Recognized \(const int \\_duration=-1, const int \\_sensitivity=1\)](#)

## 额外继承的成员函数

### 8.89.1 构造及析构函数说明

#### 8.89.1.1 Voiceprint()

```
cyberdog_visual_programming_abilityset::Voiceprint::Voiceprint ( ) [inline]
```

#### 8.89.1.2 ~Voiceprint()

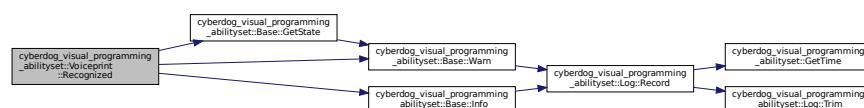
```
cyberdog_visual_programming_abilityset::Voiceprint::~Voiceprint ( ) [inline]
```

### 8.89.2 成员函数说明

#### 8.89.2.1 Recognized()

```
VoiceprintRecognizedResponse cyberdog_visual_programming_abilityset::Voiceprint::Recognized (
    const int _duration = -1,
    const int _sensitivity = 1 )
```

识别到声纹 函数调用图:



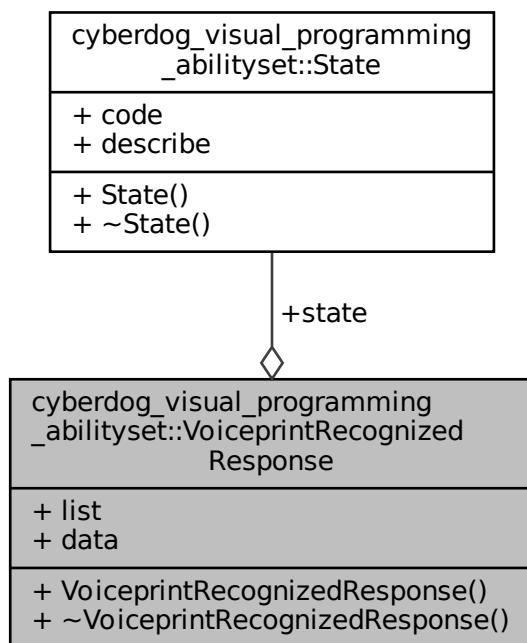
这是这个函数的调用关系图:



## 8.90 cyberdog\_visual\_programming\_abilityset::VoiceprintRecognizedResponse类 参考

```
#include <common.hpp>
```

cyberdog\_visual\_programming\_abilityset::VoiceprintRecognizedResponse 的协作图:



### Public 成员函数

- `VoiceprintRecognizedResponse ()`
- `~VoiceprintRecognizedResponse ()`

### Public 属性

- `State state`
- `std::vector< std::string > list`
- `bool data`

#### 8.90.1 详细描述

声纹识别返回信息

## 8.90.2 构造及析构函数说明

### 8.90.2.1 VoiceprintRecognizedResponse()

```
cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse::VoiceprintRecognizedResponse ( ) [inline]
```

### 8.90.2.2 ~VoiceprintRecognizedResponse()

```
cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse::~VoiceprintRecognizedResponse ( ) [inline]
```

## 8.90.3 类成员变量说明

### 8.90.3.1 data

```
bool cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse::data
```

数据

### 8.90.3.2 list

```
std::vector<std::string> cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse::list
```

反馈

### 8.90.3.3 state

```
State cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse::state
```

状态

# Chapter 9

## 文件说明

### 9.1 cyberdog\_vp/cyberdog\_vp/doc/doxygen/index.md 文件参考

### 9.2 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/audio.hpp 文件参考

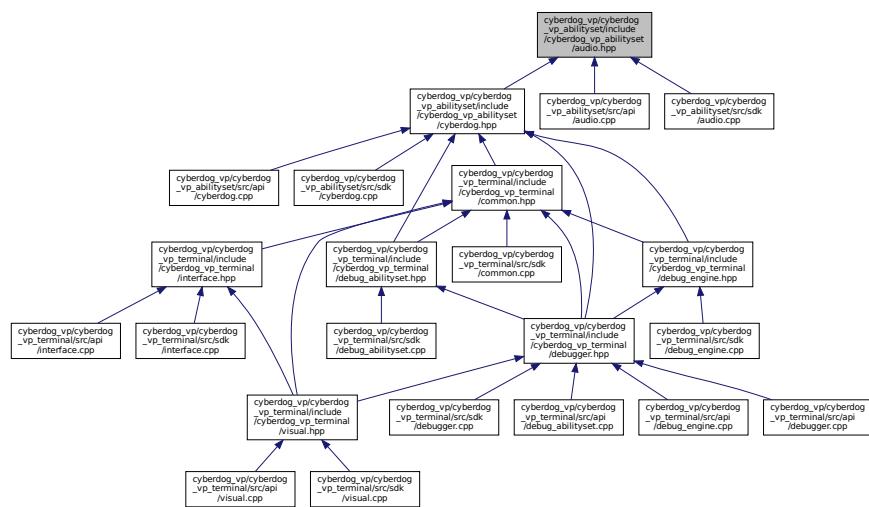
语音模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

audio.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_abilityset::Audio`

命名空间

- `cyberdog_visual_programming_abilityset`

### 9.2.1 详细描述

语音模块。

创建及初始化语音模块，以便任务调用语音功能。

作者

尚子涵

`Shang Zihan`

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.3 `cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/base.hpp` 文件参考

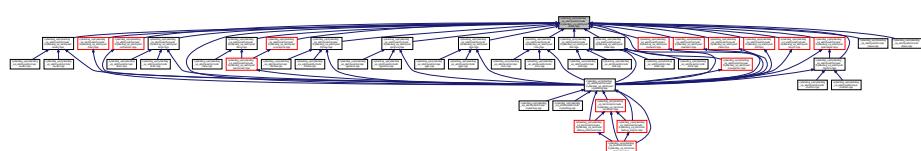
可视化编程SDK的基础模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/log.hpp"
```

base.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Base](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.3.1 详细描述

可视化编程SDK的基础模块。

创建及初始化SDK模块的通用逻辑。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化铁蛋模块。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.4 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/base.hpp 文件参考

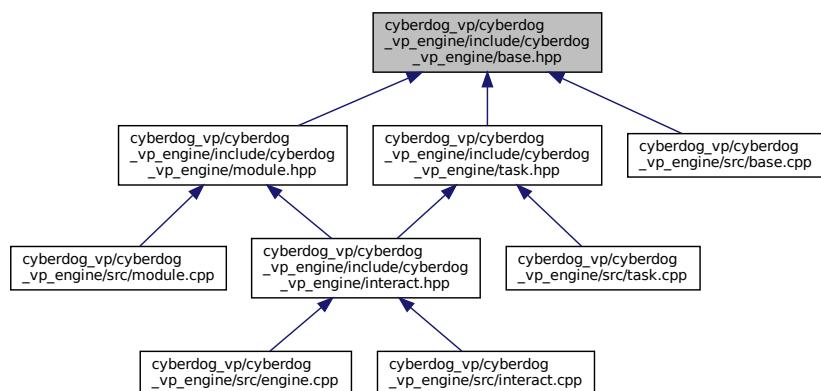
可视化编程引擎的基础模块。

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/common.hpp"
#include "cyberdog_vp_engine/frontend_message.hpp"
#include "cyberdog_vp_engine/python_interpreter.hpp"
```

base.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_engine::Base](#)

命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.4.1 详细描述

可视化编程引擎的基础模块。

创建及初始化“任务编程”和“模块编程”模块的通用逻辑。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化任务编程或模块编程模块。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.5 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/bms.hpp 文件参考

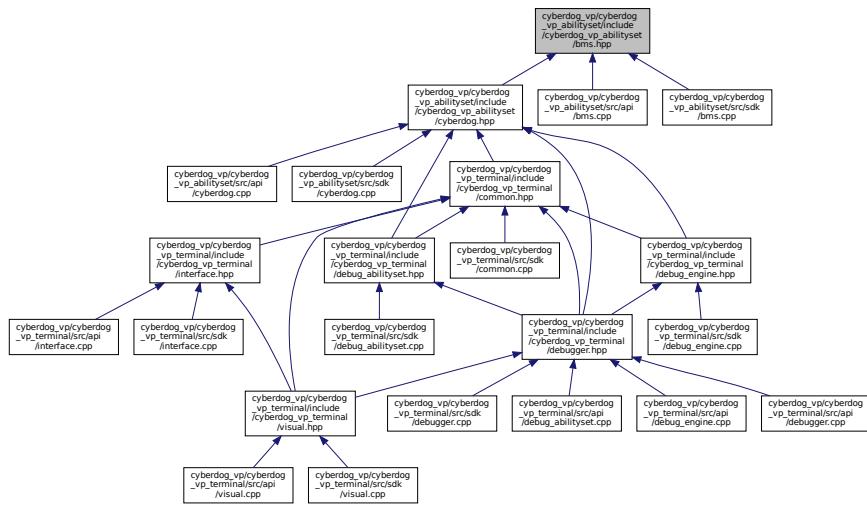
电池管理系统模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

bms.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Bms](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.5.1 详细描述

电池管理系统模块。

创建及初始化电池管理系统模块，以便任务调用电池管理系统功能。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化设备。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

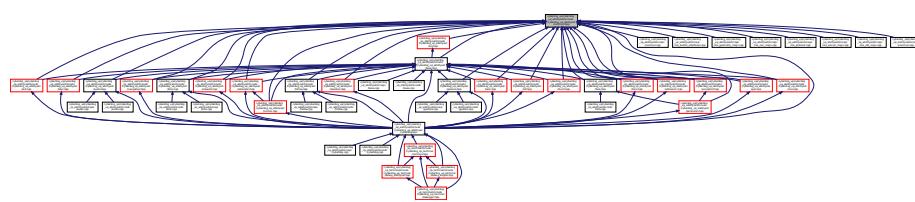
## 9.6 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/common.hpp 文件参考

可视化编程参数模块。

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <stdio.h>
#include <stdarg.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/time.h>
#include <rclcpp/rclcpp.hpp>
#include <rclcpp/wait_set.hpp>
#include <rclcpp_action/rclcpp_action.hpp>
#include <rclcpp_components/register_node_macro.hpp>
#include <ament_index_cpp/get_package_share_directory.hpp>
#include <ament_index_cpp/get_package_prefix.hpp>
#include <cyberdog_common/cyberdog_log.hpp>
#include <cyberdog_common/cyberdog_toml.hpp>
#include <cyberdog_common/cyberdog_json.hpp>
#include <rapidjson/document.h>
#include <rapidjson/prettywriter.h>
#include <rapidjson/stringbuffer.h>
#include <std_msgs/msg/header.hpp>
#include <std_msgs/msg/string.hpp>
#include <std_msgs/msg/bool.hpp>
#include <sensor_msgs/msg/imu.hpp>
#include <sensor_msgs/msg/range.hpp>
#include <sensor_msgs/msg/laser_scan.hpp>
#include <tf2_msgs/msg/tf_message.hpp>
#include <tf2/LinearMath/Quaternion.h>
#include <tf2_geometry_msgs/tf2_geometry_msgs.h>
#include <nav_msgs/msg/odometry.hpp>
#include <geometry_msgs/msg/twist.hpp>
#include <geometry_msgs/msg/point.hpp>
#include <geometry_msgs/msg/pose.hpp>
#include <geometry_msgs/msg/poseStamped.hpp>
#include <geometry_msgs/msg/quaternion.hpp>
#include <geometry_msgs/msg/vector3.hpp>
#include <std_srvs/srv/set_bool.hpp>
#include <protocol/msg/train_plan.hpp>
#include <protocol/msg/user_information.hpp>
#include <protocol/msg/face_recognition_result.hpp>
#include <protocol/msg/gesture_action_result.hpp>
#include <protocol/msg/motion_sequence_pace.hpp>
#include <protocol/msg/audio_play_extend.hpp>
#include <protocol/msg/audio_play.hpp>
#include <protocol/msg/connector_status.hpp>
#include <protocol/msg/motion_id.hpp>
#include <protocol/msg/bms_status.hpp>
#include <protocol/msg/touch_status.hpp>
#include <protocol/msg/head_tof_payload.hpp>
#include <protocol/msg/rear_tof_payload.hpp>
#include <protocol/msg/gps_payload.hpp>
```

```
#include <protocol/msg/single_to_f_payload.hpp>
#include <protocol/msg/motion_servo_cmd.hpp>
#include <protocol/msg/motion_servo_response.hpp>
#include <protocol/msg/motion_status.hpp>
#include <protocol/msg/label.hpp>
#include <protocol/msg/sport_counts_result.hpp>
#include <protocol/msg/algo_task_status.hpp>
#include <protocol/msg/visual_programming_operate.hpp>
#include <protocol/srv/visual_programming_operate.hpp>
#include <protocol/srv/audio_text_play.hpp>
#include <protocol/srv/audio_volume_get.hpp>
#include <protocol/srv/audio_volume_set.hpp>
#include <protocol/srv/led_execute.hpp>
#include <protocol/srv/motion_result_cmd.hpp>
#include <protocol/srv/motion_sequence_show.hpp>
#include <protocol/srv/all_user_search.hpp>
#include <protocol/srv/face_rec.hpp>
#include <protocol/srv/gesture_action_control.hpp>
#include <protocol/srv/get_map_label.hpp>
#include <protocol/srv/stop_algo_task.hpp>
#include <protocol/srv/sport_manager.hpp>
#include <protocol/srv/train_plan_all.hpp>
#include <protocol/action/navigation.hpp>
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <vector>
#include <memory>
#include <map>
#include <unordered_map>
#include <mutex>
#include <tuple>
#include <regex>
#include <atomic>
#include <algorithm>
#include <cmath>
#include <condition_variable>
#include <thread>
#include <chrono>
```

此图展示该文件直接或间接的被哪些文件引用了：



## 类

- struct [cyberdog\\_visual\\_programming\\_abilityset::DefaultAndMaximum](#)
- struct [cyberdog\\_visual\\_programming\\_abilityset::MotionParams](#)
- struct [cyberdog\\_visual\\_programming\\_abilityset::RPY](#)
- struct [cyberdog\\_visual\\_programming\\_abilityset::ObstacleMeta](#)

- struct `cyberdog_visual_programming_abilityset::TofObstacle`
- class `cyberdog_visual_programming_abilityset::TofPayload`
- class `cyberdog_visual_programming_abilityset::State`
- class `cyberdog_visual_programming_abilityset::MotionResultServiceResponse`
- class `cyberdog_visual_programming_abilityset::MotionSequenceServiceResponse`
- class `cyberdog_visual_programming_abilityset::MotionServoCmdResponse`
- class `cyberdog_visual_programming_abilityset::AudioPlaySeviceResponse`
- class `cyberdog_visual_programming_abilityset::AudioGetVolumeSeviceResponse`
- class `cyberdog_visual_programming_abilityset::AudioSetVolumeSeviceResponse`
- class `cyberdog_visual_programming_abilityset::LedSeviceResponse`
- class `cyberdog_visual_programming_abilityset::FaceSeviceResponse`
- class `cyberdog_visual_programming_abilityset::FaceRecognizedSeviceResponse`
- class `cyberdog_visual_programming_abilityset::VoiceprintRecognizedResponse`
- struct `cyberdog_visual_programming_abilityset::GestureData`
- class `cyberdog_visual_programming_abilityset::GestureRecognizedSeviceResponse`
- class `cyberdog_visual_programming_abilityset::GestureRecognizedMessageResponse`
- class `cyberdog_visual_programming_abilityset::SkeletonRecognizedSeviceResponse`
- class `cyberdog_visual_programming_abilityset::SkeletonRecognizedMessageResponse`
- class `cyberdog_visual_programming_abilityset::TrainingWordsRecognizedSeviceResponse`
- class `cyberdog_visual_programming_abilityset::TrainingWordsRecognizedMessageResponse`
- class `cyberdog_visual_programming_abilityset::MapPresetSeviceResponse`
- class `cyberdog_visual_programming_abilityset::NavigationActionResponse`
- class `cyberdog_visual_programming_abilityset::MotionSequence`

## 命名空间

- `cyberdog_visual_programming_abilityset`

## 宏定义

- `#define STREAM(msg)`
- `#define FORMAT(format, ...)`
- `#define ONE_FORMAT(format)`

## 类型定义

- using `cyberdog_visual_programming_abilityset::CyberdogJson` = `cyberdog::common::CyberdogJson`
- using `cyberdog_visual_programming_abilityset::CyberdogToml` = `cyberdog::common::CyberdogToml`
- using `cyberdog_visual_programming_abilityset::MsgHeader` = `std_msgs::msg::Header`
- using `cyberdog_visual_programming_abilityset::MsgString` = `std_msgs::msg::String`
- using `cyberdog_visual_programming_abilityset::MsgBool` = `std_msgs::msg::Bool`
- using `cyberdog_visual_programming_abilityset::MsgLaserScan` = `sensor_msgs::msg::LaserScan`
- using `cyberdog_visual_programming_abilityset::MsgImu` = `sensor_msgs::msg::Imu`
- using `cyberdog_visual_programming_abilityset::MsgRange` = `sensor_msgs::msg::Range`
- using `cyberdog_visual_programming_abilityset::MsgOdometry` = `nav_msgs::msg::Odometry`
- using `cyberdog_visual_programming_abilityset::MsgPose` = `geometry_msgs::msg::Pose`
- using `cyberdog_visual_programming_abilityset::MsgPoseStamped` = `geometry_msgs::msg::PoseStamped`
- using `cyberdog_visual_programming_abilityset::MsgPoint` = `geometry_msgs::msg::Point`
- using `cyberdog_visual_programming_abilityset::MsgMotionID` = `protocol::msg::MotionID`
- using `cyberdog_visual_programming_abilityset::MsgMotionServoCmd` = `protocol::msg::MotionServoCmd`
- using `cyberdog_visual_programming_abilityset::MsgMotionStatus` = `protocol::msg::MotionStatus`
- using `cyberdog_visual_programming_abilityset::MsgSingleTofPayload` = `protocol::msg::SingleTofPayload`

- using `cyberdog_visual_programming_abilityset::MsgHeadToPayload` = protocol::msg::HeadToPayload
- using `cyberdog_visual_programming_abilityset::MsgRearToPayload` = protocol::msg::RearToPayload
- using `cyberdog_visual_programming_abilityset::MsgGpsPayload` = protocol::msg::GpsPayload
- using `cyberdog_visual_programming_abilityset::MsgAudioPlayExtend` = protocol::msg::AudioPlayExtend
- using `cyberdog_visual_programming_abilityset::MsgAudioPlay` = protocol::msg::AudioPlay
- using `cyberdog_visual_programming_abilityset::MsgConnectorStatus` = protocol::msg::ConnectorStatus
- using `cyberdog_visual_programming_abilityset::MsgMotionServoResponse` = protocol::msg::MotionServoResponse
- using `cyberdog_visual_programming_abilityset::MsgMotionSequenceGait` = protocol::msg::MotionSequenceGait
- using `cyberdog_visual_programming_abilityset::MsgMotionSequencePace` = protocol::msg::MotionSequencePace
- using `cyberdog_visual_programming_abilityset::MsgTouchStatus` = protocol::msg::TouchStatus
- using `cyberdog_visual_programming_abilityset::MsgBmsStatus` = protocol::msg::BmsStatus
- using `cyberdog_visual_programming_abilityset::MsgPersonnel` = protocol::msg::UserInformation
- using `cyberdog_visual_programming_abilityset::MsgFaceRes` = protocol::msg::FaceRecognitionResult
- using `cyberdog_visual_programming_abilityset::MsgGesture` = protocol::msg::GestureActionResult
- using `cyberdog_visual_programming_abilityset::MsgSport` = protocol::msg::SportCountsResult
- using `cyberdog_visual_programming_abilityset::MsgPreset` = protocol::msg::Label
- using `cyberdog_visual_programming_abilityset::MsgTrainingWords` = protocol::msg::TrainPlan
- using `cyberdog_visual_programming_abilityset::MsgAlgoStatus` = protocol::msg::AlgoTaskStatus
- using `cyberdog_visual_programming_abilityset::MsgVisualProgrammingOperate` = protocol::msg::VisualProgrammingOperate
- using `cyberdog_visual_programming_abilityset::SrvVisualProgrammingOperate` = protocol::srv::VisualProgrammingOperate
- using `cyberdog_visual_programming_abilityset::SrvTrainingWords` = protocol::srv::TrainPlanAll
- using `cyberdog_visual_programming_abilityset::SrvPersonnel` = protocol::srv::AllUserSearch
- using `cyberdog_visual_programming_abilityset::SrvFaceRec` = protocol::srv::FaceRec
- using `cyberdog_visual_programming_abilityset::SrvGesture` = protocol::srv::GestureActionControl
- using `cyberdog_visual_programming_abilityset::SrvSport` = protocol::srv::SportManager
- using `cyberdog_visual_programming_abilityset::SrvMotionSequenceShow` = protocol::srv::MotionSequenceShow
- using `cyberdog_visual_programming_abilityset::SrvMotionResultCmd` = protocol::srv::MotionResultCmd
- using `cyberdog_visual_programming_abilityset::SrvLedExecute` = protocol::srv::LedExecute
- using `cyberdog_visual_programming_abilityset::SrvAudioTextPlay` = protocol::srv::AudioTextPlay
- using `cyberdog_visual_programming_abilityset::SrvAudioGetVolume` = protocol::srv::AudioVolumeGet
- using `cyberdog_visual_programming_abilityset::SrvAudioSetVolume` = protocol::srv::AudioVolumeSet
- using `cyberdog_visual_programming_abilityset::SrvGetPreset` = protocol::srv::GetMapLabel
- using `cyberdog_visual_programming_abilityset::SrvCancelNavigation` = protocol::srv::StopAlgoTask
- using `cyberdog_visual_programming_abilityset::SrvSetBool` = std::srvs::srv::SetBool
- using `cyberdog_visual_programming_abilityset::ActNavigation` = protocol::action::Navigation

## 枚举

- enum `cyberdog_visual_programming_abilityset::Processor` { `cyberdog_visual_programming_abilityset::task` = 0 }
- enum `cyberdog_visual_programming_abilityset::MotionId` {
 `cyberdog_visual_programming_abilityset::emergency_stop` = 0, `cyberdog_visual_programming_abilityset::get_down` = 101, `cyberdog_visual_programming_abilityset::resume_standing` = 111, `cyberdog_visual_programming_abilityset::servo_standing` = 112,
 `cyberdog_visual_programming_abilityset::back_flip` = 121, `cyberdog_visual_programming_abilityset::front_flip` = 122, `cyberdog_visual_programming_abilityset::bow` = 123, `cyberdog_visual_programming_abilityset::roll_left` = 124,
 `cyberdog_visual_programming_abilityset::walk_the_dog` = 125, `cyberdog_visual_programming_abilityset::jump_stair`

