C13888 1.00版 / 2018年1月



Zenbo 程序玩家 使用手册

目录内容

第一章: 欢迎使用 Zenbo 程序玩家

语音指令	5
开启方式	5
Zenbo Qrobot 端	5
网页/手机端	6

第二章:程序玩家功能介绍

Zenbo 程序玩家主要功能	7
程序玩家主面板	7
模块菜单(Blocks Menu)	10
工作区(Work Space)	10
Zenbo 面部表情模拟器(Zenbo facial expression Simulator	14
Zenbo 3D 模拟器(Zenbo 3D Simulator)	15
模块功能	16
了解图形模块	16
在 Zenbo 程序玩家中使用模块	18
Zenbo 的头部运动参数	21
Zenbo 的身体运动参数	22

第三章: 模块详情说明

Neck (颈部)	23
Neck Movement(颈部移动)	23
Body (身体)	24
Body Movement(身体移动)	24
Body Rotation(身体转动)	25
Face (脸部)	25
Face Expression(面部表情)	25
Speech (语音)	26
Speak(说话)	26
Audio (声音)	27
Audio Recording(声音录制)	27
Play Recorded Audio(播放录制的声音)	27
Stop Audio Recording(停止声音录制)	27
LED	28
Wheel LED at Non-Blocking(非阻止的轮灯)	28
Stop Wheel LED(停止轮灯)	28



Camera (相机)	.29
Video Recording(影像录制)	.29
Play Recorded Video(播放已录制的影像)	.29
Stop Video Recording(停止录制影像)	.29
Stop Recorded Video(停止播放录制的影像)	.29
Take a Picture(照相)	.30
Show Took Picture(显示刚照的照片)	.30
Detection (检测)	.30
When Face was Detected(当检测到脸部时)	.30
When Not in Charge(当非充电时)	.31
When In Charge(当充电时)	.31
When Battery(当电池)	.31
When Hear(当听到)	.32
Multimedia(多媒体)	.32
Play Media at Non–Blocking(以非阻止方式播放媒体)	. 32
Play Media at Blocking(以阻止方式播放媒体)	. 33
Play Music with Image at Blocking(以阻止方式播放音乐与图片)	. 33
Play Music with Image at Non–Blocking(以非阻止方式播放音乐与图片)	34
Stop Media Playing(停止播放媒体)	.34
Interface (界面)	.35
Tips UI(提示UI界面)	.35
GridView UI(GridView界面)	.36
ListView UI(ListView 界面)	.37
Built-in Movement (智能动作)	.38
One-Time Movement(一次性动作)	.38
Recursive Movement(重复性动作)	.39
Stop Movement (停止动作)	.39
Loop(循环)	.40
Repeat _ timesdo(重复_次执行)	.40
Whiledo(当…执行)	.40
Break(循环中断)	.40
Variable (变量)	.41
Number to Setting(设置的数)	.41
Set to(设为值)	.41
Set to(设为字符串)	.41
Number Variable(数字变量)	.41



String Variable(字符变量)	42
Is In Charge(是充电)	42
Human Body Detected(检测到人体身影)	42
Battery Percentage(电量百分比)	42
Speaker Volume(喇叭音量)	43
Listen(听到)	43
Variable Operation(变量操作)	43
Logic (逻辑)	44
If do(如果…执行)	44
Operation	44
Condition Setting Blocks(条件设置模块)	44
Boolean	44
Event (事件)	45
Broadcast(广播)	45
When receive (当接收到)	45
Event's ID and Media Source(事件 ID 与媒体来源)	45
Event's ID and String(事件 ID 与字串)	46
Hint Content(提示…内容)	46
Others (其他)	46
Start	46
Waiting(sec) (等待(秒))	47
String (字串)	47
Start at(sec)(开始于(秒))	47
Play Media Behavior(播放媒体行为)	48
General Rules (一般规则)	48
Media: Video (媒体:视频)	49
Media: Audio(媒体:声音)	50
Media: Image(媒体:图片)	50
案例学习	52
模块如何执行	53

第四章:程序玩家示例

情境		55
构建一	-个 ZBA APP	57

欢迎使用程序玩家



您可以利用 Zenbo 程序玩家设计工程师专业等级的 Zenbo Qrobot 独特的动作 和表情,并可直接应用到 Zenbo Qrobot 上进行展演。

说明:

- Zenbo 程序玩家网页版网址为: http://appbuilder.asus.com 。
- 若要在手机端使用 Zenbo 程序玩家,请先下载 Zenbo 程序玩家手机版编辑器。您可以直接扫描下方的二维码。



Android



iOS

语音指令

"开启程序玩家"

"打开程序玩家"

"帮我打开程序玩家"

"帮我开启程序玩家"

开启方式

Zenbo Qrobot 端

以下以"开启程序玩家"为例:

- 1. 向 Zenbo Qrobot 说 "嗨, 小布" "或 "Hey Zenbo"。
- 2. 确认 Zenbo Qrobot 脸部左右分別亮起蓝色提示。
- 3. 向 Zenbo Qrobot 说"开启程序玩家"。
- 4. 导入网页 / 手机端编辑好的程序在 Zenbo Qrobot 上展演。

网页/手机端

- 1. 开启网页版或手机版 "Zenbo 程序玩家" 编辑器。
- 点击左侧的列表並拖曳至右边开始设计,使用拖曳和积木堆叠的原理设计 Zenbo的表情、动作、声音、灯光、对话、呈现轮子 LED 灯光效果和移动等, 设计中可自行调整各个变量。
- 3. 保存程序,并将其导入 Zenbo 展演。

