用户手册



# Al-Starter 机器人用户手册

文档版本: V1 发布日期: 2019-01-07

深圳市越疆科技有限公司



#### 版权所有 C 越疆科技有限公司2017。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

#### 免责申明

在法律允许的最大范围内,本手册所描述的产品(含其硬件、软件、固件等)均"按照现状"提供,可能存在瑕疵、错误或故障,越疆不提供任何形式的明示或默示保证,包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证;亦不对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害进行赔偿。

在使用本产品前详细阅读本使用手册及网上发布的相关技术文档并了解相关信息,确保 在充分了解机器人及其相关知识的前提下使用机械臂。越疆建议您在专业人员的指导下使用 本手册。该手册所包含的所有安全方面的信息都不得视为Dobot的保证,即便遵循本手册及 相关说明,使用过程中造成的危害或损失依然有可能发生。

本产品的使用者有责任确保遵循相关国家的切实可行的法律法规,确保在越疆机械臂的 使用中不存在任何重大危险。

# 越疆科技有限公司

地址: 深圳市南山区同富裕工业城三栋三楼

网址: <u>http://cn.dobot.cc/</u>

用户手册



### 前 言

# 前 言

# 目的

本手册介绍了如何使用智能小车,使用户在使用时能挖掘更多功能,帮助用户快速上手。

# 读者对象

本手册适用于:

- 客户工程师
- 销售工程师
- 安装调测工程师
- 技术支持工程师

# 修订记录

时间	修订记录
2019/01/07	V1.1.0 增加小车demo,更新AIP
2017/9/26	V1.0.0 第一次发布

# 符号约定

在本手册中可能出现下列标志,它们所代表的含义如下。

符号	说明
<b>企</b>	表示有高度潜在危险,如果不能避免,会导致人员 死亡或严重伤害
▲警告	表示有中度或低度潜在危害,如果不能避免,可能 导致人员轻微伤害、机械臂毁坏等情况
▲ 注意	表示有潜在风险,如果忽视这些文本,可能导致机 械臂损坏、数据丢失或不可预知的结果
□□说明	表示是正文的附加信息,是对正文的强调和补充



# 目 录

1.	注意	事项	1
	1.1	使用安全	.1
	1.2	售后条款	.1
		1.2.1 质保细则	.1
2.	产品	简介	3
	2.1	产品特性	.3
	2.2	零件清单	.3
	2.3	技术规格	.4
		2.3.1 技术参数	.4
3.	特性	说明	6
	3.1	AI-Starter 控制器	.6
		3.1.1 概述	.6
		3.1.2 AT Mega2560 处理器	.6
		3.1.3 按键	.6
		3.1.4 LED	.7
		3.1.5 USB	.7
		3.1.6 接口说明	.8
	3.2	红外巡线传感器1	0
	3.3	超声波传感器1	0
	3.4	颜色传感器1	0
4.	安装	指南1	1
	4.1	Mixly 软件安装1	1
	4.2	Arduino IDE1	1
5.	使用	指南1	2
	5.1	Mixly 使用说明1	2
	5.2	Arduino IDE 使用说明1	2
6.	AI-St	arter Demo1	4
	6.1	自动巡线 Demo1	4
		6.1.1 介绍1	4
		6.1.2 操作步骤1	4
		6.1.3 关键代码说明1	4
	6.2	自动避障 Demo1	6
		6.2.1 介绍1	6
		6.2.2 操作步骤1	6
		6.2.3 关键代码说明1	6
	6.3	白平衡校准 Demo1	8
		6.3.1 介绍1	8
		6.3.2 操作步骤1	9
		6.3.3 关键代码说明1	9
	6.4	颜色识别与自动巡线 Demo1	9
		6.4.1 介绍1	9
		6.4.2 操作步骤1	9
	文档	版本 V1.1.0(2019-01-07) 用户手册 版权所有 © 越疆科技有限公	司