```
= 126, cyberdog_visual_programming_abilityset::right_somersault = 127, cyberdog_visual_programming_abilityset::left_somersault = 128,
cyberdog_visual_programming_abilityset::run_and_jump_front_flip = 129, cyberdog_visual_programming_abilityset::jump3d_left90deg = 130, cyberdog_visual_programming_abilityset::jump3d_right90deg = 131, cyberdog_visual_programming_abilityset::jump3d_forward = 132,
cyberdog_visual_programming_abilityset::jump3d_forward30cm = 133, cyberdog_visual_programming_abilityset::jump3d_left20cm = 134, cyberdog_visual_programming_abilityset::jump3d_right20cm = 135, cyberdog_visual_programming_abilityset::jump3d_up30cm = 136,
cyberdog_visual_programming_abilityset::jump3d_down_stair = 137, cyberdog_visual_programming_abilityset::roll_right = 138, cyberdog_visual_programming_abilityset::dance_collection = 140, cyberdog_visual_programming_abilityset::hold_left_hand = 141,
cyberdog_visual_programming_abilityset::hold_right_hand = 142, cyberdog_visual_programming_abilityset::sit_down = 143, cyberdog_visual_programming_abilityset::butt_circle = 144, cyberdog_visual_programming_abilityset::head_circle = 145,
cyberdog_visual_programming_abilityset::stretch_the_body = 146, cyberdog_visual_programming_abilityset::shake_ass_left = 148, cyberdog_visual_programming_abilityset::shake_ass_right = 149, cyberdog_visual_programming_abilityset::shake_ass_front = 150,
cyberdog_visual_programming_abilityset::ballet = 151, cyberdog_visual_programming_abilityset::space_walk = 152, cyberdog_visual_programming_abilityset::front_leg_jumping = 153, cyberdog_visual_programming_abilityset::hind_leg_jumping = 154,
cyberdog_visual_programming_abilityset::lift_the_left_leg_and_nod = 155, cyberdog_visual_programming_abilityset::lift_the_right_leg_and_nod = 156, cyberdog_visual_programming_abilityset::left_front_right_back_legs_apart = 157, cyberdog_visual_programming_abilityset::left_right_back_legs_apart = 158,
cyberdog_visual_programming_abilityset::walk_nodding = 159, cyberdog_visual_programming_abilityset::walking_with_divergence = 160, cyberdog_visual_programming_abilityset::nodding_in_place = 161, cyberdog_visual_programming_abilityset::front_legs_jump_back_and_forth = 162,
cyberdog_visual_programming_abilityset::hind_legs_jump_back_and_forth = 163, cyberdog_visual_programming_abilityset::alternately_hind_leg_lift = 164, cyberdog_visual_programming_abilityset::jump_cold_feet = 165, cyberdog_visual_programming_abilityset::jump_cold_feet_and_nod = 166,
cyberdog_visual_programming_abilityset::stretching_left_and_right = 167, cyberdog_visual_programming_abilityset::jump_forward_and_backward = 168, cyberdog_visual_programming_abilityset::step_left_and_right = 169, cyberdog_visual_programming_abilityset::right_leg_backward_and_left = 170,
cyberdog_visual_programming_abilityset::left_leg_back_and_forth_stepping = 171, cyberdog_visual_programming_abilityset::squats = 172, cyberdog_visual_programming_abilityset::push_ups = 174, cyberdog_visual_programming_abilityset::bow_to_each_other = 175,
cyberdog_visual_programming_abilityset::absolute_force_control_attitude = 201, cyberdog_visual_programming_abilityset::relative_force_control_attitude = 202, cyberdog_visual_programming_abilityset::absolute_position_control_attitude = 211, cyberdog_visual_programming_abilityset::relative_position_control_attitude = 212,
cyberdog_visual_programming_abilityset::relatively_position_control_attitude_insert_frame_1 = 213, cyberdog_visual_programming_abilityset::small_jump_walking = 214, cyberdog_visual_programming_abilityset::jump_back_and_forth = 301, cyberdog_visual_programming_abilityset::small_jump_walking_and_nodding = 302,
cyberdog_visual_programming_abilityset::trot_walking = 303, cyberdog_visual_programming_abilityset::automatic_frequency_conversion = 304, cyberdog_visual_programming_abilityset::run_fast_walking = 305, cyberdog_visual_programming_abilityset::sequence_custom = 400,
cyberdog_visual_programming_abilityset::illegal_motion_id = 999 }
```

- enum cyberdog\_visual\_programming\_abilityset::StateCode {  
 cyberdog\_visual\_programming\_abilityset::invalid = -1, cyberdog\_visual\_programming\_abilityset::success = 0,  
 cyberdog\_visual\_programming\_abilityset::error\_base = 5800, cyberdog\_visual\_programming\_abilityset::fail = 5801,  
 cyberdog\_visual\_programming\_abilityset::uninitialized = 5802, cyberdog\_visual\_programming\_abilityset::fsm\_does\_not\_allow = 5803, cyberdog\_visual\_programming\_abilityset::module\_status\_error = 5804, cyberdog\_visual\_programming\_abilityset::network\_error = 5805,  
 cyberdog\_visual\_programming\_abilityset::no\_operation\_authority = 5806, cyberdog\_visual\_programming\_abilityset::timeout = 5807, cyberdog\_visual\_programming\_abilityset::command\_does\_not\_support = 5808, cyberdog\_visual\_programming\_abilityset::parameter\_is\_invalid = 5809,  
 cyberdog\_visual\_programming\_abilityset::parameter\_is\_invalid = 5810, cyberdog\_visual\_programming\_abilityset::status\_is\_busy = 5811, cyberdog\_visual\_programming\_abilityset::hardware\_error = 5812, cyberdog\_visual\_programming\_abilityset::command\_error = 5821,

```

cyberdog_visual_programming_abilityset::spin_future_interrupted = 5831, cyberdog_visual_programming_abilityset::spin_future_time_out = 5832, cyberdog_visual_programming_abilityset::no_data_update = 5841, cyberdog_visual_programming_abilityset::service_client_error = 5851,
cyberdog_visual_programming_abilityset::service_appear_timeout = 5852, cyberdog_visual_programming_abilityset::service_request_rejected = 5853, cyberdog_visual_programming_abilityset::service_request_rejected = 5854, cyberdog_visual_programming_abilityset::service_request_rejected = 5855,
cyberdog_visual_programming_abilityset::action_request_timeout = 5861, cyberdog_visual_programming_abilityset::action_request_timeout = 5862, cyberdog_visual_programming_abilityset::action_result_timeout = 5863 }

• enum cyberdog_visual_programming_abilityset::RPYType { cyberdog_visual_programming_abilityset::ROLL = 0, cyberdog_visual_programming_abilityset::PITCH, cyberdog_visual_programming_abilityset::YAW }

• enum cyberdog_visual_programming_abilityset::LedConstraint {
    cyberdog_visual_programming_abilityset::target_head, cyberdog_visual_programming_abilityset::target_tail,
    cyberdog_visual_programming_abilityset::target_mini, cyberdog_visual_programming_abilityset::effect_line_on,
    cyberdog_visual_programming_abilityset::effect_line_blink, cyberdog_visual_programming_abilityset::effect_line_blink_fast,
    cyberdog_visual_programming_abilityset::effect_line_breath, cyberdog_visual_programming_abilityset::effect_line_breath_fast,
    cyberdog_visual_programming_abilityset::effect_line_one_by_one, cyberdog_visual_programming_abilityset::effect_line_one_by_one,
    cyberdog_visual_programming_abilityset::effect_line_back_and_forth, cyberdog_visual_programming_abilityset::effect_line_trailing,
    cyberdog_visual_programming_abilityset::system_effect_line_off, cyberdog_visual_programming_abilityset::system_effect_line_red,
    cyberdog_visual_programming_abilityset::system_effect_line_red_blink, cyberdog_visual_programming_abilityset::system_effect_line_red_breath,
    cyberdog_visual_programming_abilityset::system_effect_line_red_one_by_one, cyberdog_visual_programming_abilityset::system_effect_line_blue,
    cyberdog_visual_programming_abilityset::system_effect_line_blue_on, cyberdog_visual_programming_abilityset::system_effect_line_blue_blink,
    cyberdog_visual_programming_abilityset::system_effect_line_blue_blink_fast, cyberdog_visual_programming_abilityset::system_effect_line_blue_breath,
    cyberdog_visual_programming_abilityset::system_effect_line_blue_one_by_one_fast, cyberdog_visual_programming_abilityset::system_effect_line_yellow,
    cyberdog_visual_programming_abilityset::system_effect_line_yellow_blink, cyberdog_visual_programming_abilityset::system_effect_line_yellow_breath,
    cyberdog_visual_programming_abilityset::system_effect_line_yellow_one_by_one, cyberdog_visual_programming_abilityset::system_effect_mini_circular,
    cyberdog_visual_programming_abilityset::effect_mini_circular_breath, cyberdog_visual_programming_abilityset::effect_mini_circular_blink,
    cyberdog_visual_programming_abilityset::system_effect_mini_off, cyberdog_visual_programming_abilityset::system_effect_mini_centre_color,
    cyberdog_visual_programming_abilityset::system_effect_mini_one_by_one }

• enum cyberdog_visual_programming_abilityset::GestureType {
    cyberdog_visual_programming_abilityset::no_gesture = 0, cyberdog_visual_programming_abilityset::pulling_hand_or_two_fingers_in = 1,
    cyberdog_visual_programming_abilityset::pushing_hand_or_two_fingers_away = 2, cyberdog_visual_programming_abilityset::stop = 3,
    cyberdog_visual_programming_abilityset::sliding_hand_or_two_fingers_down = 4, cyberdog_visual_programming_abilityset::sliding_hand_or_two_fingers_right = 5,
    cyberdog_visual_programming_abilityset::sliding_hand_or_two_fingers_left = 6, cyberdog_visual_programming_abilityset::stop = 7,
    cyberdog_visual_programming_abilityset::thumb_up = 8, cyberdog_visual_programming_abilityset::zooming_in_with_hand_or_two_fingers = 9,
    cyberdog_visual_programming_abilityset::zooming_out_with_hand_or_two_fingers = 10, cyberdog_visual_programming_abilityset::stop = 11,
    cyberdog_visual_programming_abilityset::id_upper_limit = 12 }

• enum cyberdog_visual_programming_abilityset::SkeletonType {
    cyberdog_visual_programming_abilityset::SPORT_SQUAT = 1, cyberdog_visual_programming_abilityset::SPORT_HIGHKNEES,
    cyberdog_visual_programming_abilityset::SPORT_SITUP, cyberdog_visual_programming_abilityset::SPORT_PRESSUP,
    cyberdog_visual_programming_abilityset::SPORT_PLANK, cyberdog_visual_programming_abilityset::SPORT_JUMPJACK }

• enum cyberdog_visual_programming_abilityset::TimeMode { cyberdog_visual_programming_abilityset::Ms1970 = 0, cyberdog_visual_programming_abilityset::Y_M_D_H_M_S, cyberdog_visual_programming_abilityset::STANDARD,
    cyberdog_visual_programming_abilityset::DETAILED }

```

## 函数

- template<typename ... Args>  
std::string **cyberdog\_visual\_programming\_abilityset::Format** (const char \*format, Args... args)

- geometry\_msgs::msg::Quaternion [cyberdog\\_visual\\_programming\\_abilityset::RPY2Qrientation](#) (const double, const double, const double)
- geometry\_msgs::msg::Vector3 [cyberdog\\_visual\\_programming\\_abilityset::Qrientation2RPY](#) (const double, const double, const double, const double)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::GetTime](#) (int nowModo=0)
- double [cyberdog\\_visual\\_programming\\_abilityset::Angle2Radian](#) (const double \_degree)
- double [cyberdog\\_visual\\_programming\\_abilityset::Radian2Angle](#) (const double \_rad)
- uint64\_t [cyberdog\\_visual\\_programming\\_abilityset::GetTimeNs](#) ()
- bool [cyberdog\\_visual\\_programming\\_abilityset::JudgeToml](#) (const std::string &)
- bool [cyberdog\\_visual\\_programming\\_abilityset::GetWorkspace](#) (std::string &)
- bool [cyberdog\\_visual\\_programming\\_abilityset::Timeout](#) (const uint64\_t &\_old\_ns, uint64\_t \_timeout\_ms=3000)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::int2binary](#) (const int)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::covariance36](#) (const std::array< double, 36 > &, const std::string, const int line\_size=6)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::covariance9](#) (const std::array< double, 9 > &, const std::string, const int line\_size=3)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::sequenceGaitVector](#) (const std::vector< MsgMotion< SequenceGait > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::sequencePaceVector](#) (const std::vector< MsgMotion< SequencePace > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgFaceResVector](#) (const std::vector< MsgFaceRes > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgPersonnelVector](#) (const std::vector< MsgPersonnel > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgFaceResMap](#) (const std::map< std::string, MsgFaceRes > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgPresetVector](#) (const std::vector< MsgPreset > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgPresetMap](#) (const std::map< std::string, MsgPreset > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgTrainingWordsVector](#) (const std::vector< MsgTrainingWords > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::msgTrainingWordsMap](#) (const std::map< std::string, MsgTrainingWords > &, const std::string)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::stringVector](#) (const std::vector< std::string > &)
- std::string [cyberdog\\_visual\\_programming\\_abilityset::intVectorToString](#) (const std::vector< int > &)

## 9.6.1 详细描述

可视化编程参数模块。

定义参数约束及约束参数。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

依据参数列表均可直接调用。

版权所有

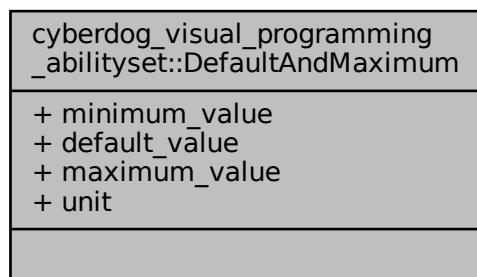
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.6.2 类说明

### 9.6.2.1 struct cyberdog\_visual\_programming\_abilityset::DefaultAndMaximum

参数约束:默认值和最大值

cyberdog\_visual\_programming\_abilityset::DefaultAndMaximum 的协作图:



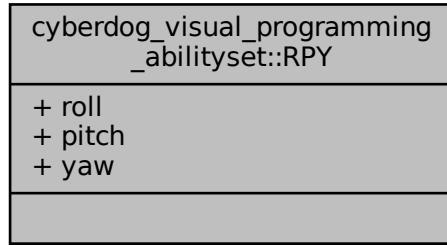
类成员

double	default_value	默认值
double	maximum_value	最大值
double	minimum_value	最小值
string	unit	单位

### 9.6.2.2 struct cyberdog\_visual\_programming\_abilityset::RPY

欧拉角约束调用

cyberdog\_visual\_programming\_abilityset::RPY 的协作图:



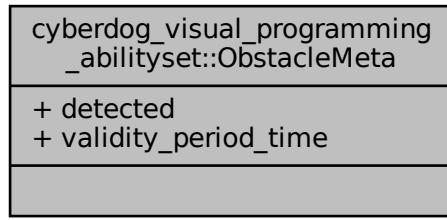
类成员

double	pitch	俯仰
double	roll	横滚
double	yaw	偏航

#### 9.6.2.3 struct cyberdog\_visual\_programming\_abilityset::ObstacleMeta

障碍物

cyberdog\_visual\_programming\_abilityset::ObstacleMeta 的协作图:



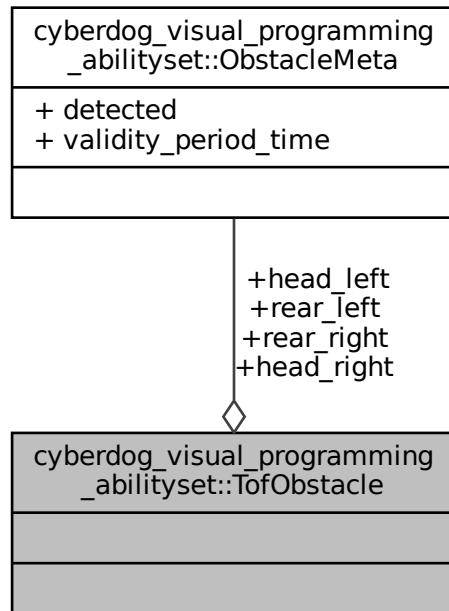
类成员

bool	detected	检测到
time_point< system_clock >	validity_period_time	有效期时间

#### 9.6.2.4 struct cyberdog\_visual\_programming\_abilityset::TofObstacle

障碍物

cyberdog\_visual\_programming\_abilityset::TofObstacle 的协作图:



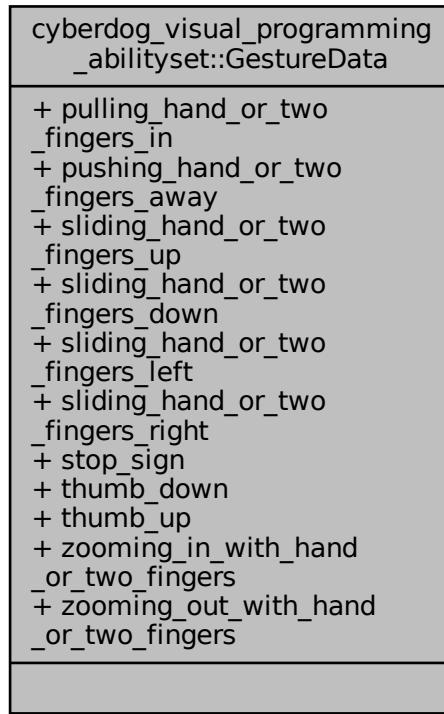
类成员

ObstacleMeta	head_left	头部左侧
ObstacleMeta	head_right	头部右侧
ObstacleMeta	rear_left	尾部左侧
ObstacleMeta	rear_right	尾部右侧

#### 9.6.2.5 struct cyberdog\_visual\_programming\_abilityset::GestureData

手势数据

cyberdog\_visual\_programming\_abilityset::GestureData 的协作图:



### 类成员

bool	pulling_hand_or_two_fingers_in	手掌拉近
bool	pushing_hand_or_two_fingers_away	手掌推开
bool	sliding_hand_or_two_fingers_down	手向下压
bool	sliding_hand_or_two_fingers_left	手向左推
bool	sliding_hand_or_two_fingers_right	手向右推
bool	sliding_hand_or_two_fingers_up	手向上抬
bool	stop_sign	停止手势
bool	thumb_down	大拇指朝下
bool	thumb_up	大拇指朝上
bool	zooming_in_with_hand_or_two_fingers	张开手掌或手指
bool	zooming_out_with_hand_or_two_fingers	闭合手掌或手指

### 9.6.3 宏定义说明

### 9.6.3.1 FORMAT