程序玩家功能介绍



您可以利用 Zenbo 程序玩家设计工程师专业等级的 Zenbo Qrobot 独特的动作和表情,并可直接应用到 Zenbo Qrobot 上进行展演。

Zenbo 程序玩家主要功能

程序玩家主面板

进入程序玩家主面板,以下项目显示在主面板中: Menu(菜单), Toolbar(工具 栏), Work Space(工作区), Zenbo facial expression Simulator(Zenbo 面部表 情模拟器), Zenbo 3D Simulator(Zenbo 3D 模拟器)和 Blocks Menu(模块菜 单)。



菜单项目说明

菜单	No.	名称	说明
File	1.1	New	创建一个新项目。 若 Work Space(工作区)已保存,则清除当前进 程。若 Work Space(工作区)尚未保存,则创建 一个新项目标签。
	1.2	Open	用文件管理器打开之前保存的扩展名为 .zba 的 项目。
	1.3	Save	保存当前项目
Edit	1.4	Redo	重做在 Work Space(工作区)中的进行的最后 修改。
Luit	1.5	Undo	撤销在 Work Space(工作区)中进行的最后修改。
Connect	1.6 Connect		连接到 Zenbo。 当与 Zenbo 位于同一网域时, 通过输入 Zenbo 的 IP 地址与 Zenbo 连接
1.7 Disconnect		Disconnect	断开与 Zenbo 的连接 断开当前与 Zenbo 的连接
	1.8	General documentation	开始使用
Help	1.9	Documentation about IDE	程序玩家 使用指南
	1.10	API reference	程序玩家 Block (API) 使用指南

Toolbar 项目说明

工具栏	No.	名称	说明			
			创建一个新项[
File	同1.1	New	程。若 Work Space 程。若 Work Space 一个新项目标	26(工作区)已保存,则有际当即进 Space(工作区)尚未保存,则创建 签。		
	同1.3	Save	保存当前项目	保存当前项目		
	同1.4	4 Redo 重做在 Work Space (工作区)中的进行的 改。		Space(工作区)中的进行的最后修		
	同1.5	Undo	撤销在 Work 改。	Space(工作区)中进行的最后修		
1.14 Zo		Zoom in	放大所有模块大小			
Edit	1.15	Zoom out	缩小所有模块大小			
	1.16	Start point	居中并重置所著	有模块的大小		
	1.20	Block Count	计算 Work Sp 包含子模块。	ace (工作区) 中所有模块的数量,		
	1.21	Delete block	将模块拖曳到这里以删除。			
Connect	1 17	Connection	显示 Zenbo 的连接状态。当连接断开时,点击 连接至 Zenbo。			
Connect	1.17	status	当与 Zenbo 位于同一网域时,输入 Zenbo 的 IF 地址可连接至 Zenbo。			
Bun	1.18	Play	Simulate	在 3D Simulator 中执行项目		
nun	1.19	ιαγ	Connect	将项目发送至 Zenbo 并执行		

模块菜单(Blocks Menu)

模块是 Zenbo 程序玩家的核心编程组件。在 Beta 版中,模块分为 17 个类别,包括移动控制、面部表情、讲话、LED 控制、传感器、相机模块、基本逻辑表达,等等。选择一个标签以显示对应的模块(API)。将模块组合和连接起来以创建您特有的 APP。

工作区(Work Space)

工作区(Work Space)是您在 Zenbo 程序玩家中的编程区域。要在 Zenbo 程序 玩家中编程,请选择您需要的模块并将其从 Blocks Menu(模块菜单)拖动到 Work Space(工作区)。



工作区的下拉菜单

在工作区点击鼠标右键可开启下拉菜单。下拉菜单包含一些有用的选项,可帮助 用户写代码。



工作区的下拉菜单项目有:

Undo(撤销):撤销在工作区中的进行的最后修改。Redo(重做):重做在工作区中的进行的最后修改。Delete Blocks(删除模块):删除工作区中的所有模块。Collapse Blocks(折叠模块):折叠以工作区中显示较少模块。Expand Blocks(展开模块):展开以显示工作区中的所有模块。Clean up Blocks(清理模块):重新排列模块布局。

Start		Start
Play Media (Non-Blocking)		Play Media (Non-Blocking) S <
Source 🌔	Media Source	
	Select your media source 🔹	
Volume(%) 100		
Duration(sec) 10		
Start From (sec) 0		
Face Expression	INTERESTED *	
Neck Movement		
Blocking -		
Speed 1 ·		
Turn Vertical(°) 0		
Turn Horizontal(°)		

展开/折叠模块





清理模块

模块的下拉菜单

在模块上点击鼠标右键可开启下拉菜单。下拉菜单包含一些有用的选项,可帮助 用户写代码。



模块的下拉菜单项目有:

Collapse Blocks (折叠模块): 折叠以显示较少模块。

Expand Blocks (展开模块):展开以显示所有模块。

Duplicate (复制):复制选定的模块。

Copy Blocks Group (复制模块组):复制当前选定的模块,包含与之连接的模块。

Add Comment (添加注释): 为模块添加注释。

Remove Comment (删除注释):删除模块的注释。此选项仅在模块已添加注释时 才可用。

Inline Inputs (单行输入): 在一行中显示一个完整的模块

External Inputs (外部输入): 用多行显示一个完整的模块。此选项仅在选择了"单行输入"时可用。



添加/删除注释



Wheel LED at Non-Blocking Side Both • Type Blinking • Display All • Color Brightness(%) 100 Duration(sec) 10