		6.4.3	关键代码说明			19
	6.5	机械臂	协作 Demo			20
		6.5.1	介绍			20
		6.5.2	操作步骤			21
		6.5.3	关键代码说明			21
	6.6	舵机驱	动 Demo			23
		6.6.1	介绍			23
		6.6.2	操作步骤			23
		6.6.3	关键代码说明			23
	6.7	Xbee 通	信 Demo			24
		6.7.1	介绍			24
		6.7.2	操作步骤			24
		6.7.3	关键代码说明			24
7. A	API 持	妾口	••••••			
	7.1	初始化				26
	7.2	设置小	车速度			26
	7.3	设置小	车运动方向/速度	/时间		
	7.4	设置电	机转速			27
	7.5	设置电	机参数			27
	7.6	获取电	机位姿			27
	7.7	初始化	超声波传感器			
	7.8	获取超	声波探测距离			
	7.9	探测障碍	碍物			
	7.10	获取巡	线数据			29
	7.11	获取地	磁角度			
	7.12	地磁校	准			
	7.13	设置颜	色白平衡			
	7.14	启停颜	色传感器			
	7.15	检测颜	色			
	7.16	获取 RC	GB 色值			
	7.17	初始化	按键			
	7.18	获取按	键状态			
	7.19	设置灯	光状态			
	7.20	获取光	敏数值	·····		
	7.21	设置超	声波传感器检测。	上呙		
	7.22	连接舵	机			
	7.23	设置舵	机角度			
	7.24	断开舵	机连接			
	7.25	开启后	台定时器任务			
	7.26	关闭后	台定时器任务			
	7.27	Xbee 数	[据读取			
	7.28	Xbee 数	[据写入			35
	7.29	Xbee 数	(据对比			35
	7.30	Xbee 数	[据清除			35
	文档	钣木 V1 1	0 (2019-01-07)	田户手册	版权所有	◎ 栽疆科技有限公司



# 1. 注意事项

# 1.1 使用安全

- 安装电池时请根据电池盒里的正负极提示进行安装,以防电池装反。请使用18560 型号的可充电锂电池,其他类型电池不支持。
- 本产品需用户自行组装AI-Starter,组装时请注意不要用力拧紧螺丝,以防螺丝滑丝。
- AI-Starter使用过程中需要时刻监视其运行状态,防止AI-Starter卡死导致电机长时间 堵转,可能损坏AI-Starter主控板和电机。

# 1.2 售后条款

# 1.2.1 质保细则

- 本产品自用户收到商品之日起计算,机器本体质保期限6个月。
- 传感器电子类部件质保期限3个月。
- 电池不享受质保服务,只提供出厂损坏更换。
- 产品的包装、赠送物品、USB线、螺钉、扳手、结构件等易损件不享受质保服务。
   但若在购机后初次使用时有问题,确认非人为损坏时可联系更换。
- 签收货品时,请确认包装的完整性。在购买商品7日内(自用户收到商品之日起计算,含7日),若发现货品漏发或者由于运输原因造成的破损问题,以及产品或配件发生非人为性能故障,请联系当地售后进行补发或者更换,超出此时间范围视为无效。
- 在三包有效期内,符合换货条件的,销售者因无同型号同规格产品,顾客不愿调换 其他型号、规格产品而要求退货的,可以予以退货。
- 质保期间更换的产品配件,保修期限2个月,更换下来的不良品归越疆科技所有。

# 

符合以上情况的换货或者退货运费由越疆科技官方承担,如因各国政策原因需要 交纳目的地关税,由客户自行承担。

# 实行免费维修服务必须同时符合以下条件:

- 自购买产品在规定的产品保修期限内正常使用,并出现非人为的性能故障。
- 提供有效的购买证明、单据及单号。
- 产品需要增添包装、配件,或有翻新需求的,我司将收取一定的成本费用。

# 以下情况不属于免费产品维修服务的情形:

- 擅自改装、拆解等行为而造成的损坏。
- 使用环境不符合标准导致的损坏,如过载、高电压、高电流、高温等导致的损坏。
- 人为磕碰、跌落导致的外观以及功能异常。
- 人为置于潮湿环境、浸水、火烧等导致的功能异常。
- 未按官方说明书要求进行的不正确使用、安装及操作所造成的损坏。

文档版本 V1.1.0	(2019-01-07)
-------------	--------------

# 用户手册



- 未按官方说明书要求进行的电路改造、电池反装、过度充电、错误的安装、接线、 充电器的匹配使用不当导致的损坏。
- 与非本公司认证的第三方部件同时使用时发生可靠性及兼容性问题导致的损坏。