```
#define FORMAT(
    format,
    ...
)
```

值:

```
(__extension__( \
{int size = snprintf(NULL, 0, format, ## __VA_ARGS__);size++; \
char * result = new char[size]; \
snprintf(result, size, format, ## __VA_ARGS__);(result);}))
```

### 9.6.3.2 ONE\_FORMAT

```
#define ONE_FORMAT(
    format )
```

值:

```
(__extension__( \
{int size = snprintf(NULL, 0, format);size++; \
char * result = new char[size]; \
snprintf(result, size, format);(result);}))
```

### 9.6.3.3 STREAM

```
#define STREAM(
    msg )
```

值:

```
(__extension__( \
{std::ostringstream nowmsg;nowmsg << msg; \
const char * result = nowmsg.str().c_str();(result);}));
```

## 9.7 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/common.hpp 文件参考

可视化编程引擎的共享模块。

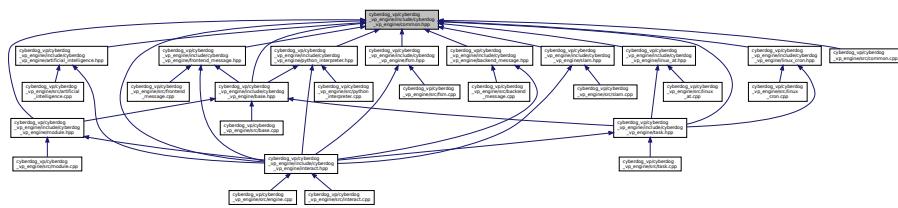
```
#include <unistd.h>
#include <rclcpp/rclcpp.hpp>
#include <rclcpp/wait_set.hpp>
#include <ament_index_cpp/get_package_share_directory.hpp>
#include <ament_index_cpp/get_package_prefix.hpp>
#include <cyberdog_common/cyberdog_log.hpp>
#include <cyberdog_common/cyberdog_toml.hpp>
#include <cyberdog_common/cyberdog_json.hpp>
#include <cyberdog_machine/cyberdog_fs_machine.hpp>
#include <rapidjson/document.h>
#include <rapidjson/prettywriter.h>
```

```
#include <rapidjson/stringbuffer.h>
#include <protocol/msg/train_plan.hpp>
#include <protocol/msg/label.hpp>
#include <protocol/msg/user_information.hpp>
#include <protocol/msg/audio_play_extend.hpp>
#include <std_msgs/msg/string.hpp>
#include <protocol/msg/visual_programming_operate.hpp>
#include <protocol/srv/visual_programming_operate.hpp>
#include <protocol/srv/get_map_label.hpp>
#include <protocol/srv/bes_http.hpp>
#include <protocol/srv/all_user_search.hpp>
#include <protocol/srv/train_plan_all.hpp>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/time.h>
#include <pybind11/pybind11.h>
#include <pybind11/embed.h>
#include <pybind11/stl.h>
#include <pybind11/complex.h>
#include <pybind11/functional.h>
#include <pybind11/chrono.h>
#include <cstdio>
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <vector>
#include <memory>
#include <map>
#include <unordered_map>
#include <mutex>
#include <tuple>
#include <regex>
#include <algorithm>
```

common.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- struct [cyberdog\\_visual\\_programming\\_engine::BackendMsg](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

## 类型定义

- using `cyberdog_visual_programming_engine::CyberdogJson` = `cyberdog::common::CyberdogJson`
- using `cyberdog_visual_programming_engine::TrainingWordsMsg` = `protocol::msg::TrainPlan`
- using `cyberdog_visual_programming_engine::AudioPlayExtendMsg` = `protocol::msg::AudioPlayExtend`
- using `cyberdog_visual_programming_engine::PersonnelMsg` = `protocol::msg::UserInformation`
- using `cyberdog_visual_programming_engine::PresetMsg` = `protocol::msg::Label`
- using `cyberdog_visual_programming_engine::GRPCMMsg` = `std_msgs::msg::String`
- using `cyberdog_visual_programming_engine::ASRMsg` = `protocol::msg::TrainPlan`
- using `cyberdog_visual_programming_engine::OperateMsg` = `protocol::msg::VisualProgrammingOperate`
- using `cyberdog_visual_programming_engine::OperateSrv` = `protocol::srv::VisualProgrammingOperate`
- using `cyberdog_visual_programming_engine::PersonnelSrv` = `protocol::srv::AllUserSearch`
- using `cyberdog_visual_programming_engine::TrainingWordsSrv` = `protocol::srv::TrainPlanAll`
- using `cyberdog_visual_programming_engine::PresetSrv` = `protocol::srv::GetMapLabel`
- using `cyberdog_visual_programming_engine::HTPSrv` = `protocol::srv::BesHttp`
- using `cyberdog_visual_programming_engine::ShellEnum` = `ShellEnum`
- using `cyberdog_visual_programming_engine::TimeMode` = `TimeMode`

## 枚举

- enum `cyberdog_visual_programming_engine::ShellEnum` { `cyberdog_visual_programming_engine::shell` = 1993, `cyberdog_visual_programming_engine::command`, `cyberdog_visual_programming_engine::command_popen`, `cyberdog_visual_programming_engine::command_error` }
- enum `cyberdog_visual_programming_engine::TimeMode` { `cyberdog_visual_programming_engine::Ms1970` = 0, `cyberdog_visual_programming_engine::Y_M_D_H_M_S`, `cyberdog_visual_programming_engine::STANDARD` }
- enum `cyberdog_visual_programming_engine::StateEnum` {
 `cyberdog_visual_programming_engine::normally` = 0, `cyberdog_visual_programming_engine::abnormally_mkdir_path` = 21, `cyberdog_visual_programming_engine::abnormally_open_file`, `cyberdog_visual_programming_engine::abnormally_build`, `cyberdog_visual_programming_engine::abnormally_register`, `cyberdog_visual_programming_engine::abnormally_update_list`, `cyberdog_visual_programming_engine::abnormally_perform`, `cyberdog_visual_programming_engine::abnormally_operate`, `cyberdog_visual_programming_engine::abnormally_request`, `cyberdog_visual_programming_engine::abnormally_other_errors`, `cyberdog_visual_programming_engine::service_request_interrupted`, `cyberdog_visual_programming_engine::service_appear_time`, `cyberdog_visual_programming_engine::service_request_timeout` }

## 函数

- std::string `cyberdog_visual_programming_engine::GetTime` (int nowModo=0)
- bool `cyberdog_visual_programming_engine::Shell` (const std::string &, int &, std::string &)
- std::vector< std::string > `cyberdog_visual_programming_engine::GetVector` (const std::string &, char, const std::string &\_head="")
- bool `cyberdog_visual_programming_engine::JudgeConfFileFile` (std::string \_file)
- bool `cyberdog_visual_programming_engine::GetWorkspace` (std::string &)
- bool `cyberdog_visual_programming_engine::Mkdir` (std::string &)
- std::string `cyberdog_visual_programming_engine::Subreplace` (const std::string &, const std::string &, const std::string &)

## 9.7.1 详细描述

可视化编程引擎的共享模块。

功能繁杂，详情参见具体内容。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

依据参数列表均可直接调用。

版权所有

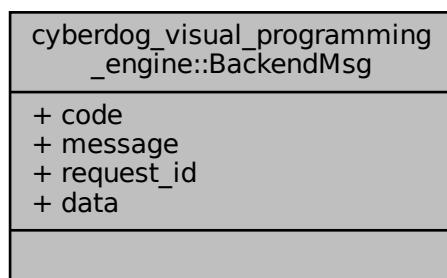
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.7.2 类说明

### 9.7.2.1 `struct cyberdog_visual_programming_engine::BackendMsg`

参数约束:后端消息

`cyberdog_visual_programming_engine::BackendMsg` 的协作图:



类成员

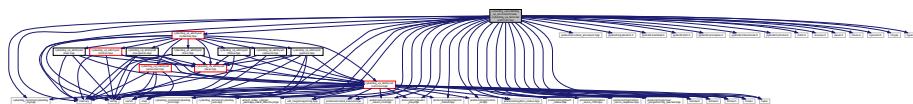
string	code	状态码
vector< string >	data	数据
string	message	消息
string	request_id	请求id

## 9.8 cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/common.hpp 文件参考

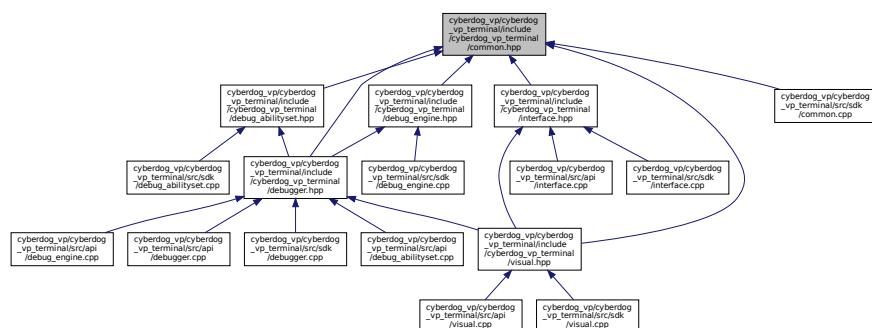
通用模块。

```
#include <cyberdog_common/cyberdog_log.hpp>
#include <cyberdog_common/cyberdog_toml.hpp>
#include <cyberdog_common/cyberdog_json.hpp>
#include <ament_index_cpp/get_package_share_directory.hpp>
#include <std_msgs/msg/string.hpp>
#include <protocol/srv/led_execute.hpp>
#include <protocol/srv/task_processor.hpp>
#include <protocol/srv/motion_result_cmd.hpp>
#include <protocol/msg/audio_play.hpp>
#include <protocol/msg/connector_status.hpp>
#include <protocol/msg/motion_id.hpp>
#include <protocol/msg/bms_status.hpp>
#include <protocol/msg/touch_status.hpp>
#include <protocol/msg/motion_servo_cmd.hpp>
#include <protocol/msg/motion_servo_response.hpp>
#include <protocol/msg/visual_programming_operate.hpp>
#include <cyberdog_vp_abilityset/cyberdog.hpp>
#include <pybind11/pybind11.h>
#include <pybind11/embed.h>
#include <pybind11/stl.h>
#include <pybind11/complex.h>
#include <pybind11/functional.h>
#include <pybind11/chrono.h>
#include <math.h>
#include <ncurses.h>
#include <panel.h>
#include <menu.h>
#include <sys/ioctl.h>
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <vector>
#include <memory>
#include <map>
#include <mutex>
#include <tuple>
#include <cctype>
```

```
#include <typeinfo>
common.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



## 命名空间

- `cyberdog_visual_programming_terminal`

## 类型定义

- using `cyberdog_visual_programming_terminal::CyberdogJson` = `cyberdog::common::CyberdogJson`
- using `cyberdog_visual_programming_terminal::CyberdogToml` = `cyberdog::common::CyberdogToml`
- using `cyberdog_visual_programming_terminal::GRPCMsg` = `std_msgs::msg::String`
- using `cyberdog_visual_programming_terminal::OperateMsg` = `protocol::msg::VisualProgrammingOperate`

## 枚举

- enum `cyberdog_visual_programming_terminal::TimeMode` { `cyberdog_visual_programming_terminal::Ms1970` = 0, `cyberdog_visual_programming_terminal::Y_M_D_H_M_S`, `cyberdog_visual_programming_terminal::STANDARD` }

## 函数

- std::string `cyberdog_visual_programming_terminal::GetTime` (int nowModo=0)
- bool `cyberdog_visual_programming_terminal::JudgeConfileFile` (std::string \_file)
- std::vector< std::string > `cyberdog_visual_programming_terminal::GetVector` (const std::string &, char, const std::string &\_head="")
- void `cyberdog_visual_programming_terminal::CoutJson` (const std::string &, const std::string &)

### 9.8.1 详细描述

通用模块。

创建及初始化机器人通用模块，以便测试调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保通用文档的正确性。

注解

确保通用的持续补充及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.9 **cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/cyberdog.hpp** 文件参考

铁蛋模块。

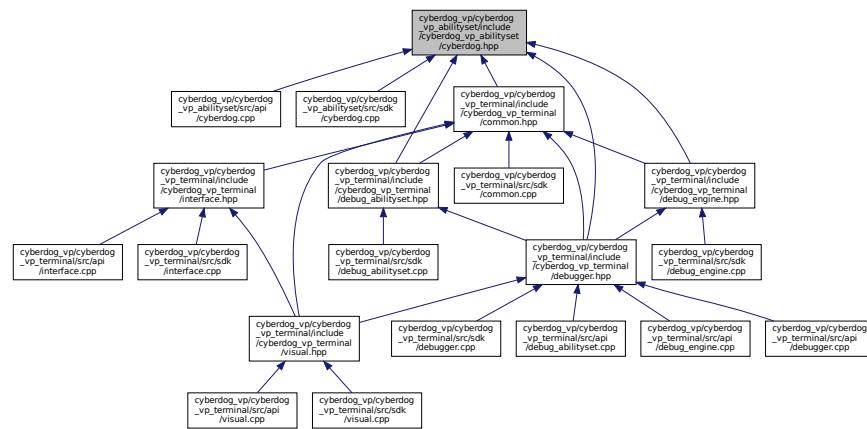
```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
#include "cyberdog_vp_abilityset/network.hpp"
#include "cyberdog_vp_abilityset/follow.hpp"
#include "cyberdog_vp_abilityset/motion.hpp"
#include "cyberdog_vp_abilityset/navigation.hpp"
#include "cyberdog_vp_abilityset/task.hpp"
```

```
#include "cyberdog_vp_abilityset/train.hpp"
#include "cyberdog_vp_abilityset/personnel.hpp"
#include "cyberdog_vp_abilityset/gesture.hpp"
#include "cyberdog_vp_abilityset/skeleton.hpp"
#include "cyberdog_vp_abilityset/audio.hpp"
#include "cyberdog_vp_abilityset/bms.hpp"
#include "cyberdog_vp_abilityset/led.hpp"
#include "cyberdog_vp_abilityset/touch.hpp"
#include "cyberdog_vp_abilityset/imu.hpp"
#include "cyberdog_vp_abilityset/odometer.hpp"
#include "cyberdog_vp_abilityset/gps.hpp"
#include "cyberdog_vp_abilityset/lidar.hpp"
#include "cyberdog_vp_abilityset/tof.hpp"
#include "cyberdog_vp_abilityset/ultrasonic.hpp"
```

cyberdog.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Cyberdog](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.9.1 详细描述

铁蛋模块。

创建及初始化铁蛋模块，以便任务调用各接口。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

ROS2 环境。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.10 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/face.hpp 文件参考

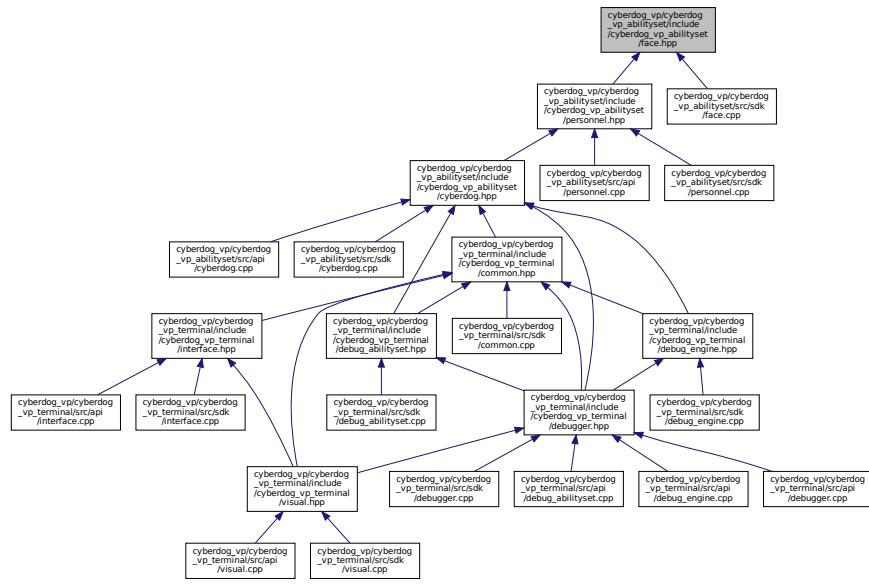
人脸识别模块。

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

face.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Face](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.10.1 详细描述

人脸识别模块。

创建及初始化人脸识别模块，以便任务调用人脸识别功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.11 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/follow.hpp 文件参考

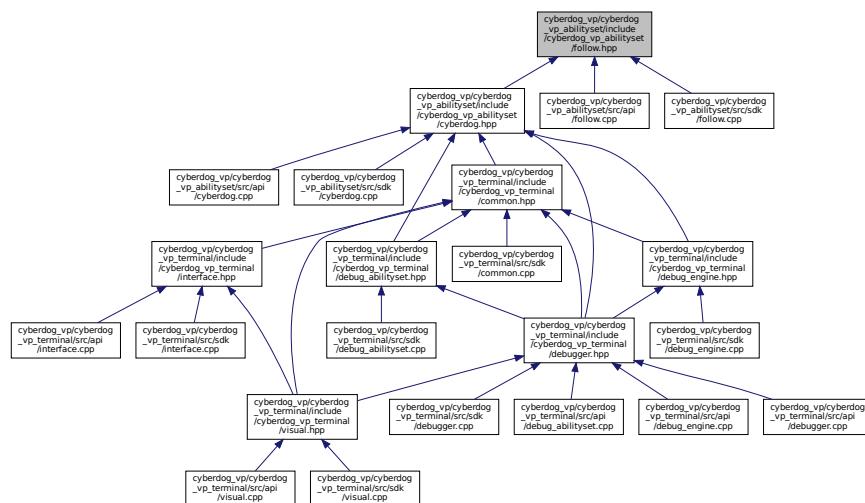
跟随模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

follow.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Follow](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.11.1 详细描述