单行或多行显示

Zenbo 面部表情模拟器(Zenbo facial expression Simulator

Face display: Zenbo 面部表情模拟器。在模拟面板上实时显示 Zenbo 的面部表情或 APP 的界面。

Speech info: Zenbo 对话模拟器。以文字信息显示 Zenbo 的响应。

Input text: 文本对话模拟器。输入文字模拟语音对话。



Zenbo 3D 模拟器 (Zenbo 3D Simulator)

Zenbo 3D 模拟器可实时显示编程的头部及身体动作,包括移动、旋转、点头、摇动、LED 灯效等。

相机角度与缩放控制:

在 Zenbo 3D 模拟器中,点击并按住鼠标左键并移动鼠标,可调整相机角度。

在 Zenbo 3D 模拟器中, 滚动鼠标滚轮可进行放大或缩小。



模块功能 了解图形模块

开始模块(Start)

外观:顶部平坦,底部有一个小三角凸起。 特征:顶部不连接其他模块,底部可以连接其他中间模块。 功能:开始



中间模块

外观:顶部有一个小三角凹陷,底部有一个三角形凸起。 特征:连接顶部和底部的其他模块。 功能:表情,说话,等。



独立模块

外观:顶部和底部都平坦且完整。

- 特征:顶部和底部不连接任何模块,内部可以嵌入多个模块。这些模块通常在外部事件被触发时执行。
- 功能:如果侦测到了人脸,或接收到了一个事件等。



复合模块

外观:顶部有三角形凹陷,底部有三角形凸起,内部可以嵌入多个模块。 特征:内部可以嵌入多个模块,与中间模块类似。

功能:重复。



变量模块

外观::左侧有一个凸起的题型块以连接其他模块。

特征:通常用作变量或条件输入。

功能:字符串,说话音量,等等。



在 Zenbo 程序玩家中使用模块

这里列出了各种模块并描述它们的特征。

开始模块

确保使用开始模块作为程序的起始。

动作模块(Non-Blocking/Blocking(阻止/非阻止))

Blocking(阻止):当一个动作完成后再执行下一个命令。

Non-Blocking(非阻止):当一个动作开始后,无论其是否完成,立即执行下一个命令。

示例如下:

若设为 blocking (阻止),则按顺序执行头部和身体移动。

若设为 non-blocking (非阻止),则同时执行头部和身体移动。



阻止与非阻止动作的区别

检测模块

当外部事件被触发时执行一个动作。当检测到外部事件时,当前操作将暂停,直 到外部事件模块完成。



多媒体模块 (Non-Blocking/Blocking(阻止/非阻止))

Blocking(阻止):只有在音乐播放结束后才能运行下一个命令。

Non-Blocking(非阻止):当音乐开始播放时,同时执行内部模块动作。

如果音乐播放完毕,内部模块的动作还没有完成,下一个命令只有在内部模块的动作完成后才会运行。

如果在内部模块的动作完成时音乐仍在播放,则下一个命令将仅在音乐播放结束 时运行。



ListView 模块 (ASUS Common UI)

ListView 拥有以下功能:

Zenbo 的语音交互:提供简单的语音交互,如下图中"Which one do you like"所示。

Package interface: 开发者可以通过定义如下图中的"'I am eventA"所示的简短描述来呈现信息。

ASUS Common Behavior: 通过设置如下图的 Interface Tips 模块所示的简单文本 来实现通用行为的效果。

激活 ListView 后, Zenbo 会发出语音提示,让用户发出命令。如果图12中的界面提示框显示,如果在12秒内没有响应, Zenbo 将显示提示屏幕。

- Tips (提示): 定义一个 Interface Tips 模块来显示提示屏幕。确保 ListView 和 Interface Tips 中的提示名称与下图中的红色方块所示的相同。若没 有定义 Interface Tips,则如果没有响应, ListView 将退出并继续下 一个操作。
- Countdown (倒数): 在 Interface Tips 模块中定义倒计时时间 (Countdown Duration)以使用倒计时。设置为 0 可禁用倒计时页面。
- ListView 选项:点击或使用语音命令选择一个选项,相应的事件模块将会执行。确保 ListView 和事件模块中的名称与下图中的粉色方块所示的名称相同。



变量与逻辑模块

组合这两个模块来进行条件输入。

示例如下:

当电池电量达到一定水平时执行动作,或者当用户说出某些语音串等时执行任务。



Zenbo 的头部运动参数

机器人头部动作的最大角度

方向	最大角度
上	55度
下	15 度
左	45 度
右	45 度

机器人头部的垂直旋转速度

速度	弧度 / 秒	角度 / 秒
L1	0.2443	14
L2	0.7329	42
L3	2.443	140

机器人头部的水平旋转速度

速度	弧度 / 秒	角度 / 秒
L1	0.3141	18
L2	0.57	32.5
L3	2.6175	150

Zenbo 的身体运动参数

机器人身体的移动速度

速度	最大速度(米/秒)
L1	0.1
L2	0.1667
L3	0.2334
L4	0.3001
L5	0.3668
L6	0.4335
L7	0.5

机器人身体的旋转速度

速度	弧度 / 秒	角度 / 秒
L1	0.9423	54
L2	1.5705	90
L3	2.1987	126
L4	2.8269	162
L5	3.4551	198
L6	4.0833	234
L7	4.7115	270