特别说明

- 本产品相关售后服务由当地代理商提供。
- 如果对相关售后服务有疑问,请及时联系官方售后,我们会尽快协助您解决问题。



# 2. 产品简介

# 2.1 产品特性

AI-Starter是一款针对教育、比赛推出的的智能小车,采用了免焊接式安装设计,以铜螺 柱结构连接主控板,以排针对插方式连接电路,接线清晰。其主控板是基于Arduino Mega2560 设计,兼容Arduino的开发板,非常容易上手。

功能特性:

- 智能避障
- 自动巡线行驶
- 根据颜色识别场景并执行不同任务
- 图形化编程,用户可通过积木搭建的方式进行编程来控制AI-Starter

# 2.2 零件清单

零件	数量
小车底盘	1
小车外壳	1
主控板	1
超声波传感器	3
红外巡线传感器模块	1
颜色传感器	2
直流减速电机 (含编码器)	2
万向轮	1
联轴器	2
轮胎	2
18650 锂电池	2
18650 电池盒	1
铜螺柱, M3*32+4	4
R2048 黑色尼龙铆钉	10
M3*5 圆头十字螺丝	30
M3*5 沉头十字螺丝	4
M3*6 圆头十字螺丝	6
M4*6 圆头十字螺丝	3

#### 表 2.1 零件清单

文档版本 V1.1.0 (2019-01-07)



零件	数量
4PIN 超声波连接线	3
6PIN 颜色传感器连接线	2
8PIN 巡线模块连接线	1
6PIN 电机连接线	2
USB连接线	1
载物托盘A	1
载物托盘B	1
十字螺丝刀	1
用户手册	1

# 2.3 技术规格

# 2.3.1 技术参数

# 表 2.2 参数说明

参数	说明
工作电压	7.4V
主控板	DuDuino Mega (兼容Arduino Mega 2560)
微处理器	ATmega2560
电池	18560可充电锂电池
超声波可测范围	3mm~500mm
组装后尺寸	195mm*172mm*79mm
整机重量	810g
最大载重	500g
车轮直径	67mm
运行环境	0° C~40° C
控制软件	Arduino IDE或Mixly
通信方式	USB通信、串口通信
扩展接口	2个4PIN通用I/O接口,UART扩展接口
传感器	• 超声波传感器*3
	• 颜色传感器*2
	• 红外巡线传感器模块*1
文档版本 V1.1.0(2019-01-07)	用户手册 版权所有 © 越疆科技有限公司



参数	说明
	• 地磁传感器*1
	• 光敏传感器*1
电机参数	• 减速比: 48: 1
	• 电机电压: 7V
	• 电机空载电流: 150mA
	• 电机堵转电流: 700mA
	• 最大转速: 100r/m
	• 编码器分辨率: 585pulse/r



# 3. **特性说明**

# 3.1 Al-Starter 控制器

### 3.1.1 概述

AI-Starter控制板是以AT Mega2560为核心,兼容Arduino的开发板,不仅集成了电机驱动、地磁感应器、光敏、按键、LED等模块,还集成了红外巡线信号接口、超声波模块信号接口、USB、Xbee、蓝牙、串口等接口。AI-Starter主控板预置了两个舵机信号接口,方便用户扩展。AI-Starter主控板如图 3.1所示。



图 3.1 Al-Starter 控制板示意图

# 3.1.2 AT Mega2560 处理器

AI-Starter处理器为atmega2560,兼容arduino2560,可以直接通过arduinoIDE进行开发,同时我们还提供了Mixly的图形化编程环境。

### 3.1.3 按键

AI-Starter控制器尾部集成了四位独立按键,如图 3.1所示,详细说明如表 3.1所示。

序号	说明
1	开关按键,开启或停止AI-Starter
	当停止AI-Starter时,需长按此按键约3秒

### 表 3.1 按键说明



序号	说明	
2	用户自定义	
	用户可在Mixly或Arduino IDE环境中设置其功能	
3	用户自定义	
	用户可在Mixly或Arduino IDE环境中设置其功能	
4	用户自定	
	用户可在Mixly或Arduino IDE环境中设置其功能	