跟随模块。

创建及初始化机器人跟随能力，以便任务调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.12 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/gesture.hpp 文件参

考

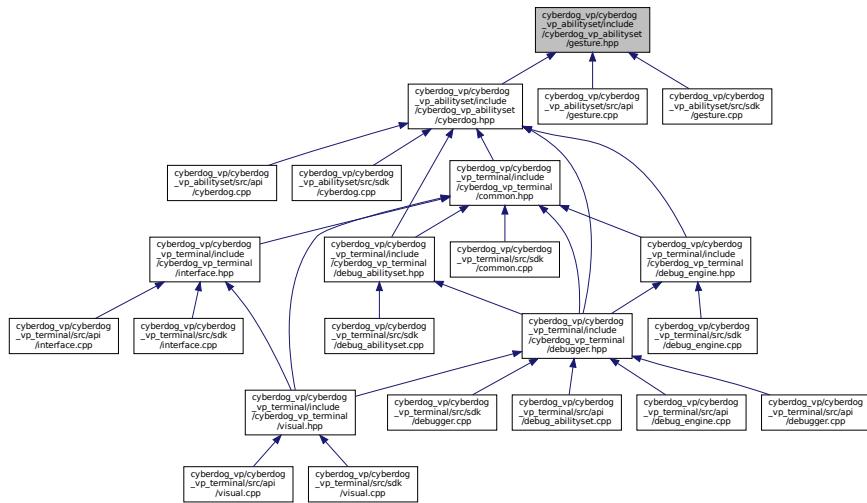
手势识别模块。

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

gesture.hpp 的引用(include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Gesture](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.12.1 详细描述

手势识别模块。

声纹识别模块。

创建及初始化手势识别模块，以便任务调用手势识别功能。

## 作者

尚子涵

Shang Zihan

## 版本

1.1.0.0

## 日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

创建及初始化声纹识别模块，以便任务调用声纹识别功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.13 **cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/gps.hpp** 文件参考

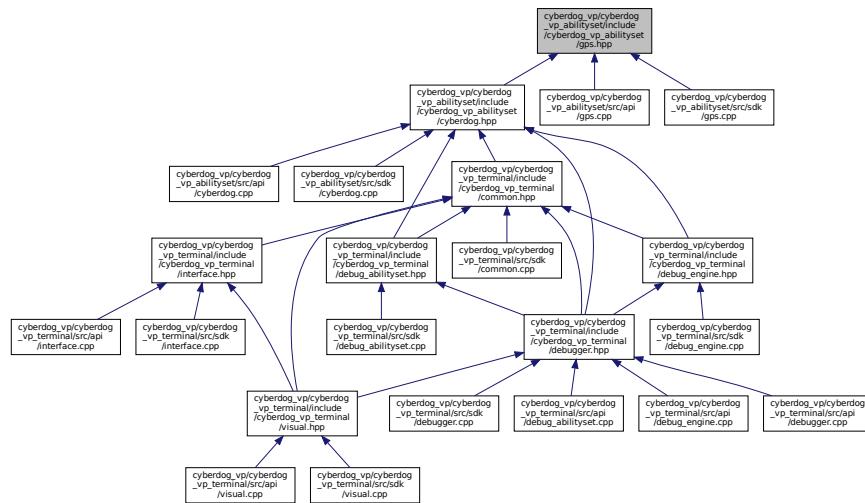
GPS模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

gps.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Gps](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.13.1 详细描述

GPS模块。

创建及初始化GPS模块，以便任务调用GPS功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化传感器。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.14 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset imu.hpp 文件参考

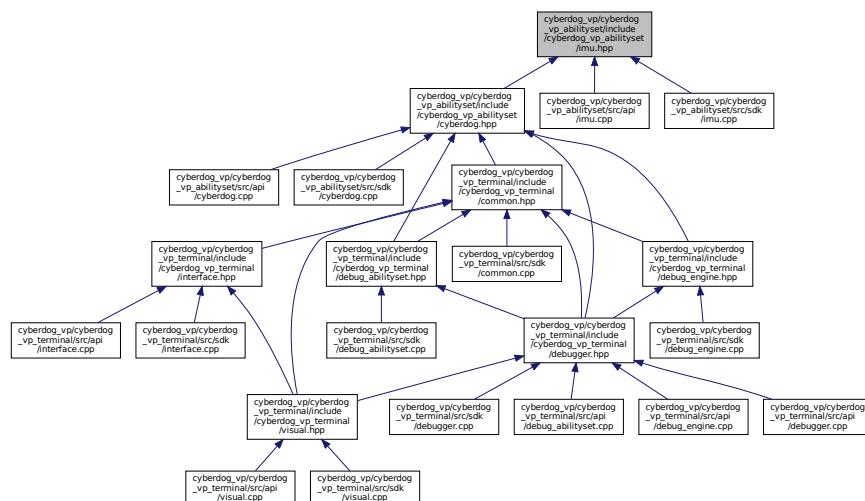
imu模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

imu.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Imu](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.14.1 详细描述

imu模块。

创建及初始化机器人惯导imu能力，以便任务调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.15 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/led.hpp 文件参考

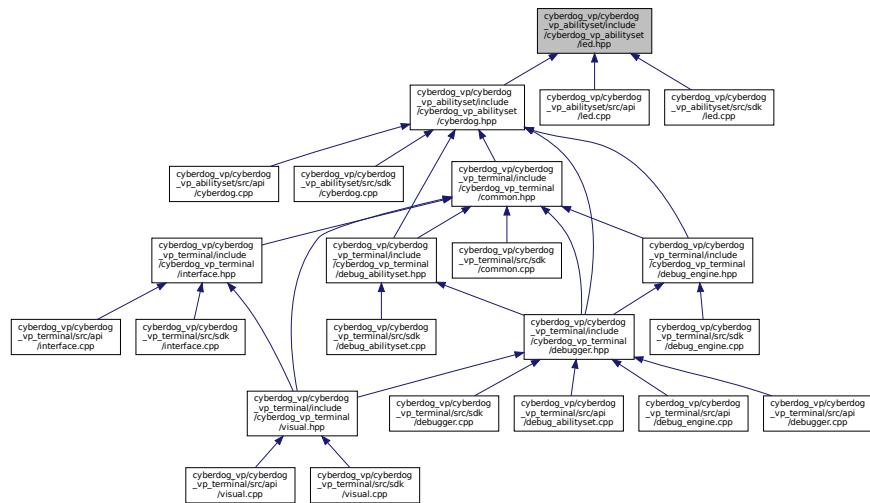
led模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

led.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_abilityset::Led`

命名空间

- `cyberdog_visual_programming_abilityset`

### 9.15.1 详细描述

led模块。

创建及初始化led模块，以便任务调用led功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.16 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/lidar.hpp 文件参考

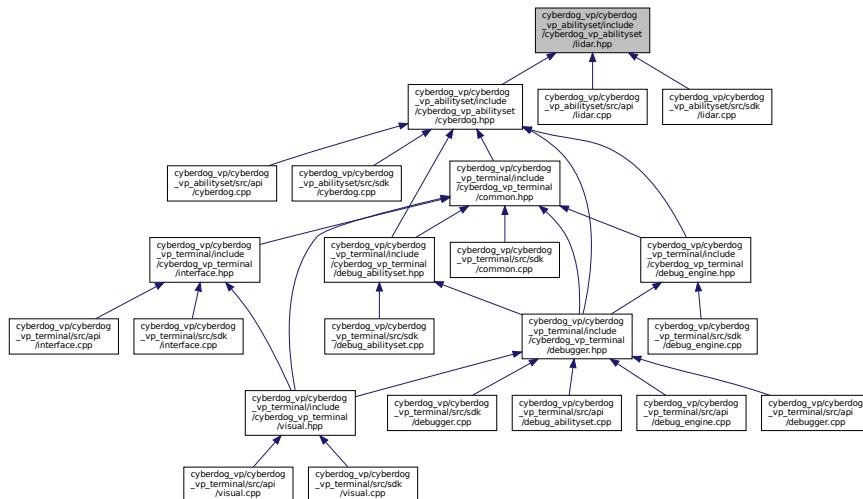
雷达模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

lidar.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_abilityset::Lidar`

命名空间

- `cyberdog_visual_programming_abilityset`

### 9.16.1 详细描述

雷达模块。

创建及初始化雷达模块，以便任务调用雷达功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化传感器。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.17 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/log.hpp 文件参考

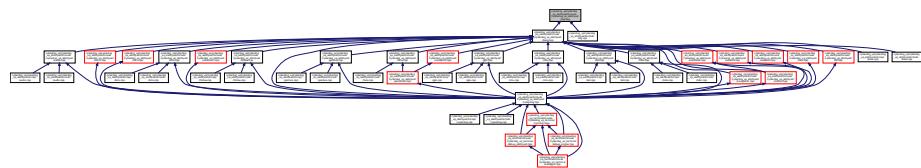
可视化编程SDK的日志模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
```

log.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Log](#)

## 命名空间

- `cyberdog_visual_programming_abilityset`

### 9.17.1 详细描述

可视化编程SDK的日志模块。

创建及初始化SDK模块的日志逻辑。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化铁蛋模块。

#### 版权所有

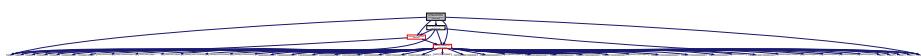
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.18 `cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/motion.hpp` 文件参考

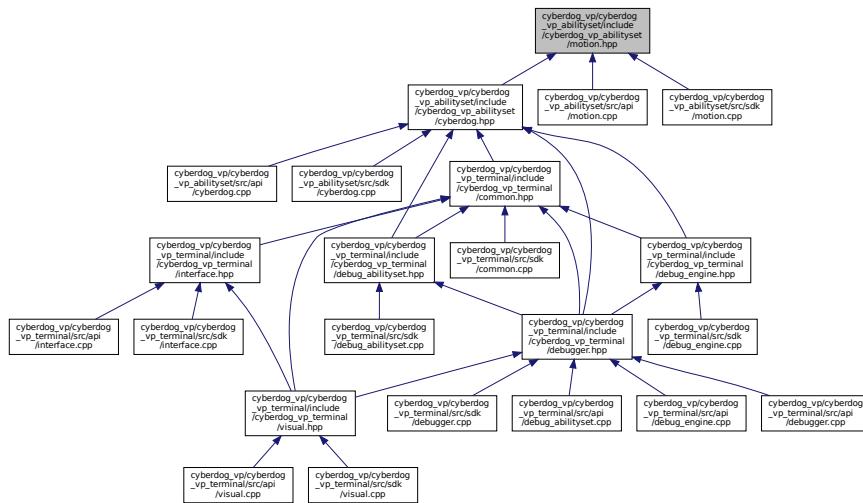
运动模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
#include "cyberdog_vp_abilityset/odometer.hpp"
```

`motion.hpp` 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Motion](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.18.1 详细描述

运动模块。

创建及初始化机器人运动能力，以便任务调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.19 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/navigation.hpp 文件参考

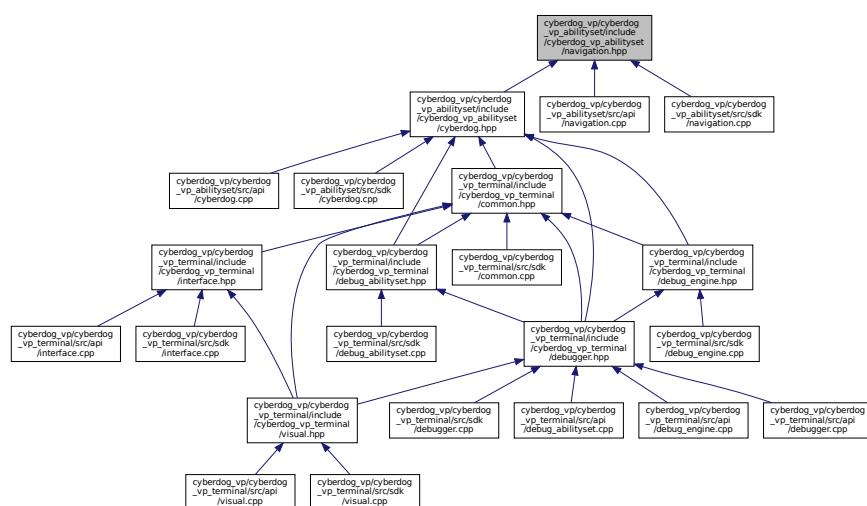
导航模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

navigation.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Navigation](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.19.1 详细描述

导航模块。

创建及初始化机器人导航能力，以便任务调用。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

#### 注解

确保接口的优化及同步。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

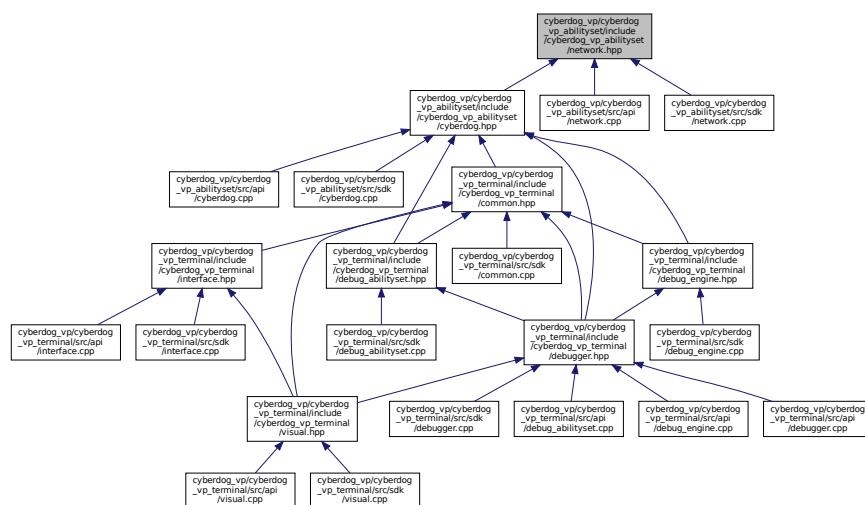
## 9.20 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/network.hpp 文件参考

网络模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
network.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Network](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.20.1 详细描述

网络模块。

创建及初始化机器人网络能力，以便任务调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.21 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/odometer.hpp 文件参考

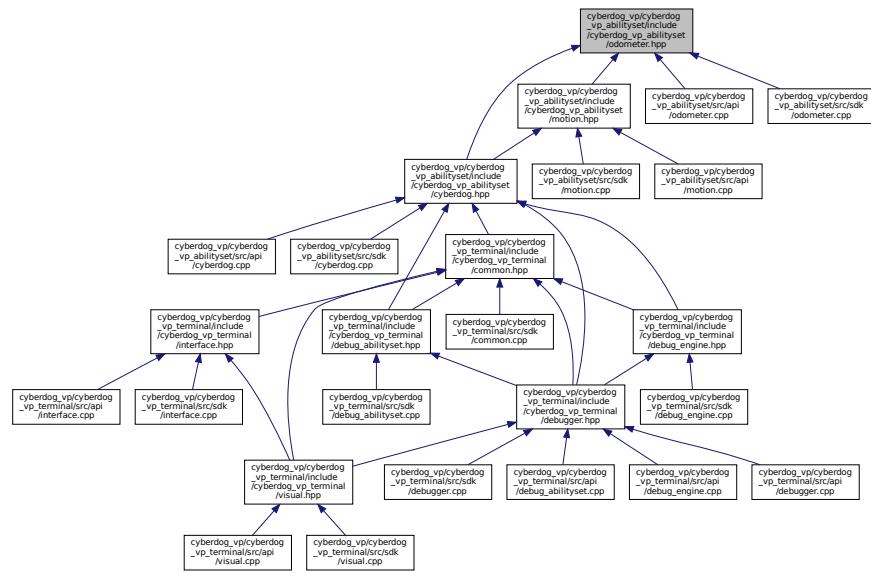
里程计模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

odometer.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Odometer](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.21.1 详细描述

里程计模块。

创建及初始化机器人里程计能力，以便任务调用。

## 作者

尚子涵

Shang Zihan

## 版本

1.1.0.0

## 日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.22 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/personnel.hpp 文件参考