程序玩家使用一个简单的模块来显示 Zenbo 的每个不同的行为功能。您可以将 模块从网页图形界面中拉出,单独使用或组合起来使用。

以下将分别介绍这些行为功能。

Neck (颈部)

Neck Movement (颈部移动)

要求机器人按指定的速度转动它的头部。

Neck Movement
Blocking
Speed 1
Turn Vertical(°) 20
Turn Horizontal(°) 0

参数	
Blocking(阻止)	当前功能执行完毕后执行下一个模块动作。
Non-Blocking(非阻止)	当前功能开始后即开始执行下一个模块动作。
Speed(速度)	1 ~ 3
Vertical (垂直转动)	–15° ~ 55° 正值表示 Zenbo 将向上看。
Horizontal(水平转动)	-45° ~ 45° 正值表示 Zenbo 将向左转头。

Body (身体)

Body Movement (身体移动)

按指定的方向和速度移动一段距离。

Body Movement
Blocking 🔹
Speed 1
Direction Forward *
Distance(cm) 30

参数	
Blocking(阻止)	当前功能执行完毕后执行下一个模块动作。
Non-Blocking (非阻止)	当前功能开始后即开始执行下一个模块动作。
Speed(速度)	1 ~ 7
Direction (方向)	上下左右。 Forward (前进): 向前移动。 Backward (后退): 向后移动。向后转, 然后直行, 然 后再转回身来。 Left (左): 向左转然后直行。 Right (右): 向右转然后直行。
Distance(距离)	相对距离,以厘米为单位。

Body Rotation (身体转动)

按指定的速度旋转一定角度。



参数	
Blocking(阻止)	当前功能执行完毕后执行下一个模块动作。
Non-Blocking(非阻止)	当前功能开始后即开始执行下一个模块动作。
Speed(速度)	1 ~ 7
Angle(角度)	相对转角,以度为单位。

Face (脸部)

Face Expression(面部表情)

设定机器人表情。



参数					
	1. interested	2. doubting	3. proud	4. default	5. happy
	(有兴趣)	(疑惑)	(骄傲)	(预设)	(开心)
	6. expecting	7. shocked	8 .questioning	9. impatient	10. confident
	(期待)	(震惊)	(质疑)	(不耐烦)	(自信)
Face type	11. active	12. pleased	13. helpless	14. serious	15. worried
(面部表情)	(有活力)	(得意)	(无助)	(严肃)	(烦恼)
	16. pretending	17. lazy	18. aware_R	19. tiredZenbo	20. shy
	(预订)	(慵懒)	(察觉_R)	(倦怠)	(害羞)
	21. innocent	22. singing	23. aware_L	24. default_still	25. hide face
	(无辜)	(叹气)	(觉察_L)	(预设_平静)	(隐藏表情)

Speech (语音) Speak(说话)

1. 要求机器人讲出一个给定字符串。

Speak Hi, my name is Zenbo Speed Medium

参数	
Speak(说话)	需要说的句子
Speed(速度)	慢,中,快

2. 要求机器人讲出三个字符串中的一个。

Speak	
Speed Medium -	
String 🏮	String "string1 "
String 🌔	String "string2"
String 🏮	String " string3 "

参数	
Speed(速度)	慢,中,快
String (字串)	备选句子。

Audio (声音) Audio Recording (声音录制)

要求机器人录制声音。

Audio Recording Duration(sec) 10

参数

Duration(时长)

录音时长,以秒为单位

Play Recorded Audio (播放录制的声音)

要求机器人播放刚才录制的声音。

Play Recorded Audio Volume(%) 100 Duration(sec) 10

参数	
Volume(音量)	媒体音量
Duration (时长)	录音播放的时长,以秒为单位

Stop Audio Recording (停止声音录制)

要求机器人停止录制声音。

Stop Audio Recording

参数	
无	

LED

Wheel LED at Non-Blocking (非阻止的轮灯)

控制轮灯。



参数

Side (位置)	左, 右, 或两边 Right (右): 右侧轮灯亮起 Left (左): 左侧轮灯亮起 Both (两边): 两边同时亮起
Type(种类)	LED 模式 闪烁, 呼吸, 字幕, 充电, 恒亮
Display (显示)	开启 LED 的数量。每侧有 8 个LED。 One (一个):开启一个 LED。 Quarter (四分之一):开启两个 LED。 Half (一半):开启四个 LED。 All (全部):开启全部 LED。
Color (颜色)	LED 的颜色 从选色器中选择颜色。
Brightness (亮度)	亮度。0~100,0表示关闭灯效。
Duration (时长)	LED 开启的时长,以秒为单位。

Stop Wheel LED (停止轮灯)

停止轮灯。

Stop Wheel LED

参数 无

Camera (相机) Video Recording (影像录制)

要求机器人录制视频。

Video Recording Duration(sec) 10

~	~~~	
D	uration	(时长)

录像时长,以秒为单位

Play Recorded Video (播放已录制的影像)

要求机器人播放刚才录制的视频。

Play Recorded Video Volume(%) 100 Duration(sec) 10

参数	
Volume(音量)	媒体音量
Duration (时长)	录音播放的时长,以秒为单位

Stop Video Recording (停止录制影像)

要求机器人停止录制视频。

Stop Video Recording

参数	
无	

Stop Recorded Video (停止播放录制的影像)

要求机器人停止播放录制的视频。

Stop Recorded Video

参数

无

Take a Picture(照相)