### 3.1.4 **LED**

AI-Starter控制器尾部集成了四个LED灯,如图 3.1所示,详细说明如表 3.2所示。

序号	颜色	说明	
А	蓝色	用户自定义	
		用户可在Mixly或Arduino IDE环境中设置其功能	
В	红色	• 灭:表示AI-Starter处于未充电状态	
		• 红色常亮:表示AI-Starter处于充电状态	
С	红色	• 灭:表示AI-Starter电池电压正常	
		• 红色常亮:表示AI-Starter电池电压低	
D	蓝色	用户自定义	
		用户可在Mixly或Arduino IDE环境中设置其功能	

### 表 3.2 LED 指示灯说明

# 3.1.5 **USB**

AI-Starter控制板集成了USB下载功能,可通过USB连接电脑下载程序至AI-Starter,使 AI-Starter根据程序指令运行。同时USB接口还具备供电功能,当AI-Starter电池电压低时通过 USB连接电脑即可充电。

USB驱动可支持的操作系统:

- Win7
- Win8
- Win10

一般情况下,安装了Mixly或Arduino IDE后,系统会自动安装Arduino USB驱动。 将AI-Starter通过USB线连接计算机并开机后,可在设备管理器找到相应的COM口, 如图 3.2所示。如果未找到相应的COM口,需在"Arduino-X/drivers"目录下重新

文档版本 V1.1.0(2019-01-07)



安装Arduino USB驱动,其中X为Arduino版本号,请根据实际情况选择。

🎥 计算机管理	
文件(F) 操作(A) 查看(V) 帮	助(H)
🗢 🄿 🙇 🖬 🚺 🗾	
🎥 计算机管理(本地)	🗸 📲 Evan-PC
➤ 🕌 系统工具	> 🦏 IDE ATA/ATAPI 控制器
> 🕒 任务计划程序	> 🏣 SM Driver
> 🚼 事件查看器	> 🔲 处理器
> 🕺 共享文件夹	> 🔚 传感器
> Ň 性能	> 🔜 磁盘驱动器
昌 设备管理器	> 🍙 存储控制器
✓  < 存储	> 💼 打印队列
<b>唐</b> 磁盘管理	✓ 開端口(COM和LPT)
> 🛼 服务和应用程序	💭 USB Serial Device (COM6)
	> 📃 计算机
	> 🛄 监视器
	> 🔤 鍵盘
	> 😢 蓝牙
	> 🥅 人体学输入设备
	> 量 软件设备
	> 👖 声音、视频和游戏控制器
	> 🔰 鼠标和其他指针设备
	> 🛡 通用串行总线控制器
	> 1 图像设备
	> 🤤 网络造配器
	> 🌄 系统设备
	> 叫 首规输入和输出
1	1

图 3.2 USB 驱动

### 3.1.6 接口说明





### 图 3.3 AI-Starter 主控板接口示意图

表 3.3 接口说明

序号	说明
1	编码器接口,与AI-Starter底盘上的右侧电机 相连
2	电机接口,与AI-Starter底盘上的右侧电机相 连
3	电机接口,与AI-Starter底盘上的左侧电机相 连
4	编码器接口,与AI-Starter底盘上的左侧电机 相连
5	红外巡线传感器接口,与AI-Starter的红外巡 线传感器相连
6	颜色传感器接口,与AI-Starter底盘上右侧的 颜色传感器相连
7	左侧颜色传感器接口,与AI-Starter底盘上左侧的颜色传感器相连

文档版本 V1.1.0(2019-01-07)



序号	说明
8	超声波传感器接口,与AI-Starter右侧的超声 波传感器相连
9	超声波传感器接口,与AI-Starter前方的超声 波传感器相连
10	超声波传感器接口,与AI-Starter左侧的超声 波传感器相连
11	Xbee接口,采用UART接口
12	预留舵机接口
13	蓝牙接口,采用UART接口
14	预留舵机接口
15	预留串口接口,采用UART接口
16	Wifi接口,采用UART接口
17	USB接口,采用标准的Micro-USB接口
18	电源接口,与AI-Starter底盘上的电池相连

# 3.2 红外巡线传感器

AI-Starter小车底部内置一个红外巡线传感器,通过判断地面颜色来识别固定轨迹黑线,以实现自动巡线行驶。红外巡线传感器内置6路高精度红外对管,配合6个可调电位器,用来调节红外对管检测的距离的远近,可精确检测到黑线,检测距离为3cm,精度可达0.1cm。