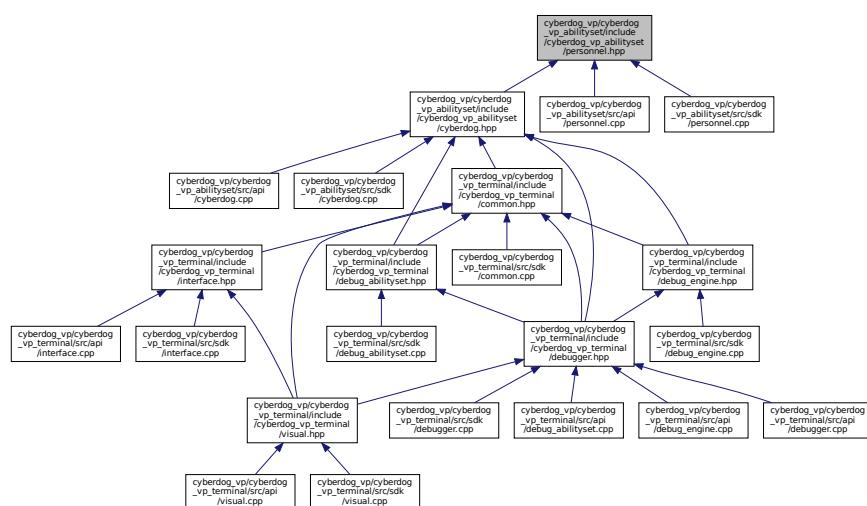
人员识别模块。

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/face.hpp"
#include "cyberdog_vp_abilityset/voiceprint.hpp"
```

personnel.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Personnel](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.22.1 详细描述

人员识别模块。

创建及初始化人脸识别模块，以便任务调用人脸识别功能。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化设备。

#### 版权所有

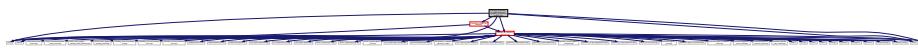
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.23 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/skeleton.hpp 文件参考

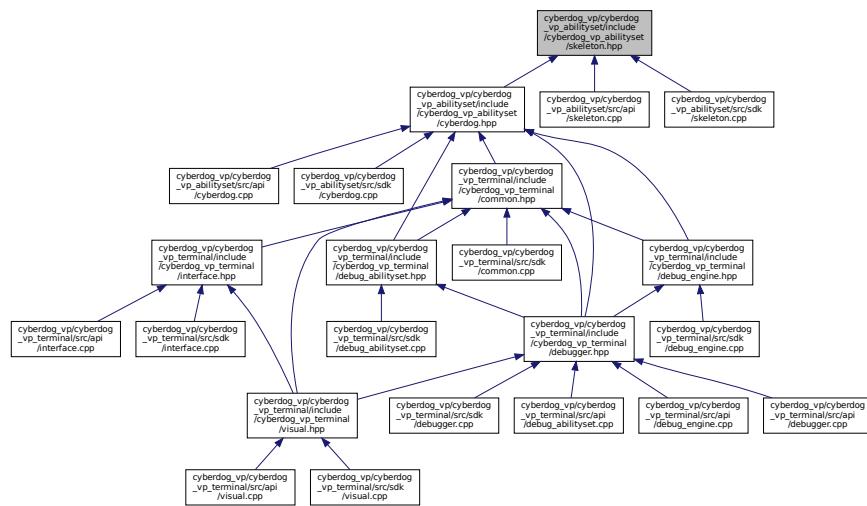
骨骼（点）识别模块。

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

skeleton.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Skeleton](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.23.1 详细描述

骨骼（点）识别模块。

创建及初始化骨骼（点）识别模块，以便任务调用骨骼（点）识别功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化设备。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.24 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/task.hpp 文件参考

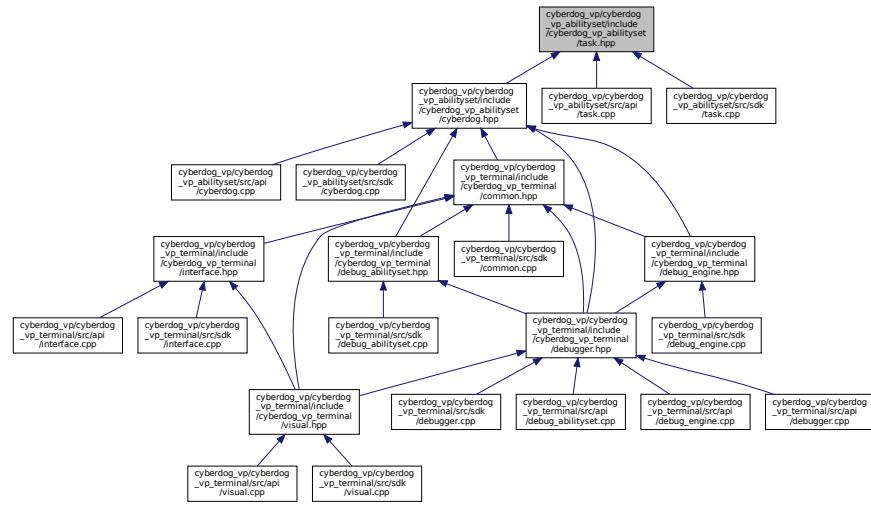
任务模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

task.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Task](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.24.1 详细描述

任务模块。

创建及初始化任务模块，以便控制任务执行状态。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

注解

留意各种场景下控制是否有效

版权所有

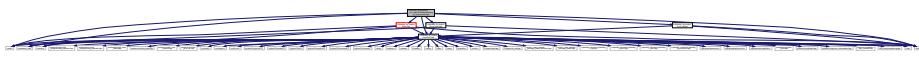
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.25 `cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/task.hpp` 文件参考

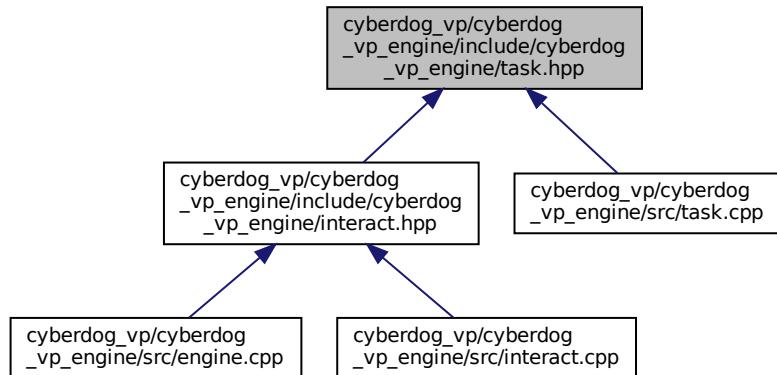
任务编程模块。

```
#include <string>
#include <vector>
#include <unordered_map>
#include "cyberdog_vp_engine/common.hpp"
#include "cyberdog_vp_engine/base.hpp"
#include "cyberdog_vp_engine/linux_at.hpp"
#include "cyberdog_vp_engine/linux_cron.hpp"
```

task.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_engine::Task`

命名空间

- `cyberdog_visual_programming_engine`

### 9.25.1 详细描述

任务编程模块。

负责任务编程，生成任务文件以便任务注册过程中调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

相互作用有效。

**Bug** 任务操作尚待调试

任务依赖模块的筛查尚待调试

依赖模块的任务的编程尚待调试

警告

小心注册表的更新及任务的删除操作

注解

留意注册表的稳定性

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.26 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/tof.hpp 文件参考

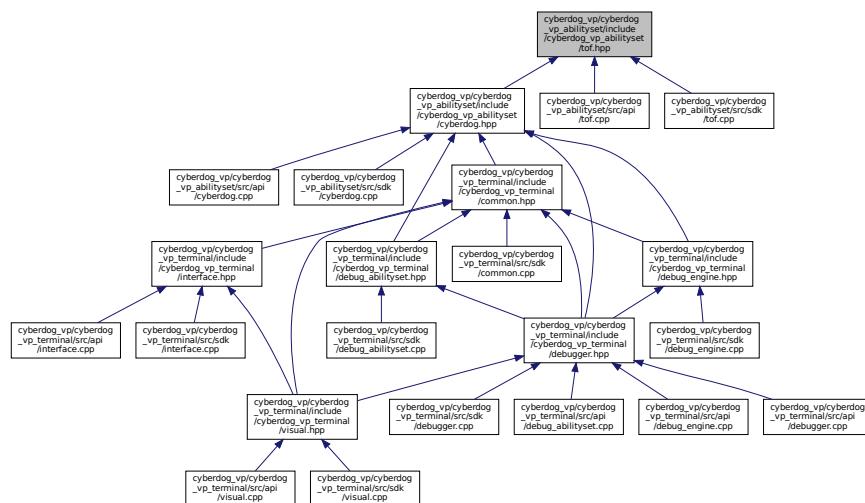
激光测距模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

tof.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_abilityset::Tof](#)

命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.26.1 详细描述

激光测距模块。

创建及初始化激光测距模块，以便任务调用激光测距功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化传感器。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

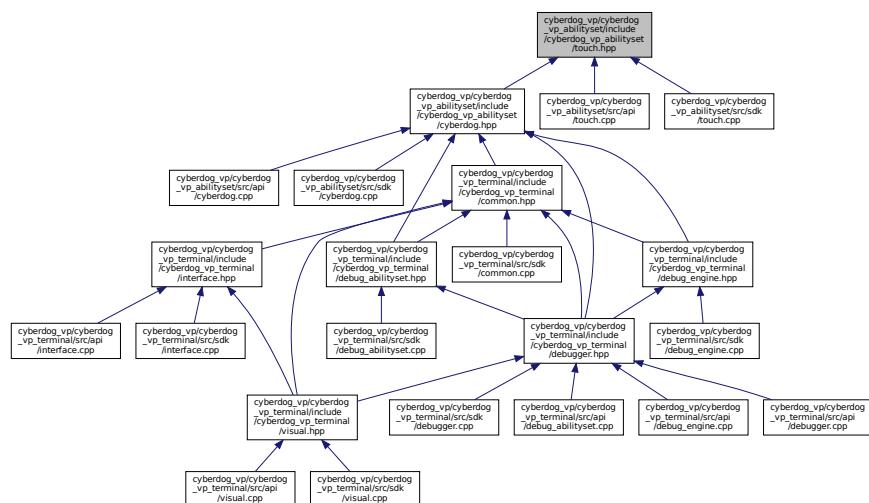
9.27 `cyberdog_vp/cyberdog_vp_abilityset/include/cyberdog_vp_abilityset/touch.hpp` 文件参考

触摸板模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
touch.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Touch](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.27.1 详细描述

触摸板模块。

创建及初始化触摸板模块，以便任务调用触摸板功能。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化设备。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.28 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/train.hpp 文件参考

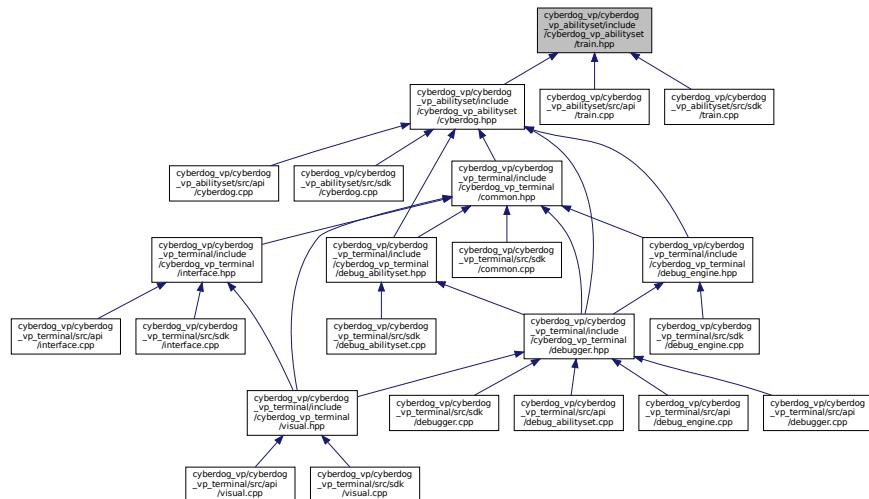
训练模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

train.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_abilityset::Train`

命名空间

- `cyberdog_visual_programming_abilityset`

### 9.28.1 详细描述

训练模块。

创建及初始化训练模块，以便训练接口在任务中正常使用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

注解

训练词的增、删、改逻辑由开发者用户完成。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

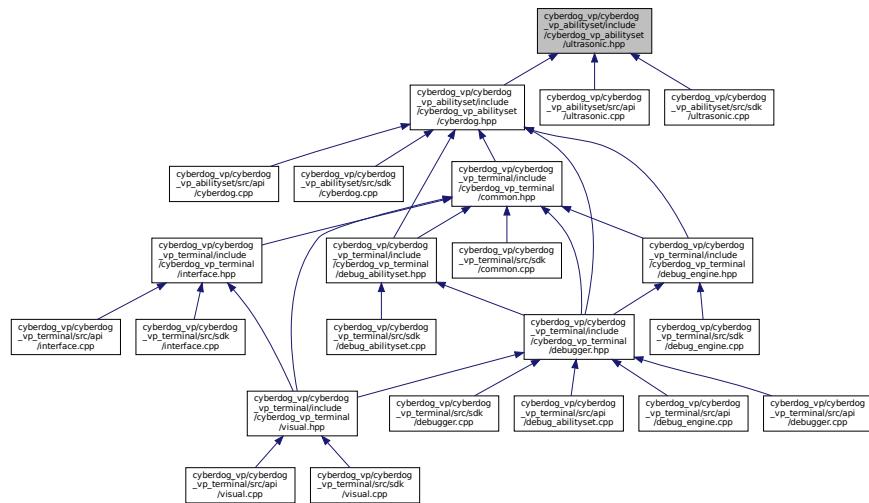
## 9.29 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/ultrasonic.hpp 文件参考

超声波模块。

```
#include <string>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
ultrasonic.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_abilityset::Ultrasonic](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.29.1 详细描述

超声波模块。

创建及初始化超声波模块，以便任务调用超声波功能。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

初始化传感器。

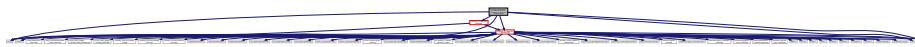
#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

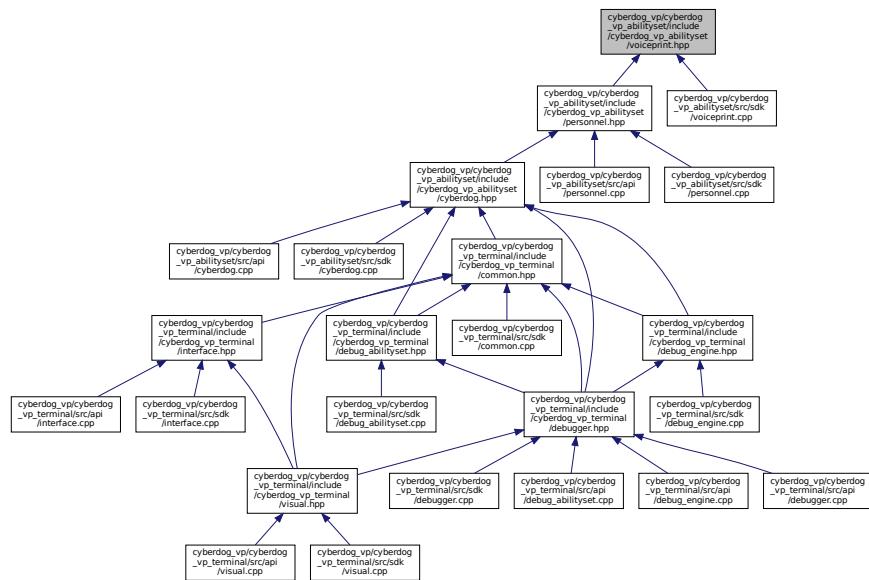
## 9.30 cyberdog\_vp/cyberdog\_vp\_abilityset/include/cyberdog\_vp\_abilityset/voiceprint.hpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
```

voiceprint.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_abilityset::Voiceprint`

命名空间

- `cyberdog_visual_programming_abilityset`

### 9.31 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/audio.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/audio.hpp"
audio.cpp 的引用(Include)关系图:
```



#### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

#### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineAudio](#) (py::object m)

### 9.32 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/audio.cpp 文件参考

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/audio.hpp"
audio.cpp 的引用(Include)关系图:
```



#### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 9.33 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/base.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/base.hpp"
base.cpp 的引用(Include)关系图:
```



#### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void `cyberdog_visual_programming_abilityset_py::DefineBase (py::object m)`

## 9.34 `cyberdog_vp/cyberdog_vp_abilityset/src/sdk/base.cpp` 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/base.hpp"
base.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 9.35 `cyberdog_vp/cyberdog_vp_engine/src/base.cpp` 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/base.hpp"
base.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_engine`

## 9.36 `cyberdog_vp/cyberdog_vp_abilityset/src/api/bms.cpp` 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/bms.hpp"
bms.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset_py`

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineBms](#) (py::object m)

## 9.37 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/bms.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/bms.hpp"
bms.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 变量

- std::mutex [cyberdog\\_visual\\_programming\\_abilityset::bms\\_data\\_cvm\\_](#)
- std::condition\_variable [cyberdog\\_visual\\_programming\\_abilityset::bms\\_data\\_cv\\_](#)

## 9.38 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/common.cpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include <map>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/common.hpp"
common.cpp 的引用(Include)关系图:
```



## 9.39 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/common.cpp 文件参考

```
#include <string>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/common.hpp"
common.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 函数

- `geometry_msgs::msg::Quaternion cyberdog_visual_programming_abilityset::RPY2Qrientation (const double, const double, const double)`
- `geometry_msgs::msg::Vector3 cyberdog_visual_programming_abilityset::Orientation2RPY (const double, const double, const double, const double)`
- `std::string cyberdog_visual_programming_abilityset::GetTime (int nowModo=0)`
- `double cyberdog_visual_programming_abilityset::Angle2Radian (const double _degree)`
- `double cyberdog_visual_programming_abilityset::Radian2Angle (const double _rad)`
- `uint64_t cyberdog_visual_programming_abilityset::GetTimeNs ()`
- `bool cyberdog_visual_programming_abilityset::JudgeToml (const std::string &)`
- `bool cyberdog_visual_programming_abilityset::GetWorkspace (std::string &)`
- `bool cyberdog_visual_programming_abilityset::Timeout (const uint64_t &old_ns, uint64_t _timeout_ms=3000)`
- `std::string cyberdog_visual_programming_abilityset::int2binary (const int)`
- `std::string cyberdog_visual_programming_abilityset::covariance36 (const std::array< double, 36 > &, const std::string, const int line_size=6)`
- `std::string cyberdog_visual_programming_abilityset::covariance9 (const std::array< double, 9 > &, const std::string, const int line_size=3)`
- `std::string cyberdog_visual_programming_abilityset::sequenceGaitVector (const std::vector< MsgMotion< SequenceGait > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::sequencePaceVector (const std::vector< MsgMotion< SequencePace > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgFaceResVector (const std::vector< MsgFaceRes > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgPersonnelVector (const std::vector< MsgPersonnel > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgFaceResMap (const std::map< std::string, MsgFaceRes > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgPresetVector (const std::vector< MsgPreset > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgPresetMap (const std::map< std::string, MsgPreset > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgTrainingWordsVector (const std::vector< MsgTrainingWords > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::msgTrainingWordsMap (const std::map< std::string, MsgTrainingWords > &, const std::string)`
- `std::string cyberdog_visual_programming_abilityset::stringVector (const std::vector< std::string > &)`
- `std::string cyberdog_visual_programming_abilityset::intVectorToString (const std::vector< int > &)`

## 9.40 `cyberdog_vp/cyberdog_vp_engine/src/common.cpp` 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/common.hpp"
common.cpp 的引用(Include)关系图:
```



## 命名空间

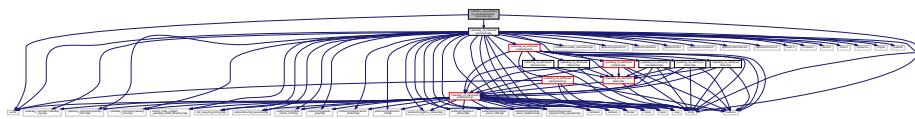
- `cyberdog_visual_programming_engine`

## 函数

- `std::string cyberdog_visual_programming_engine::GetTime (int nowModo=0)`
- `bool cyberdog_visual_programming_engine::Shell (const std::string &, int &, std::string &)`
- `std::vector< std::string > cyberdog_visual_programming_engine::GetVector (const std::string &, char, const std::string &_head="")`
- `bool cyberdog_visual_programming_engine::JudgeConfileFile (std::string _file)`
- `bool cyberdog_visual_programming_engine::GetWorkspace (std::string &)`
- `bool cyberdog_visual_programming_engine::Mkdir (std::string &)`
- `std::string cyberdog_visual_programming_engine::Subreplace (const std::string &, const std::string &, const std::string &)`

## 9.41 cyberdog\_vp/cyberdog\_vp\_terminal/src/sdk/common.cpp 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_terminal/common.hpp"
common.cpp 的引用(Include)关系图:
```



## 命名空间

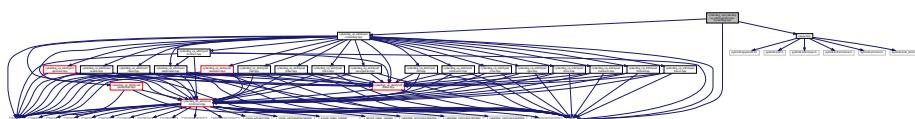
- `cyberdog_visual_programming_terminal`

## 函数

- `std::string cyberdog_visual_programming_terminal::GetTime (int nowModo=0)`
- `bool cyberdog_visual_programming_terminal::JudgeConfileFile (std::string _file)`
- `std::vector< std::string > cyberdog_visual_programming_terminal::GetVector (const std::string &, char, const std::string &_head="")`
- `void cyberdog_visual_programming_terminal::CoutJson (const std::string &, const std::string &)`

## 9.42 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/cyberdog.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/cyberdog.hpp"
cyberdog.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineCyberdog](#) (py::object m)

## 9.43 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/cyberdog.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include "cyberdog_vp_abilityset/cyberdog.hpp"
cyberdog.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 9.44 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/follow.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/follow.hpp"
follow.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineFollow](#) (py::object m)