要求机器人拍照。

Take a Picture

参数	
无	

Show Took Picture (显示刚照的照片)

要求机器人显示刚才拍摄的画面。

Show Took Picture Duration(sec) 10

参数

参数

Duration (时长)

照片显示的时长,以秒为单位。

Detection (检测)

When Face was Detected (当检测到脸部时)

如果机器人检测到人的面部,运行内部模块。



Disired Interval(sec) (间隔时间)

间隔几秒检测一次

When Not in Charge (当非充电时)

如果机器人检测到硬件不在充电,运行内部模块。



When In Charge (当充电时)

如果机器人检测到硬件正在充电,运行内部模块。

When In Charge	
参数	
无	

When Battery (当电池)

如果电池遇到此情况,运行内部模块。



参数	
Condition (条件)	高于,低于
Value(电量)	剩余电量百分比 (0~100)

When Hear (当听到)

如果机器人听到指定的指令,运行内部模块。



参数	
Context	对话系统的输入上下文
Intent Id	让对话状态设置为指定的plan。

Multimedia (多媒体)

Play Media at Non-Blocking (以非阻止方式播放媒体)

播放已选媒体资源,按照给定的值调整音量和时间。

机器人播放媒体的同时会运行内部模块。

Play Media (Non-Blocking)	
Source 🌔	Media Source
	Select your media source 🔹
Volume(%) 100	
Duration(sec) 10	
Start From (sec) 0	

参数	
Source(来源)	您想要播放的文件,可以是视频或音乐文件。
Volume(音量)	媒体音量
Duration (时长)	媒体来源播放的时长,以秒为单位。
Start Form (开始时间)	在指定的时间开始播放媒体内容,以秒为单位。

Play Media at Blocking (以阻止方式播放媒体)

播放已选媒体资源,按照给定的值调整音量和时间。

Play Media (Blocking)	
Source 🌔	Media Source
	Select your media source 🔹
Volume(%) 100	
Duration(sec) 10	
Start From (sec) 0	

参数	
Source(来源)	您想要播放的文件,可以是视频或音乐文件。
Volume(音量)	媒体音量
Duration (时长)	媒体源播放的时长,以秒为单位。
Start Form (开始时间)	在指定的时间开始播放媒体内容,以秒为单位。

Play Music with Image at Blocking(以阻止方式播放音乐 与图片)

播放已选媒体资源,按照给定的值调整音量和时间。

Play Music with Image (Blocking)	
Music Source 🌘	Music Source
	Select your media source 🕤
Image Source 🌘	Image Source
	Select your media source 🔹
Volume(%) 100	
Duration(sec) 10	
Start From (sec) 0	

参数	
Music Source (音乐来源)	您想要播放的音乐
Image Source(图片来源)	您想要显示的图片。
Volume(音量)	媒体音量
Duration (时长)	媒体源播放的时长,以秒为单位。
Start Form (开始时间)	在指定的时间开始播放音乐,以秒为单位。

Play Music with Image at Non-Blocking (以非阻止方式播放音乐与图片)

播放已选媒体资源,按照给定的值调整音量和时间。

机器人播放媒体的同时会运行内部模块。



参数	
Music Source (音乐来源)	您想要播放的音乐。
Image Source(图片来源)	您想要显示的图片。
Volume(音量)	媒体音量。
Duration (时长)	媒体源播放的时长,以秒为单位。
Start Form (开始时间)	在指定的时间开始播放音乐,以秒为单位。

Stop Media Playing (停止播放媒体)

停止播放媒体资源。



· 参数	
无	

Interface(界面)

Tips UI (提示 UI 界面)

显示内置用户界面以提示用户如何命令机器人。



参数	
Tips UI's ID (提示 UI 界面的 ID)	Tips UI 的 ID
Countdown UI Title (倒数 UI 界面的标题)	提示页面标题
Countdown UI Duration (倒数 UI 界面持续时间)	内置使用界面将会显示的时长,以秒为 单位
ltem (项目)	提示列表 Hint (提示) : 命令子标题。 Content (内容) : 命令的详细说明。

GridView UI (GridView 界面)

在屏幕上显示网格视图,在视图中显示相关字串,通过语音或点击来选择项目。



参数			
Text-To-Speech(文字转语音)	当网格视图出现时让机器人说话		
Title(标题)	在 GridView 界面上方显示的标题		
Intent Id	语音命令设置: Plans Id (计划 Id): 对话系统 (DS) 中的 Plans id Value (值): 对话系统 (DS) 中 Intent 的 Belief 设置 Context (背景): 对话系统 (DS) 中 Plans id 的输入		
Tips UI's ID (提示 UI 界面的 ID)	定义将执行哪些提示。		
Event (事件)	Event's ID (事件 ID): 当选定项目后, 调用用户想要的 事件 (event)。 Media Source (媒体来源): 选择您想要显示的媒体 来源。		

ListView UI (ListView 界面)

在屏幕上显示列表视图, 在视图上显示相关字符, 通过语音或点击来选择项目。 每种项目的行为将由事件接收模块决定。



参数	
Text-To-Speech(文字转语音)	当列表视图出现时让机器人说话
Title(标题)	在列表视图上方显示的标题
	语音命令设置:
	Plans Id (计划 Id):对话系统 (DS) 中的 Plans id
Intent Id	Value (值): 对话系统 (DS) 中 Intent 的 Belief 设置
	Context (背景) : 对话系统 (DS) 中 Plans id 的输入 上下文
Tips UI's ID (提示 UI 界面的 ID)	定义将执行哪些提示。
Event (事件)	Event's ID (事件 ID) :当选定项目后,调用用户想要的事件(event)。
	Event string (字串): 坝目上显示的又字。

Built-in Movement (智能动作) One-Time Movement(一次性动作)

要求机器人做特定的动作。

One-Time Movement Default_1

参数					
	0. Default_1 (预设_1)	1. Default_2 (预设_2)	2. Nod_1(点 头_1)	3. Head_up_1 (头部_向上 _1)	4. Head_up_2 (头部_向上 _2)
	5. Shake_ head_1 (摇_ 头_1)	6. Head_up_3 (头部_向上 _3)	7. Head_up_4 (头部_向上 _4)	8. Head_ down_1 (头部_ 向下_1)	9. Head_ down_2 (头部 _向下_2)
	10. Head_ down_3 (头 部_向下_3)	11. Shake_ head_2(摇摆 头部_2)	12. Head_ down_4 (头部 _向下_4)	13. Head_up_5 (头部_向上 _5)	14. Head_ down_5 (头部 _向下_5)
Action type (动作类型) (动作类型) 23 tw 体, 44 rig 转, 15 rig rev (7 转,	16. Head_ up_7 (头部_ 向上_7)	18. Turn_ left_1 (左_转 _1)	19. Turn_ left_2 (左_转 _2)	20. Shake_ head_3 (摇_头 _3)	22. Body_ twist_1(身体_ 扭曲_1)
	23. Body_ twist_2(身 体_扭曲_2)	27. Dance_3 (跳舞_3)	28. Shake_ head_5 (摇_头 _5)	42. Head_ down_6 (头部_ 向下_6)	43. Head_ down_7 (头部 _向下_7)
	44. Turn_ right_1 (右_ 转_1)	45. Turn_ right_2 (右_ 转_2)	46. Turn_left_ reverse_1(左_ 转_翻转_1)	47. Turn_right_ reverse_1(右_ 转_翻转_1)	48. Turn_left_ reverse_2(左 _转_翻转_2)
	49.Turn_ right_ reverse_2 (右_转_翻 转_2)	54. Head_ up_6 (

Recursive Movement (重复性动作)

要求机器人做特定的动作。

该动作将不会间断直到接收到"停止指定动作"的要求。还可以使用如"逻辑"或 "检测"等其他模块来中止动作。

Recursive Movement	Dance_b_1 *
Dur	ation(sec) 10

参数			
Action type	15. Dance_b_1 (跳舞_b_1)	17. Music_1 (音乐_1)	21. Dance_s_1 (跳舞_1)
(动作类型)	24. Dance_2	25. Shake_head_4	26. Head_twist_1
	(跳舞_1)	(摇_头_4)	(头部_晃动_1)

Stop Movement (停止动作)

停止运行特定的行为。

Stop Movement

参数		
无		

Loop(循环)

Repeat _ times...do(重复_次执行)

您希望播放几次内部模块。





您想要重复的次数

While...do(当…执行)

当条件为真时,重复运行内部模块。





Break(循环中断)

中断包含的循环。



Variable(变量)

Number to Setting (设置的数)

整数类型的值。



参数 设置值

Set ... to (设为值)

变量设置。将值传到指定变量中。

Set myVariable 🕤 to	b 100 J
---------------------	---------

参数	
mullariable (我的亦是)	变量名称
IIIyvuluble(我的支重)	您也可以从下拉菜单中添加一个新的变量。

Set ... to (设为字符串)

变量设置。将字符串传到指定变量中。

Set myVariable → to " n	nyVariableString
参数	
myVariable(我的变量)	变量名称 您也可以从下拉菜单中添加一个新的变量。

Number Variable (数字变量)

返回此变量的值。

Number Variable myVariable -

ALMA,	参	数
	_	

设置值

String Variable (字符变量)

返回此变量的字符串。

String Variable " myVariable - "

参数	
无	

Is In Charge (是充电)

检查机器人是否充电。



参数	
In Charge(充电)	检查机器人是否在充电。如果在充电,则模块为真。
Not In Charge (非充电)	检查机器人是否在充电。如果没有在充电,则模块为 真。

Human Body Detected (检测到人体身影)

检查是否检测到人物。

Human Body Detected

参数	
无	

Battery Percentage(电量百分比)

获得电池电量百分比。

Battery Percentage

参数	
无	

Speaker Volume (喇叭音量)

获得文字转语音的音量。

Speaker Volume

参数		
无		

Listen(听到)

要求机器人听命令。

9	Listen	== *	Con	ntext (list_myinput
1			Plans Id	bloc	klytest.mylist

参数	
liston (I匠列)	== : 检查两个操作数的值是否相等。如果相等, 则条件成立。
	!=:检查两个操作数的值是否相等。如果不相等,则 条件成立。
	语音命令设置
Intent Id	Plans Id: 让对话状态设为特定的 plan。
	Context: DS 系统中 Plans ID 的输入上下文

Variable Operation (变量操作)

从变量中添加或减去1。



Logic(逻辑)

If ... do (如果…执行)

如果条件为真,运行内部模块。



参数

其他模块的条件设置,如"变量"与"其他"模块。

Operation

调用逻辑 AND 运算符。如果操作数为非零,则条件为真。



参数

其他模块的条件设置,如"变量"与"其他"模块。

Condition Setting Blocks (条件设置模块)