## 9.45 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/follow.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/follow.hpp"
```

follow.cpp 的引用(Include)关系图:



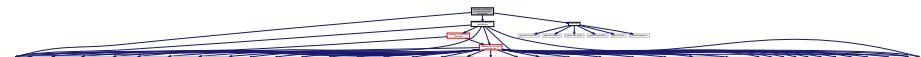
### 命名空间

- `cyberdog_visual_programming_abilityset`

## 9.46 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/gesture.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/gesture.hpp"
```

gesture.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset_py`

### 函数

- `void cyberdog_visual_programming_abilityset_py::DefineGesture (py::object m)`

## 9.47 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/gesture.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/gesture.hpp"
```

gesture.cpp 的引用(Include)关系图:



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 变量

- `std::mutex cyberdog_visual_programming_abilityset::gesture_recognition_state_cvm_`
- `std::condition_variable cyberdog_visual_programming_abilityset::gesture_recognition_state_cv_`

## 9.48 `cyberdog_vp/cyberdog_vp_abilityset/src/api/gps.cpp` 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/gps.hpp"
gps.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset_py`

## 函数

- `void cyberdog_visual_programming_abilityset_py::DefineGps (py::object m)`

## 9.49 `cyberdog_vp/cyberdog_vp_abilityset/src/sdk/gps.cpp` 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/gps.hpp"
gps.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 变量

- `std::mutex cyberdog_visual_programming_abilityset::gps_data_cvm_`
- `std::condition_variable cyberdog_visual_programming_abilityset::gps_data_cv_`

## 9.50 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api imu.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/imu.hpp"
```

imu.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset.py`

### 函数

- `void cyberdog_visual_programming_abilityset.py::Defineimu (py::object m)`

## 9.51 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk imu.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/imu.hpp"
```

imu.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset`

### 变量

- `std::mutex cyberdog_visual_programming_abilityset::imu_data_cvm_`
- `std::condition_variable cyberdog_visual_programming_abilityset::imu_data_cv_`

## 9.52 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api led.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/led.hpp"
```

led.cpp 的引用(Include)关系图:



## 命名空间

- `cyberdog_visual_programming_abilityset.py`

## 函数

- void `cyberdog_visual_programming_abilityset.py::DefineLed (py::object m)`

## 9.53 `cyberdog_vp/cyberdog_vp_abilityset/src/sdk/led.cpp` 文件参考

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/led.hpp"
led.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 9.54 `cyberdog_vp/cyberdog_vp_abilityset/src/api/lidar.cpp` 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/lidar.hpp"
lidar.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset.py`

## 函数

- void `cyberdog_visual_programming_abilityset.py::DefineLidar (py::object m)`

## 9.55 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/lidar.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/lidar.hpp"
```

lidar.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset`

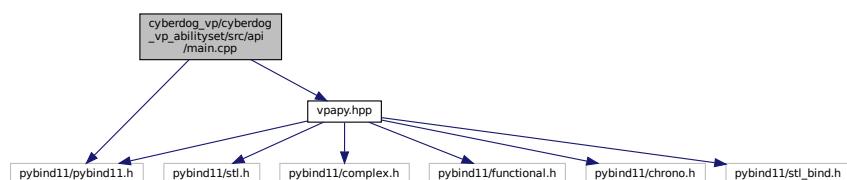
### 变量

- `std::mutex cyberdog_visual_programming_abilityset::lidar_data_cvm_`
- `std::condition_variable cyberdog_visual_programming_abilityset::lidar_data_cv_`

## 9.56 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/main.cpp 文件参考

```
#include <pybind11/pybind11.h>
#include "vpappy.hpp"
```

main.cpp 的引用(Include)关系图:



### 命名空间

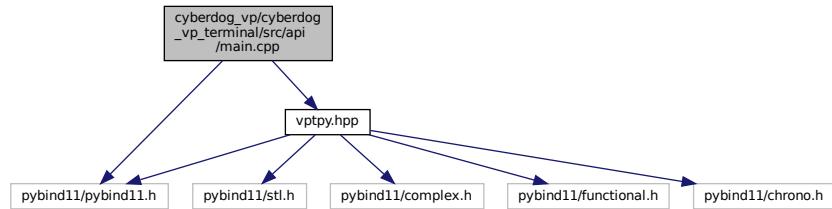
- `cyberdog_visual_programming_abilityset_py`

### 函数

- `cyberdog_visual_programming_abilityset_py::PYBIND11_MODULE` (abilityset, m)

## 9.57 cyberdog\_vp/cyberdog\_vp\_terminal/src/api/main.cpp 文件参考

```
#include <pybind11/pybind11.h>
#include "vptpy.hpp"
main.cpp 的引用(Include)关系图:
```



### 命名空间

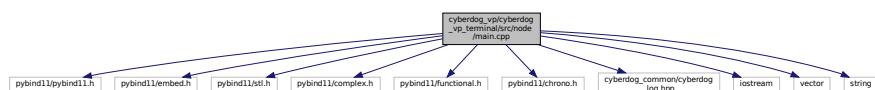
- [cyberdog\\_visual\\_programming\\_terminal.py](#)

### 函数

- [cyberdog\\_visual\\_programming\\_terminal.py::PYBIND11\\_MODULE](#) (terminal, m)

## 9.58 cyberdog\_vp/cyberdog\_vp\_terminal/src/node/main.cpp 文件参考

```
#include <pybind11/pybind11.h>
#include <pybind11/embed.h>
#include <pybind11/stl.h>
#include <pybind11/complex.h>
#include <pybind11/functional.h>
#include <pybind11/chrono.h>
#include <cyberdog_common/cyberdog_log.hpp>
#include <iostream>
#include <vector>
#include <string>
main.cpp 的引用(Include)关系图:
```



### 函数

- int [main](#) (int argc, char \*\*argv)

### 9.58.1 函数说明

#### 9.58.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

## 9.59 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/motion.cpp 文件参考

#include <string>

#include "vpapy.hpp"

#include "cyberdog\_vp\_abilityset/motion.hpp"

motion.cpp 的引用(Include)关系图:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineMotion](#) (py::object m)

## 9.60 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/motion.cpp 文件参考

#include <string>

#include <memory>

#include "cyberdog\_vp\_abilityset/motion.hpp"

motion.cpp 的引用(Include)关系图:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 变量

- std::mutex [cyberdog\\_vp/cyberdog\\_vp\\_abilityset/src/api/navigation.cpp](#)::compensation\_frame\_mtx\_

## 9.61 [cyberdog\\_vp/cyberdog\\_vp\\_abilityset/src/api/navigation.cpp](#) 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/navigation.hpp"
navigation.cpp 的引用(Include)关系图:
```



## 命名空间

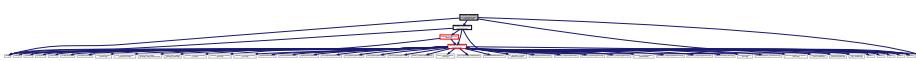
- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineNavigation](#) (py::object m)

## 9.62 [cyberdog\\_vp/cyberdog\\_vp\\_abilityset/src/sdk/navigation.cpp](#) 文件参考

```
#include <string>
#include <memory>
#include <map>
#include "cyberdog_vp_abilityset/navigation.hpp"
navigation.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 9.63 [cyberdog\\_vp/cyberdog\\_vp\\_abilityset/src/api/network.cpp](#) 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/network.hpp"
network.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineNetwork](#) (py::object m)

## 9.64 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/network.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/network.hpp"
network.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 9.65 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/odometer.cpp 文件参考

```
#include <string>
#include "vpappy.hpp"
#include "cyberdog_vp_abilityset/odometer.hpp"
odometer.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineOdometer](#) (py::object m)

## 9.66 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/odometer.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/odometer.hpp"
odometer.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 变量

- std::mutex [cyberdog\\_visual\\_programming\\_abilityset::odometer\\_data\\_cvm](#)
- std::condition\_variable [cyberdog\\_visual\\_programming\\_abilityset::odometer\\_data\\_cv](#)

## 9.67 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/personnel.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/personnel.hpp"
personnel.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset\\_py](#)

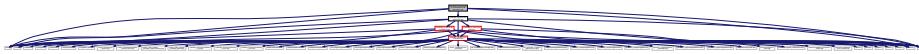
### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefinePersonnel](#) (py::object m)

## 9.68 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/personnel.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/personnel.hpp"
```

personnel.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset`

## 9.69 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/ros\_builtin\_interfaces.cpp 文件参考

```
#include <builtin_interfaces/msg/time.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

ros\_builtin\_interfaces.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset_py`

### 函数

- `void cyberdog_visual_programming_abilityset_py::DefineBuiltInInterfaces (py::object m)`
- `void cyberdog_visual_programming_abilityset_py::DefineTimer (py::object m)`

## 9.70 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/ros\_geometry\_msgs.cpp 文件参考

```
#include <geometry_msgs/msg/point.hpp>
#include <geometry_msgs/msg/quaternion.hpp>
#include <geometry_msgs/msg/pose.hpp>
#include <geometry_msgs/msg/vector3.hpp>
#include <geometry_msgs/msg/twist.hpp>
#include <geometry_msgs/msg/pose_with_covariance.hpp>
#include <geometry_msgs/msg/twist_with_covariance.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

ros\_geometry\_msgs.cpp 的引用(Include)关系图:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineGeometryMsgs \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefinePoint \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineQuaternion \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefinePose \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineVector3 \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineTwist \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefinePoseWithCovariance \(py::object m\)](#)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineTwistWithCovariance \(py::object m\)](#)

## 9.71 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/ros\_nav\_msgs.cpp 文件参考

```
#include <nav_msgs/msg/odometry.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

ros\_nav\_msgs.cpp 的引用(Include)关系图:



## 命名空间

- `cyberdog_visual_programming_abilityset.py`

## 函数

- void `cyberdog_visual_programming_abilityset.py::DefineNavMsgs` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgOdometry` (py::object m)

## 9.72 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/ros\_protocol.cpp 文件参考

```
#include <protocol/msg/bms_status.hpp>
#include <protocol/msg/touch_status.hpp>
#include <protocol/msg/connector_status.hpp>
#include <protocol/msg/motion_sequence_param.hpp>
#include <protocol/srv/audio_text_play.hpp>
#include <protocol/srv/led_execute.hpp>
#include <protocol/srv/motion_result_cmd.hpp>
#include <protocol/srv/motion_sequence.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

`ros_protocol.cpp` 的引用(Include)关系图:



## 命名空间

- `cyberdog_visual_programming_abilityset.py`

## 函数

- void `cyberdog_visual_programming_abilityset.py::DefineProtocol` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgBmsStatus` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgTouchStatus` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgConnectorStatus` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequenceGait` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequencePace` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgGpsPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgSingleTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgHeadTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgRearTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgFaceRes` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgPreset` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgSport` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgTrainingWords` (py::object m)

- void `cyberdog_visual_programming_abilityset.py::DefineSrvFaceRecResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvSportResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvTrainingWordsResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvAudioTextPlayResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvLedExecuteResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvMotionResultCmdResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvMotionSequenceShowResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineActNavigationResult` (py::object m)

## 9.73 `cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_sensor_msgs.cpp` 文件参考

```
#include <sensor_msgs/msg/range.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

`ros_sensor_msgs.cpp` 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset.py`

### 函数

- void `cyberdog_visual_programming_abilityset.py::DefineSensorMsgs` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgRange` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgLaserScan` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgIMU` (py::object m)

## 9.74 `cyberdog_vp/cyberdog_vp_abilityset/src/api/ros_std_msgs.cpp` 文件参考

```
#include <std_msgs/msg/header.hpp>
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_abilityset/common.hpp"
#include "vpapy.hpp"
```

`ros_std_msgs.cpp` 的引用(Include)关系图:



## 命名空间

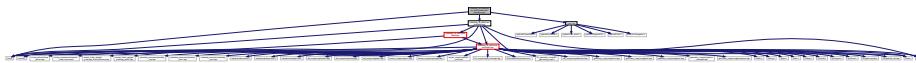
- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineStdMsgs](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineHeader](#) (py::object m)

## 9.75 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/skeleton.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/skeleton.hpp"
skeleton.cpp 的引用(Include)关系图:
```



## 命名空间

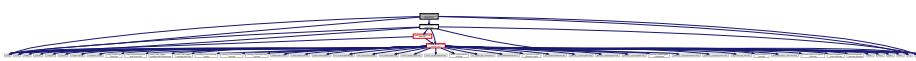
- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

## 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineSkeleton](#) (py::object m)

## 9.76 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/skeleton.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/skeleton.hpp"
skeleton.cpp 的引用(Include)关系图:
```



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 变量

- std::mutex `cyberdog_visual_programming_abilityset::skeleton_recognition_state_cvm_`
- std::condition\_variable `cyberdog_visual_programming_abilityset::skeleton_recognition_state_cv_`

## 9.77 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/task.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/task.hpp"
task.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset_py`

## 函数

- void `cyberdog_visual_programming_abilityset_py::DefineTask (py::object m)`

## 9.78 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/task.cpp 文件参考

```
#include <string>
#include <memory>
#include "cyberdog_vp_abilityset/task.hpp"
task.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 变量

- std::mutex `cyberdog_visual_programming_abilityset::task_state_cvm_`
- std::condition\_variable `cyberdog_visual_programming_abilityset::task_state_cv_`

## 9.79 cyberdog\_vp/cyberdog\_vp\_engine/src/task.cpp 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/task.hpp"
task.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

## 9.80 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/tof.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/tof.hpp"
tof.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset\\_py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineTof](#) (py::object m)

## 9.81 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/tof.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/tof.hpp"
tof.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 变量

- std::mutex `cyberdog_visual_programming_abilityset::tov_data_cvm_`
- std::condition\_variable `cyberdog_visual_programming_abilityset::tov_data_cv_`

## 9.82 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/touch.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/touch.hpp"
touch.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset.py`

## 函数

- void `cyberdog_visual_programming_abilityset.py::DefineTouch (py::object m)`

## 9.83 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/touch.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/touch.hpp"
touch.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_abilityset`

## 变量

- std::mutex `cyberdog_visual_programming_abilityset::touch_data_cvm_`
- std::condition\_variable `cyberdog_visual_programming_abilityset::touch_data_cv_`

## 9.84 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/train.cpp 文件参考

```
#include <string>
#include <memory>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/train.hpp"
```

train.cpp 的引用(Include)关系图:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineTrain \(py::object m\)](#)

## 9.85 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/train.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/train.hpp"
```

train.cpp 的引用(Include)关系图:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

### 变量

- std::mutex [cyberdog\\_visual\\_programming\\_abilityset::training\\_words\\_state\\_cvm\\_](#)
- std::condition\_variable [cyberdog\\_visual\\_programming\\_abilityset::training\\_words\\_state\\_cv\\_](#)

## 9.86 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/ultrasonic.cpp 文件参考

```
#include <string>
#include "vpapy.hpp"
#include "cyberdog_vp_abilityset/ultrasonic.hpp"
ultrasonic.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset.py::DefineUltrasonic](#) (py::object m)

## 9.87 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/ultrasonic.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/ultrasonic.hpp"
ultrasonic.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

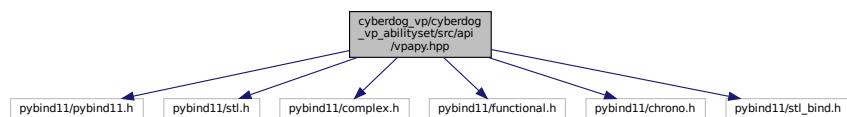
### 变量

- std::mutex [cyberdog\\_visual\\_programming\\_abilityset::ultrasonic\\_data\\_cvm\\_](#)
- std::condition\_variable [cyberdog\\_visual\\_programming\\_abilityset::ultrasonic\\_data\\_cv\\_](#)

## 9.88 cyberdog\_vp/cyberdog\_vp\_abilityset/src/api/vpapy.hpp 文件参考

```
#include <pybind11/pybind11.h>
#include <pybind11/stl.h>
#include <pybind11/complex.h>
#include <pybind11/functional.h>
#include <pybind11/chrono.h>
#include <pybind11/stl_bind.h>
```

vpapy.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



### 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset.py](#)

### 函数

- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineBuiltinInterfaces](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineTimer](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineStdMsgs](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineHeader](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineSensorMsgs](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgRange](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgLaserScan](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgImu](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineGeometryMsgs](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefinePoint](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineQuaternion](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefinePose](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineVector3](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineTwist](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefinePoseWithCovariance](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineTwistWithCovariance](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineNavMsgs](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgOdometry](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineProtocol](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgBmsStatus](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgTouchStatus](#) (py::object m)
- void [cyberdog\\_visual\\_programming\\_abilityset\\_py::DefineMsgConnectorStatus](#) (py::object m)

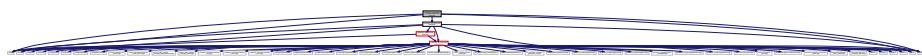
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequenceGait` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequencePace` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgGpsPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgSingleTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgHeadTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgRearTofPayload` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgSport` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgTrainingWords` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvFaceRecResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvSportResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvTrainingWordsResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvAudioTextPlayResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvLedExecuteResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvMotionResultCmdResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvMotionSequenceShowResponse` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineActNavigationResult` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgFaceRes` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgPreset` (py::object m)
- void `cyberdog_visual_programming_abilityset.py::DefineCommonType` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineState` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineAudioPlaySeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineAudioGetVolumeSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineAudioSetVolumeSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineLedConstraint` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineLedSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionResultServiceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionSequenceServiceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionServoCmdResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionParams` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionId` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequenceGaitList` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgMotionSequencePaceList` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMotionSequence` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineObstacleMeta` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineTofObstacle` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineTofPayload` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgPersonnel` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMsgPersonnelList` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineSrvPersonnelResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineFaceSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineFaceRecognizedSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineVoiceprintRecognizedResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineGestureData` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineGestureType` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineGestureRecognizedSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineGestureRecognizedMessageResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineSkeletonType` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineSkeletonRecognizedSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineSkeletonRecognizedMessageResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineTrainingWordsRecognizedSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineTrainingWordsRecognizedMessageResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineMapPresetSeviceResponse` (py::object)
- void `cyberdog_visual_programming_abilityset.py::DefineNavigationActionResponse` (py::object)

- void `cyberdog_visual_programming_abilityset_py::DefineBase` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineNetwork` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineFollow` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineMotion` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineNavigation` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineTask` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineTrain` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefinePersonnel` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineGesture` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineSkeleton` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineAudio` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineLed` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineBms` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineTouch` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineGps` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineTof` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineLidar` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineUltrasonic` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineOdometer` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineImu` (py::object m)
- void `cyberdog_visual_programming_abilityset_py::DefineCyberdog` (py::object m)

## 9.89 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/face.cpp 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/face.hpp"
```

face.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_abilityset`

## 9.90 cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/log.cpp 文件参考

```
#include <string>
#include "cyberdog_vp_abilityset/log.hpp"
```

log.cpp 的引用(Include)关系图:



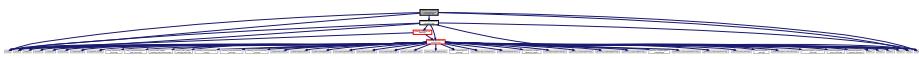
## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 9.91 **cyberdog\_vp/cyberdog\_vp\_abilityset/src/sdk/voiceprint.cpp** 文件参考

```
#include <string>
#include <memory>
#include <vector>
#include <map>
#include "cyberdog_vp_abilityset/voiceprint.hpp"
```

voiceprint.cpp 的引用(Include)关系图:



## 命名空间

- [cyberdog\\_visual\\_programming\\_abilityset](#)