如果第一个变量与第二个变量相等,运行内部模块。



参数

其他模块的条件设置,如"变量"与"其他"模块。

Boolean

布尔类型(boolean type)的值。



Event (事件)

Broadcast (广播)

事件行为可以在"当接收时"模块中定义。

Broadcast eventId

参数

调用用户需要的特定事件。

When receive (当接收到)

当"广播"以相应的事件 ID 运行时,模块将会被触发。

W	hen receive	eventId
参数		
Event's	; ID	

Event's ID and Media Source (事件 ID 与媒体来源)

网格视图扩展的一个元素。调用选择项目时用户需要的事件。

Event's	ID eventle	d	
			Media Source
			Select your media source 🔹

参数	
Event's ID (事件 ID)	当选择一个项目时,将调用用户需要的事件。
Media Source(媒体来源)	项目上要显示的图片

Event's ID and String (事件 ID 与字串)

列表视图扩展的一个元素。调用选择项目时用户需要的事件。

	Event's ID	eventId			
			String "	ListView show string) ** -
参数					

Event's ID (事件 ID)	当选择一个项目时,将调用用户想要的事件。
String (字串)	项目上要显示的文字。

Hint ... Content (提示…内容)

提示视图中的提示和内容。

Hint my hint Content my content

参数	
Hint (提示)	命令的子标题
Content (内容)	命令的详细说明。

Others(其他)

Start

项目的入口点。其显示将要执行的模块堆栈。



Waiting(sec) (等待(秒))

在指定的秒数内不执行任何操作。此模块与上一个模块相关。



Start at(sec)(开始于(秒))

模块开始于特定的时间。这些模块就是单纯的带时间标签的开始模块。

Start at	10	secs

参数

指定的时间点以执行下一个模块。

Play Media Behavior (播放媒体行为)

注意:

Play Media Behavior 的定义将在 v1.0.3.20170821后启动。

General Rules (一般规则)

子模块将在父模块设置的持续时间内运行。

如果父模块设置的时间到了,子模块仍没有执行完,Zenbo会停止当前执行的操作。



如果有其他的播放媒体(非阻止)模块作为子模块,持续时长(Duration)即为媒体的播放时长。如果父模块的持续时长已经到了,则即使子模块还未完成,也将停止执行。



Media: Video (媒体:视频)



Screen Output(屏幕输出)

如果视频播放完毕后还有剩余时间, 屏幕将显示使用屏幕的模块(例如, Face Expression(面部表情), Take Picture(照相), Show Picture(显示照片),等)。如果没有其他的模块需要屏幕, 屏幕将显示Zenbo预设的脸部表情。

Audio Output(声音输出)

仅声音模块(例如, Speach(说话), Play Recorded Audio(播放录制的声音), Play Media [Audio](播放媒体[音频]))可以与父模块同时运行。

不能同时执行的模块有:

- Play Media (Blocking/Non-Blocking) [Image/Video] (播放媒体(阻止/非阻止)[图片/视频])
- Play Music with Image (Blocking/Non-Blocking)(播放音乐与图片(阻止/非 阻止))
- Video Recording (影像录制)
- Play Recorded Video(播放已录制的影像)
- Take Picture (照相)
- Show Took Picture (显示刚照的照片)
- GridView/ListView (GridView/ListView界面)

这些模块只会在父模块中的视频文件播放完毕并且还有剩余的时间时才会运行。

Media: Audio (媒体: 声音)

仅使用音频的模块可以与 Play Media(Non-Blocking) [Audio] 模块同时执行。如: Speak (说话), GridView/ListView, Take Picture (照相), 等。



Media: Image (媒体: 图片)



使用屏幕输出的模块不能与Play Media (Non-Blocking) [Image] (播放媒体(非阻止)[图片]) 模块同时执行。

不能同时执行的模块有:

- Play Media (Blocking/Non-Blocking) [Image/Video]
- Play Music with Image (Blocking/Non-Blocking)
- Take Picture
- Show Took Picture

- Video Recording
- Play Recorded Video
- GridView/ListView
- Play Media (Blocking/Non-Blocking) [Image/Video] (播放媒体(阻止/非阻止)[图片/视频])
- Play Music with Image (Blocking/Non-Blocking)(播放音乐与图片(阻止/非 阻止))
- Take Picture (照相)
- Show Took Picture (显示刚照的照片)
- Video Recording (影像录制)
- Play Recorded Video(播放已录制的影像)
- GridView/ListView(GridView/ListView界面)



模块组合情况如下。本章节将带您了解此模块组合的执行行为。



模块如何执行

Screen Output	Body Output	Wheel Output	Audio Output
Play Media (Non-Blocking) Source Volume(%) [10] Duration(sec) Start From (sec)]	Body Mavement Blocking = Speed = Direction Forward = Destance(on) 1000 Body Movement Blocking = Speed = Direction Forward = Direction = Forward = Direction = 30	Wheel LED at Non-Blocking Side Bother Type Blinkinger Display All Color Brightmess(%) 100 Duration(sec) 10	Play Media (Non-Blocking) Source II demoVideo mp4 23 Volume(%) [100] Duration(sec) [] Start From (sec) []

模块不是按照组合的顺序来执行的,而是根据输出进行安排,并同时运行。但是, 同时还有其他规则可能会影响此行为。



在Screen Output(屏幕输出)下,除了 Play Media(播放媒体),还有 Face Expression(面部表情)。

如果视频文件的时长不到 5 秒,则剩余时长中,屏幕会显示 Face Expression (面 部表情)。

如果视频文件时长大于 5 秒,则 Face Expression (面部表情)不会显示。

Sti	art		_
Pla	ay M	edia (Non-Blocking)	
		Source	Media Source
			demoVideo.mp4 -
		Volume(%) 100	
		Duration(sec) 5	
		Start From (sec)	
	Ē	Body Movement	—
		Blocking -	
		Speed 1	
	Dir	ection Forward -	
	Di	stance(cm) 1000	
	¢	Face Express	ion EXPECTING
		Body Movement	
		Blocking -	
		Speed 1	
	Din	ection Forward -	
		Distance(cm) 30	
	Wh	eel LED at Non-Bloc	king
		Side Bo	oth 👻
		Type Blinki	ng 👻
		Display 🛛	All -
		Colo	
		Brightness(%)	100
		Duration(sec)	10
<u> </u>			

如果有模块设置为 "Blocking(阻止)" 类型,则其下面的模块必须等到该模块 执行后才能执行。

在这里的例子中,执行的模块如下:

- Play Media (Non-Blocking) (以非阻止方式播放媒体)播放, Body Movement (身体移动)执行
- 当 Play Media (Non-Blocking)(以非阻止方式播放媒体)结束后,显示 Face Expression(面部表情)然后继续执行Body Movement(身体移动)(如果之前的身体移动尚未完成,则添加到上一个身体移动)
- 当第二个 Body Movement(身体移动) 完成, 点亮 Wheel LED at Non-Blocking(非阻止的轮灯)。

程序玩家示例



本章节的示例教您如何使用程序玩家来创建您的第一个 Zenbo 应用程序。您将 学习如何使用程序玩家创建项目并在 Zenbo 模拟器中运行它。在部署到Zenbo 设备之前,您需要将项目导出为 zba 形式文件。您亦可学习 Zenbo 设计的一些基 础知识,包括如何构建简单的用户界面,多媒体播放器,移动 Zenbo 身体以及处 理对话。

情境

Zenbo 开始播放音乐并闪烁轮灯 LED。他向前移动了 10 厘米, 然后转了一圈, 抬起头来。

Zenbo: '嗨, 我是Zenbo'.

Zenbo 邀请某人加入他的表演。

Zenbo: '想听音乐还是看影片呢?'

Zenbo 在屏幕上显示选项供用户选择。

Someone: 'Music' (或 'Movie')

Zenbo 开始根据选择播放音乐或电影。在这一步中,如果您计划让用户通过语音 命令来选择项目,请在 DDE 系统中定义语句。



构建一个 ZBA APP

1. 新建项目并选择其默认语言。

File Edit Connection Help		English
	CR Disconnected 🕨 00:00.00	Block(s):1
Neck		
Body		
• Face	System language $ imes$	
Speech		
• Audio	* APP Name	
• LED	New APP	
• Camera	Language	
Detection	 正體中文 	Talk to Zenbo
Multimedia	Cancel	
 Interface 		
 Built-in Movement 		
Loop		
Zenbo		

2. 根据情境分析组合模块, 您也可以下载示例, 然后将其导入。

File Edit Connection	Help		English
		00:00.00	
Neck	Play Media (Non-Blocking)		
 Body 	Source Media So	urce	
Face	Volume(%) 20	When receive music	
Speech	Duration(sec) 30 Start From (sec) 0	Play Media (Non-Blocking) Source Media 5	
• Audio	Face Expression HAPPY	Select your me	
• LED	Wheel LED at Non-Blocking	Duration(sec) 100	
Camera	Side Both - Type Blinking -	Recursive Movement Dance_b_1	
Detection	Display All • Color 🛑	Duration(sec) (30) Stop Media Playing	Talk to Zenbo
Multimedia	Brightness(%) 80 Duration(sec) 60	One-Time Movement Default_1 +	\bigcirc
Interface	Body Movement	When receive Video	
Built-in Movement	Speed 1	Play Media (Blocking) Source Media Sour	
• Loop	Distance(cm) 10	Select your media	
	Body Rotation Blocking	Duration(sec) (30)	
Zenbo	Speed 1	Start From (sec) 0 Stop Media Playing	

3. 保存 project (.zba) 文件

请填写 package name, App ID, version code, domain UUID, domain version 等 APP 相关信息, 然后保存您的项目。

File Edit Connection Help			
	Q ⊕ / Usconnected	00:00.00	
Neck Pla	Save P	roject	×
Body		Languago	
Face	* APP Name	 English 	
Speech	ZBAHelloWorld	 正體中文 	
Audio	Preferences		
• LED	* Package Name	Icon	
Camera	com.asus.robot.appbuilder. zba.hell		
Detection	* APP ID LzMS9ZSfsmt26mlW	Icon has a maximum size of 256x256 pixels	to Zenbo 💬
Multimedia	* Version Name	* Version Code	
Interface	0.0.1	000001	
 Built-in Movement 	Domain UUID	Domain Version	
Loop	CDA3B30FD29A48379C9927880	0.0.1	
Zenbo		Cancel OK	

4. 在 Zenbo 模拟器上运行 APP。



5. 在 Zenbo 上运行 APP

通过IP连接到 Zenbo

- 在 Android 设置中的"关于平板电脑"部分找到 Zenbo 的 IP地址。
- 在 Connection 对话框中输入 Zenbo 的 IP 地址。
- 连接后, APP 将传送到 Zenbo。