## 9.92 **cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/artificial\_intelligence.hpp** 文件参考

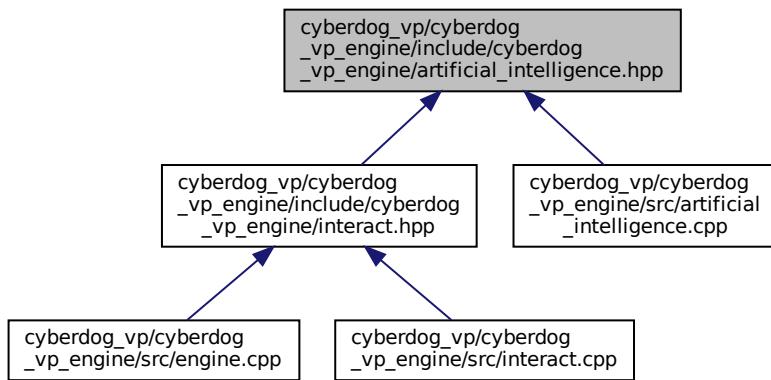
AI模块。

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/common.hpp"
```

artificial\_intelligence.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



类

- class `cyberdog_visual_programming_engine::ArtificialIntelligence`

命名空间

- `cyberdog_visual_programming_engine`

### 9.92.1 详细描述

AI模块。

负责辅助AI功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

相互作用有效。

**Bug** 模块操作尚待调试

警告

留意注册表的稳定性

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

负责辅助SLAM功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

相互作用有效。

**Bug** 模块操作尚待调试

警告

留意注册表的稳定性

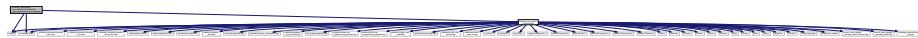
版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

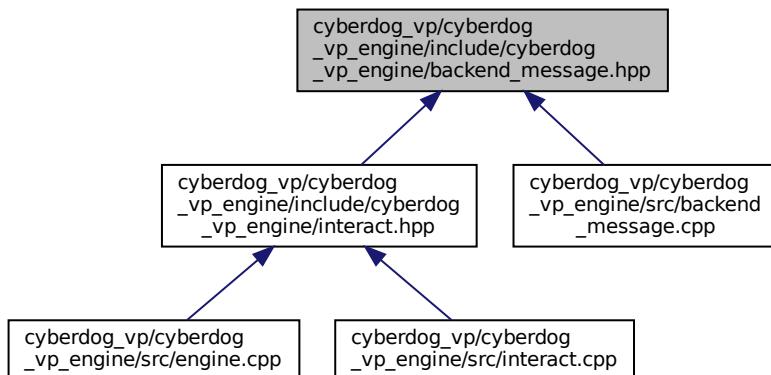
## 9.93 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/backend\_message.hpp 文件参考

消息模块。

```
#include <string>
#include <unordered_map>
#include "cyberdog_vp_engine/common.hpp"
backend_message.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_engine::BackendMessage](#)

命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.93.1 详细描述

消息模块。

负责后端消息的解析及审核。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

注册表有效。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

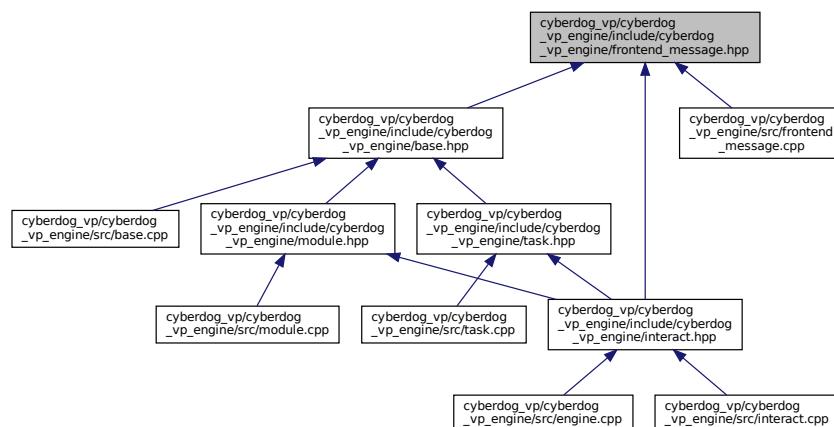
## 9.94 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/frontend\_message.hpp 文件参考

消息模块。

```
#include <string>
#include <unordered_map>
#include "cyberdog_vp_engine/common.hpp"
frontend_message.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_engine::FrontendMessage](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.94.1 详细描述

消息模块。

负责前端消息的解析及审核。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

注册表有效。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.95 `cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/fsm.hpp` 文件参考

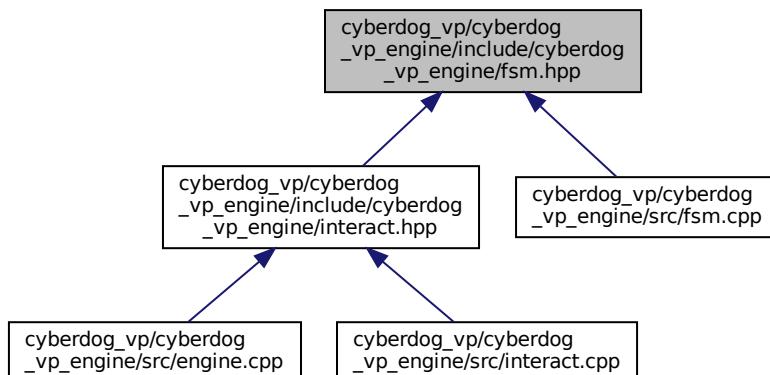
状态机模块。

```
#include <string>
#include <vector>
#include <memory>
#include <unordered_map>
#include "cyberdog_vp_engine/common.hpp"
```

fsm.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_engine::Fsm`

命名空间

- `cyberdog_visual_programming_engine`

### 9.95.1 详细描述

状态机模块。

负责辅助状态机功能。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

相互作用有效。

**Bug** 模块操作尚待调试

警告

留意注册表的稳定性

版权所有

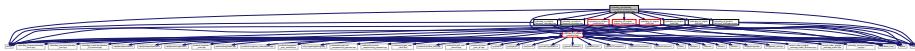
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.96 **cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/interact.hpp** 文件参考

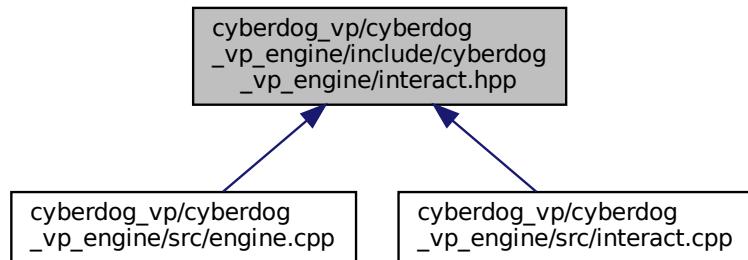
可视化编程引擎相互作用模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_engine/artificial_intelligence.hpp"
#include "cyberdog_vp_engine/backend_message.hpp"
#include "cyberdog_vp_engine/common.hpp"
#include "cyberdog_vp_engine/frontend_message.hpp"
#include "cyberdog_vp_engine/fsm.hpp"
#include "cyberdog_vp_engine/module.hpp"
#include "cyberdog_vp_engine/python_interpreter.hpp"
#include "cyberdog_vp_engine/slam.hpp"
#include "cyberdog_vp_engine/task.hpp"
```

interact.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog\\_visual\\_programming\\_engine::Interact](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.96.1 详细描述

可视化编程引擎相互作用模块。

可视化编程引擎的核心模块，负责消息收发及处理。

## 作者

尚子涵

Shang Zihan

## 版本

1.1.0.0

## 日期

2023-02-06

## 前置条件

初始化ros2。

## 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

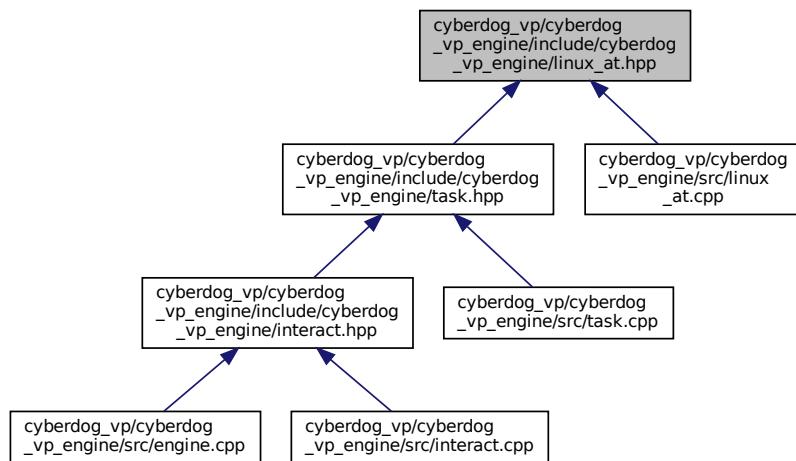
## 9.97 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/linux\_at.hpp 文件参考

单次任务模块。

```
#include <string>
#include <map>
#include "cyberdog_vp_engine/common.hpp"
linux_at.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_engine::LinuxAt](#)

命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.97.1 详细描述

单次任务模块。

负责单次任务的注册。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

注册表有效。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

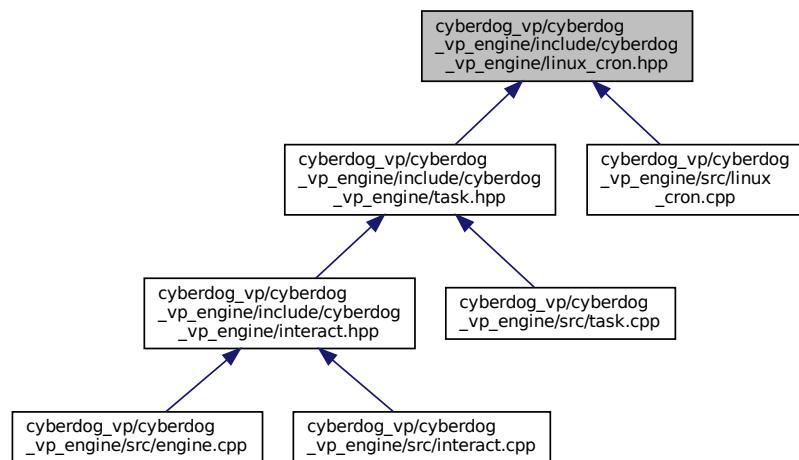
## 9.98 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/linux\_cron.hpp 文件参考

周期任务模块。

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/common.hpp"
linux_cron.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_engine::LinuxCron](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.98.1 详细描述

周期任务模块。

负责周期任务的注册。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

注册表有效。

#### 版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.99 `cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/module.hpp` 文件参考

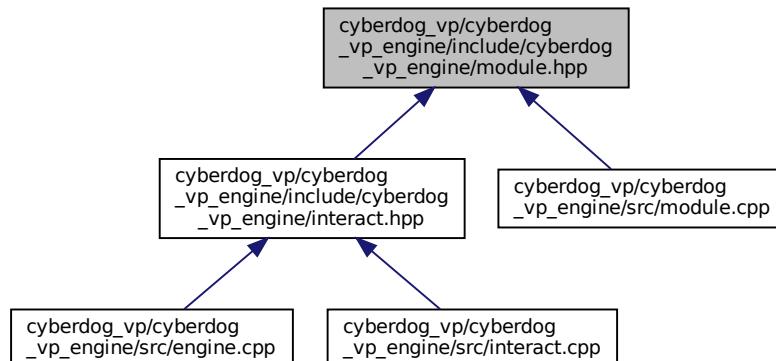
模块编程模块。

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/common.hpp"
#include "cyberdog_vp_engine/base.hpp"
```

module.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_engine::Module`

命名空间

- `cyberdog_visual_programming_engine`

### 9.99.1 详细描述

模块编程模块。

负责模块编程，生成模块文件以便任务编程过程中调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

相互作用有效。

**Bug** 模块操作尚待调试

警告

留意注册表的稳定性

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.100 `cyberdog_vp/cyberdog_vp_engine/include/cyberdog_vp_engine/python_interpreter.hpp` 文件参考

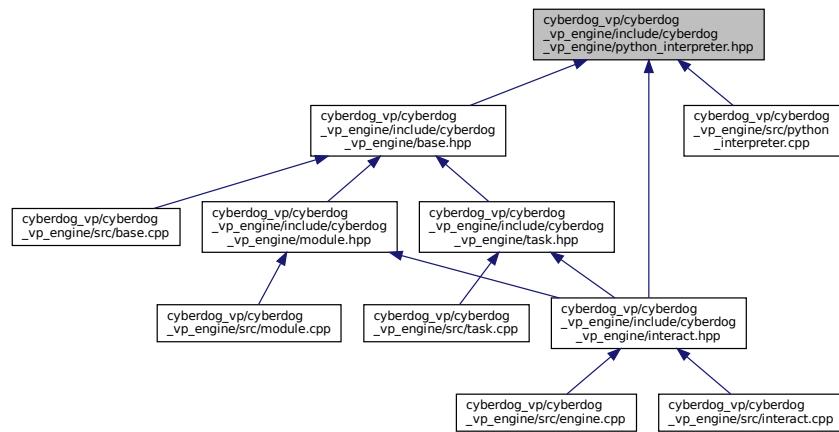
python 解释器模块。

```
#include <string>
#include <map>
#include <vector>
#include "cyberdog_vp_engine/common.hpp"
```

python\_interpreter.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了：



## 类

- class [cyberdog.visual\\_programming\\_engine::PythonInterpreter](#)

## 命名空间

- [cyberdog.visual\\_programming\\_engine](#)

### 9.100.1 详细描述

python 解释器模块。

负责执行 python。

#### 作者

尚子涵

Shang Zihan

#### 版本

1.1.0.0

#### 日期

2023-02-06

#### 前置条件

注册表有效。

#### 版权所有

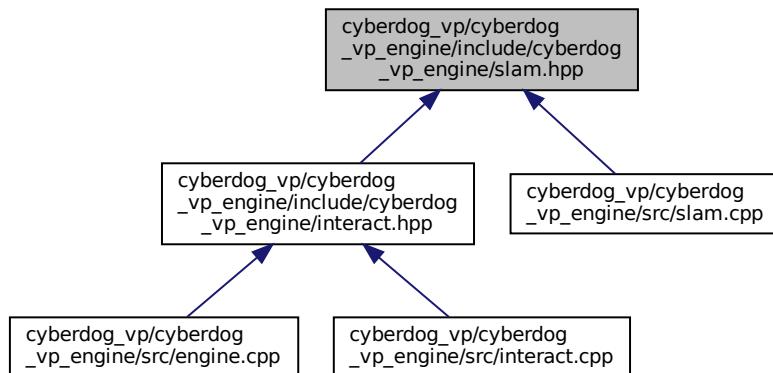
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.101 cyberdog\_vp/cyberdog\_vp\_engine/include/cyberdog\_vp\_engine/slam.hpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/common.hpp"
slam.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



类

- class [cyberdog\\_visual\\_programming\\_engine::Slam](#)

命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

## 9.102 cyberdog\_vp/cyberdog\_vp\_engine/src/artificial\_intelligence.cpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/artificial_intelligence.hpp"
artificial_intelligence.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_engine`

### 9.103 `cyberdog_vp/cyberdog_vp_engine/src/backend_message.cpp` 文件参考

```
#include <string>
#include "cyberdog_vp_engine/backend_message.hpp"
backend_message.cpp 的引用(Include)关系图:
```



## 命名空间

- `cyberdog_visual_programming_engine`

### 9.104 `cyberdog_vp/cyberdog_vp_engine/src/engine.cpp` 文件参考

```
#include <memory>
#include "cyberdog_vp_engine/interact.hpp"
engine.cpp 的引用(Include)关系图:
```



## 函数

- `int main (int argc, char **argv)`

### 9.104.1 函数说明

#### 9.104.1.1 `main()`

```
int main (
    int argc,
    char ** argv )
```

## 9.105 cyberdog\_vp/cyberdog\_vp\_engine/src/frontend\_message.cpp 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/frontend_message.hpp"
frontend_message.cpp 的引用(include)关系图:
```



命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

## 9.106 cyberdog\_vp/cyberdog\_vp\_engine/src/fsm.cpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/fsm.hpp"
fsm.cpp 的引用(include)关系图:
```

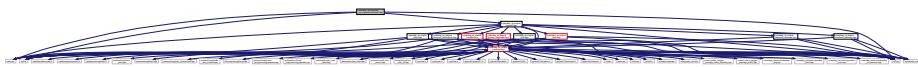


命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

## 9.107 cyberdog\_vp/cyberdog\_vp\_engine/src/interact.cpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/interact.hpp"
interact.cpp 的引用(include)关系图:
```



命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 9.108 `cyberdog_vp/cyberdog_vp_engine/src/linux_at.cpp` 文件参考

```
#include <string>
#include <vector>
#include <map>
#include "cyberdog_vp_engine/linux_at.hpp"
```

linux\_at.cpp 的引用(Include)关系图:



命名空间

- `cyberdog_visual_programming_engine`

### 9.109 `cyberdog_vp/cyberdog_vp_engine/src/linux_cron.cpp` 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/linux_cron.hpp"
```

linux\_cron.cpp 的引用(Include)关系图:



命名空间

- `cyberdog_visual_programming_engine`

### 9.110 `cyberdog_vp/cyberdog_vp_engine/src/module.cpp` 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/module.hpp"
```

module.cpp 的引用(Include)关系图:



命名空间

- `cyberdog_visual_programming_engine`

## 9.111 cyberdog\_vp/cyberdog\_vp\_engine/src/python\_interpreter.cpp 文件参考

```
#include <string>
#include <vector>
#include "cyberdog_vp_engine/python_interpreter.hpp"
python_interpreter.cpp 的引用(Include)关系图:
```



### 命名空间

- [cyberdog\\_visual\\_programming\\_engine](#)

### 变量

- py::object [cyberdog\\_visual\\_programming\\_engine::get\\_module\\_header](#)
- py::object [cyberdog\\_visual\\_programming\\_engine::get\\_task\\_header](#)
- py::object [cyberdog\\_visual\\_programming\\_engine::generate\\_derivative\\_file](#)
- py::object [cyberdog\\_visual\\_programming\\_engine::IPythonDemo](#)

## 9.112 cyberdog\_vp/cyberdog\_vp\_engine/src/slam.cpp 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include "cyberdog_vp_engine/slam.hpp"
slam.cpp 的引用(Include)关系图:
```



### 命名空间

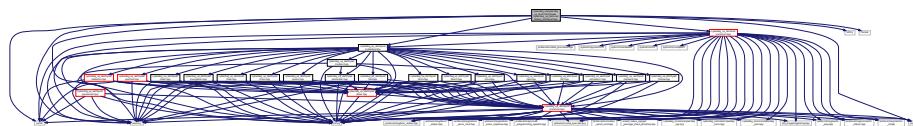
- [cyberdog\\_visual\\_programming\\_engine](#)

## 9.113 cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/debug\_abilityset.hpp 文件参考

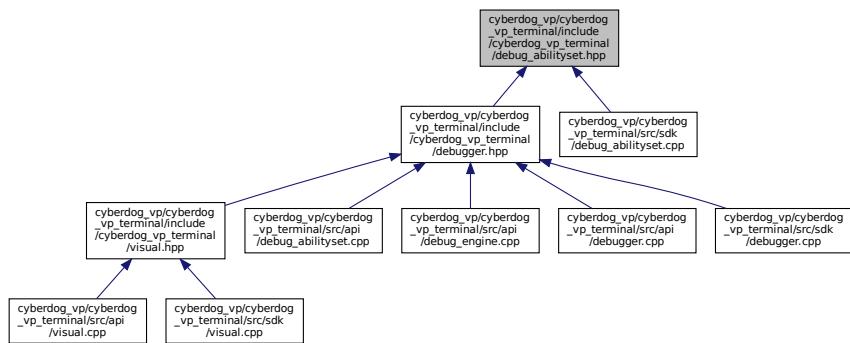
能力集调试器模块。

```
#include <cyberdog_vp_abilityset/cyberdog.hpp>
#include <memory>
#include <string>
#include <vector>
#include <utility>
#include <clocale>
#include <iostream>
#include "cyberdog_vp_terminal/common.hpp"
```

debug\_abilityset.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_terminal::DebugAbilityset](#)

命名空间

- [cyberdog\\_visual\\_programming\\_terminal](#)

### 9.113.1 详细描述

能力集调试器模块。

创建及初始化机器人能力集调试器，以便调试。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

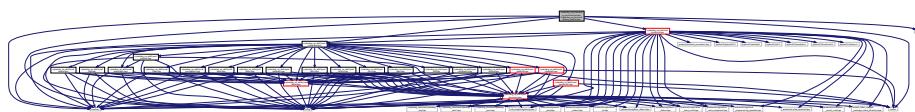
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.114 **cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/debug\_engine.hpp** 文件参考

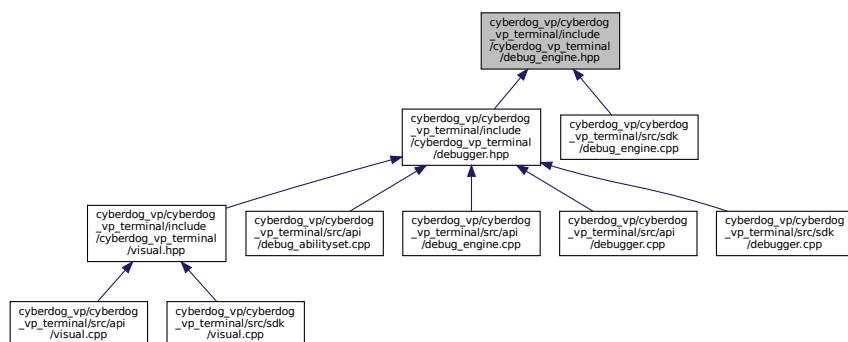
引擎调试器模块。

```
#include <cyberdog_vp_abilityset/cyberdog.hpp>
#include <memory>
#include <string>
#include <vector>
#include <utility>
```

#include "cyberdog\_vp\_terminal/common.hpp"  
 debug\_engine.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_terminal::DebugEngine](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_terminal](#)

### 9.114.1 详细描述

引擎调试器模块。

创建及初始化机器人引擎调试器，以便调试。

## 作者

尚子涵

Shang Zihan

## 版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

注解

确保接口的优化及同步。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.115 cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/debugger.hpp 文件参考

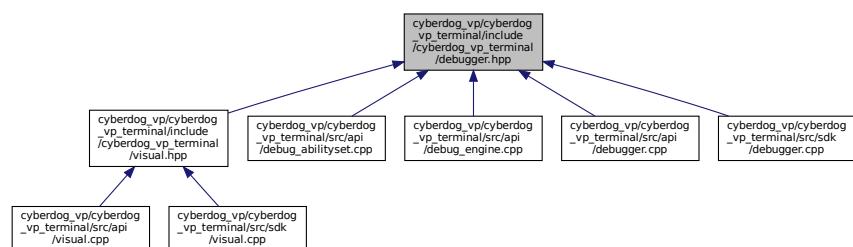
调试器模块。

```
#include <cyberdog_vp_abilityset/cyberdog.hpp>
#include <memory>
#include <string>
#include <vector>
#include <utility>
#include "cyberdog_vp_terminal/common.hpp"
#include "cyberdog_vp_terminal/debug_abilityset.hpp"
#include "cyberdog_vp_terminal/debug_engine.hpp"
```

debugger.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



## 类

- class [cyberdog\\_visual\\_programming\\_terminal::Debugger](#)

## 命名空间

- [cyberdog\\_visual\\_programming\\_terminal](#)

### 9.115.1 详细描述

调试器模块。

创建及初始化机器人调试器，以便调试。

## 作者

尚子涵

Shang Zihan

## 版本

1.1.0.0

## 日期

2023-02-06

## 前置条件

初始化铁蛋模块。

**Bug** 确保各个接口的可用及稳定性。

## 注解

确保接口的优化及同步。

## 版权所有

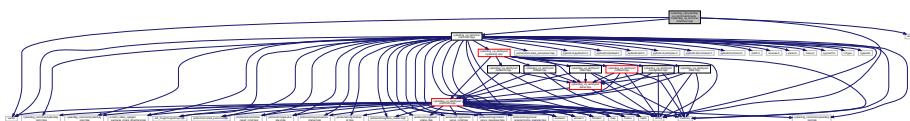
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.116 cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/interface.hpp 文件参考

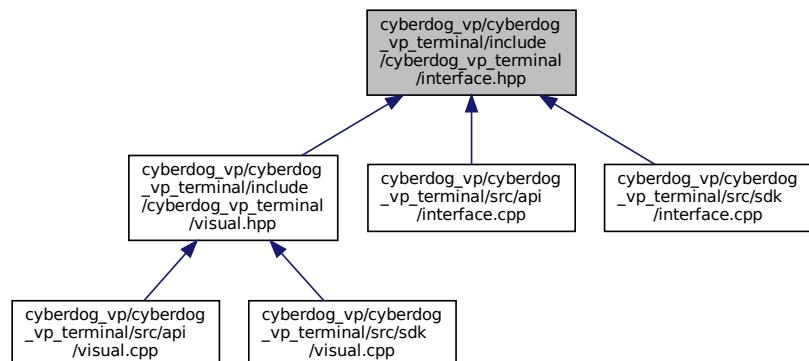
接口模块。

```
#include <string>
#include <vector>
#include <utility>
#include "cyberdog_vp_terminal/common.hpp"
```

interface.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class `cyberdog_visual_programming_terminal::Interface`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Network`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Follow`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Motion`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Navigation`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Task`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Train`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Personnel`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Gesture`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Skeleton`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Bms`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Led`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Audio`

- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Touch`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Gps`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Tof`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Lidar`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Ultrasonic`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Odometer`
- struct `cyberdog_visual_programming_terminal::Interface::Cyberdog::Imu`
- struct `cyberdog_visual_programming_terminal::Interface::Visual`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Interface`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Debugger`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Abilityset`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Debugger::Engine`
- struct `cyberdog_visual_programming_terminal::Interface::Visual::Type`
- struct `cyberdog_visual_programming_terminal::Interface::Type::Enum`
- struct `cyberdog_visual_programming_terminal::Interface::Type::Class`

## 命名空间

- `cyberdog_visual_programming_terminal`

### 9.116.1 详细描述

接口模块。

创建及初始化机器人接口模块，以便测试调用。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

初始化铁蛋模块。

**Bug** 确保接口文档的正确性。

注解

确保接口的持续补充及同步。

版权所有

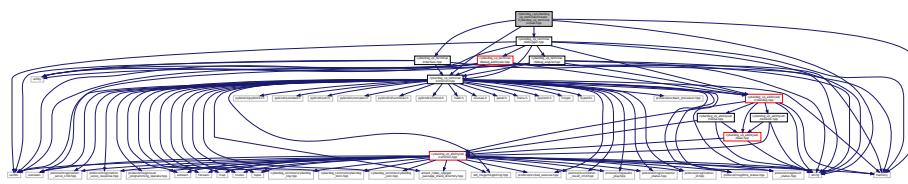
[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.117 cyberdog\_vp/cyberdog\_vp\_terminal/include/cyberdog\_vp\_terminal/visual.hpp 文件参考

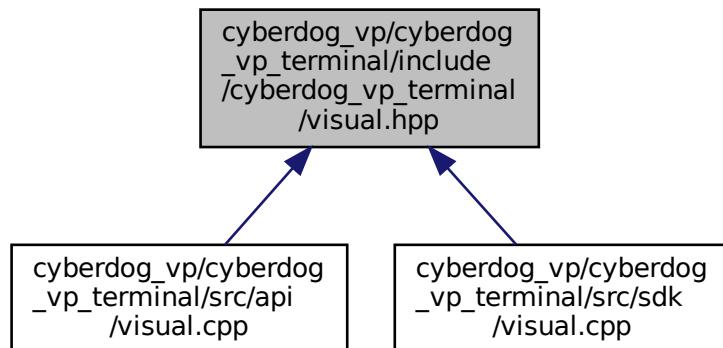
可视化模块。

```
#include <string>
#include <memory>
#include "cyberdog_vp_terminal/common.hpp"
#include "cyberdog_vp_terminal/interface.hpp"
#include "cyberdog_vp_terminal/debugger.hpp"
```

visual.hpp 的引用(Include)关系图:



此图展示该文件直接或间接的被哪些文件引用了:



类

- class [cyberdog\\_visual\\_programming\\_terminal::Visual](#)

命名空间

- [cyberdog\\_visual\\_programming\\_terminal](#)

### 9.117.1 详细描述

可视化模块。

创建及初始化可视化，以便用户编辑。

作者

尚子涵

Shang Zihan

版本

1.1.0.0

日期

2023-02-06

前置条件

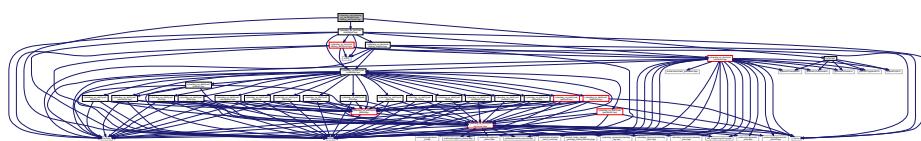
ROS2 环境。

版权所有

[2023]-[2023] [Beijing Xiaomi Mobile Software Co., Ltd. All rights reserved.]

## 9.118 cyberdog\_vp/cyberdog\_vp\_terminal/src/api/debug\_abilityset.cpp 文件参考

```
#include <string>
#include "vptpy.hpp"
#include "cyberdog_vp_terminal/debugger.hpp"
debug_abilityset.cpp 的引用(Include)关系图:
```



命名空间

- [cyberdog\\_visual\\_programming\\_terminal.py](#)

函数

- void [cyberdog\\_visual\\_programming\\_terminal.py::DefineDebugAbilityset \(py::object m\)](#)

## 9.119 cyberdog\_vp/cyberdog\_vp\_terminal/src/sdk/debug\_abilityset.cpp 文件参考

```
#include <memory>
#include <string>
#include <vector>
#include <utility>
#include "cyberdog_vp_terminal/debug_abilityset.hpp"
debug_abilityset.cpp 的引用(Include)关系图:
```



### 命名空间

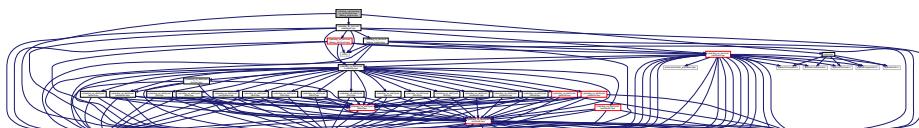
- `cyberdog_visual_programming_terminal`

### 变量

- `std::mutex cyberdog_visual_programming_terminal::cursor_mutex`

## 9.120 cyberdog\_vp/cyberdog\_vp\_terminal/src/api/debug\_engine.cpp 文件 参考

```
#include <string>
#include "vptpy.hpp"
#include "cyberdog_vp_terminal/debugger.hpp"
debug_engine.cpp 的引用(Include)关系图:
```



### 命名空间

- `cyberdog_visual_programming_terminal_py`

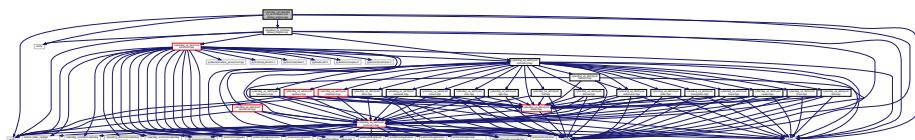
### 函数

- `void cyberdog_visual_programming_terminal_py::DefineDebugEngine (py::object m)`

## 9.121 `cyberdog_vp/cyberdog_vp_terminal/src/sdk/debug_engine.cpp` 文件参考

```
#include <string>
#include <vector>
#include <memory>
#include <utility>
#include "cyberdog_vp_terminal/debug_engine.hpp"
```

debug\_engine.cpp 的引用(Include)关系图:



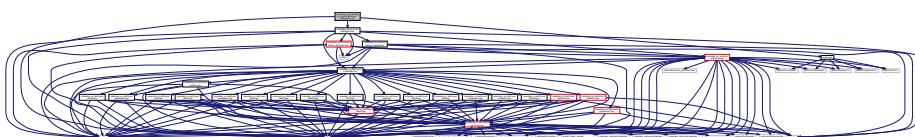
命名空间

- `cyberdog_visual_programming_terminal`

## 9.122 `cyberdog_vp/cyberdog_vp_terminal/src/api/debugger.cpp` 文件参考

```
#include <string>
#include "vptpy.hpp"
#include "cyberdog_vp_terminal/debugger.hpp"
```

debugger.cpp 的引用(Include)关系图:



命名空间

- `cyberdog_visual_programming_terminal_py`

函数

- void `cyberdog_visual_programming_terminal_py::DefineDebugger (py::object m)`

## 9.123 cyberdog\_vp/cyberdog\_vp\_terminal/src/sdk/debugger.cpp 文件参考

```
#include <memory>
#include <string>
#include <vector>
#include <utility>
#include "cyberdog_vp_terminal/debugger.hpp"
```

debugger.cpp 的引用(Include)关系图:



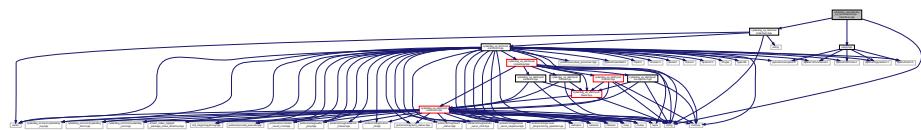
### 命名空间

- `cyberdog_visual_programming_terminal`

## 9.124 cyberdog\_vp/cyberdog\_vp\_terminal/src/api/interface.cpp 文件参考

```
#include <string>
#include "vptpy.hpp"
#include "cyberdog_vp_terminal/interface.hpp"
```

interface.cpp 的引用(Include)关系图:



### 命名空间

- `cyberdog_visual_programming_terminal_py`

### 函数

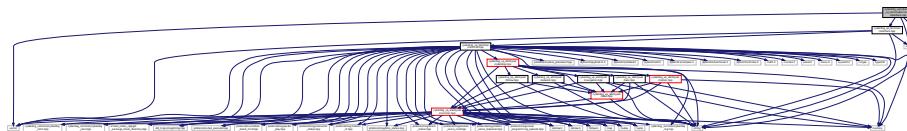
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceNetwork (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceFollow (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceMotion (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceNavigation (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTask (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTrain (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfacePersonnel (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceGesture (py::object m)`
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceSkeleton (py::object m)`

- void `cyberdog_visual_programming_terminal_py::DefineInterfaceBms` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceLed` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceAudio` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTouch` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceGps` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTof` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceLidar` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceUltrasonic` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceOdometer` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceImu` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceCyberdog` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceVisualInterface` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceVisualDebugAbilityset` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceVisualDebuggerEngine` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceVisualDebugger` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceVisual` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTypeEnum` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceTypeClass` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterfaceType` (py::object m)
- void `cyberdog_visual_programming_terminal_py::DefineInterface` (py::object m)

## 9.125 `cyberdog_vp/cyberdog_vp_terminal/src/sdk/interface.cpp` 文件参考

```
#include <string>
#include <vector>
#include <utility>
#include "cyberdog_vp_terminal/interface.hpp"
```

interface.cpp 的引用(Include)关系图:



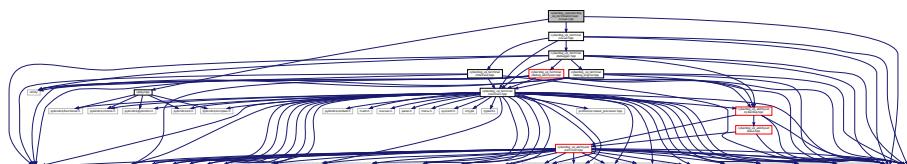
## 命名空间

- `cyberdog_visual_programming_terminal`

## 9.126 `cyberdog_vp/cyberdog_vp_terminal/src/api/visual.cpp` 文件参考

```
#include <string>
#include "vptpy.hpp"
#include "cyberdog_vp_terminal/visual.hpp"
```

visual.cpp 的引用(Include)关系图:



## 命名空间

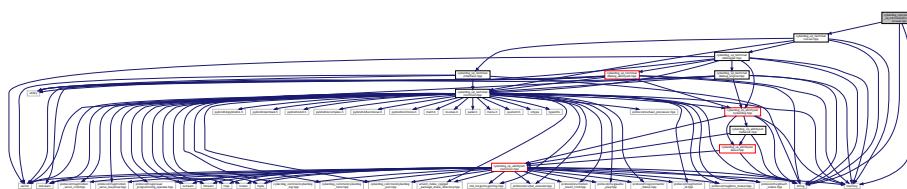
- `cyberdog_visual_programming_terminal.py`

## 函数

- void `cyberdog_visual_programming_terminal.py::DefineVisual (py::object m)`

## 9.127 cyberdog\_vp/cyberdog\_vp\_terminal/src/sdk/visual.cpp 文件参考

```
#include <string>
#include <memory>
#include "cyberdog_vp_terminal/visual.hpp"
visual.cpp 的引用(Include)关系图:
```

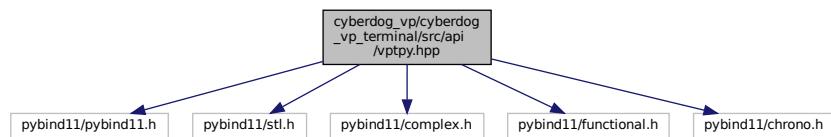


## 命名空间

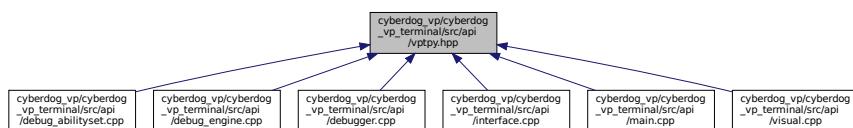
- `cyberdog_visual_programming_terminal`

## 9.128 cyberdog\_vp/cyberdog\_vp\_terminal/src/api/vptpy.hpp 文件参考

```
#include <pybind11/pybind11.h>
#include <pybind11/stl.h>
#include <pybind11/complex.h>
#include <pybind11/functional.h>
#include <pybind11/chrono.h>
vptpy.hpp 的引用(Include)关系图:
```



此图展示该文件直接或间接的被哪些文件引用了：



## 命名空间

- `cyberdog_visual_programming_terminal.py`

## 函数

- void `cyberdog_visual_programming_terminal.py::DefineInterfaceNetwork (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceFollow (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceMotion (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceNavigation (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTask (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTrain (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfacePersonnel (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceGesture (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceSkeleton (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceBms (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceLed (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceAudio (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTouch (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceGps (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTof (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceLidar (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceUltrasonic (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceOdometer (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceImu (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceCyberdog (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceVisualInterface (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceVisualDebugAbilityset (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceVisualDebuggerEngine (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceVisualDebugger (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceVisual (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTypeEnum (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceTypeClass (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterfaceType (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineInterface (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineDebugger (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineDebugAbilityset (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineDebugEngine (py::object m)`
- void `cyberdog_visual_programming_terminal.py::DefineVisual (py::object m)`