# 3.3 超声波传感器

AI-Starter小车头部内置三个超声波传感器,可通过超声波反射测量前方3mm~500mm范围内障碍物的距离。

# 3.4 颜色传感器

AI-Starter小车底部内置两个颜色传感器,可通过颜色滤波器识别场地中的颜色,使 AI-Starter小车根据不同的颜色执行不同的任务。

用户手册



# 4. 安装指南

# 4.1 Mixly 软件安装

Mixly是一款基于Google的Blockly图形化编程框架开发的免费开源的Arduino图形化编程软件,AI-Starter在Mixly增加了API接口,用户可通过积木搭建的方式调用API接口编程并上传至AI-Starter,以实现对AI-Starter的控制。

Mixly软件无需复杂安装,下载安装包解压后即可直接使用。其下载路径: http://mixly.org/explore/software/mixly-arduino。

□□说明

目前AI-Starter仅支持Mixly0.995版本,其他版本暂不支持。

# 4.2 Arduino IDE

AI-Starter同时支持Arduino C语言编程,Arduino是一款便捷灵活、方便升级的开源电子 平台,包括Arduino开发工具Arduino IDE和核心库。用户可通过Arduino IDE调用API 接口进 行编程,并将程序上传到AI-Starter实现对AI-Starter的控制。

Arduino IDE软件包嵌套在Mixly软件包,解压Mixly安装包后即可直接使用。Arduino IDE 软件路径为"*安装路径*/Dobot\_Mixly/arduino-XXX。其中,XXX为Arduino配套的版本号,请根据实际情况替换。



# 5. **使用指南**

# 5.1 Mixly 使用说明

Mixly软件界面如图 5.1所示,使用Mixly时需选择Arduino\Genuino Mega or Mega 2560[atmega2560]开发板并选择对应的串口,如图 5.1红框所示。

用户可在Mixly界面左侧拖动所需的模块,待编译无误后上传至AI-Starter,使AI-Starter 按照编写的程序指令运行。

💯 Mixly 0.995									_		×
<b>護</b> 模块	代码	Copyright © ‡	师大教育学部创	客教育实验室	maker.bnu.edu	a.cn 🖀	通机器 高级机图	5	1	简体中	文 -
▶ 函数 ^	设置移动 方向	FRONT 速度					void setup()				
👕 DFRobot							}				
▼ Dobot	设置马达 端口	MOTORR · 速度 1					void loop()				
🎁 Basic							1				
👘 Config	获取马达位置	端口 MOTORR Y					1				>
🎁 Motion	初始化超声波	端口 SONAR1 7				-					
<b>1</b> /O						Ŷ					
👘 Attachment	【 获取超声波 端	SONAR1				$(\neq)$					
🌱 SmartBot	获取巡线端口	端口 IRI Y									
🎁 MakeBlock						<b></b>					
📬 CarControl	( 秋取巡线数值										
	获取地磁角度			_							
新建 打开 保存						Arduino/Genuino Mega or	▼ COM4 ▼	串口监视器	Ö,	_	—
						Arduino Yún		â			
						Arduino/Genuino Uno					
						Arduno Duemilanove or Diecimil	a[atmega328]	-			
						Arduino Duemianove or Diecimii	alatmega iooj				
						Arduino Nano[atmega326]					
						Arduino/Genuino Mega or Mega	2560[atmega2560]				
						Arduino/Genuino Mega or Mega	2560[atmega1280]				
						Arduino Mega ADK					
						Arduino Leonardo					

图 5.1 Mixly 软件界面



如果打开Mixly软件无法获取到AI-Starter对应的串口信息,请确保已安装Arduino USB驱动。如果安装后仍无法获取串口信息,请采用管理员身份打开Mixly。

Mixly使用说明在本手册中不做详细说明,请参见Mixly官网查阅相关手册。

# 5.2 Arduino IDE 使用说明

Arduino IDE界面如图 5.2所示。

用户手册



💿 sketch_dec14a   Arduino 1.8.2	_	×
文件 编辑 项目 工具 帮助		
		P
sketch_dec14a§		
<pre>#include <aistarter.h></aistarter.h></pre>		^
<pre>void setup() {     AIStarter_SmartBotInit(); } void loop() AIStarter_SmartBotSetMovment(FRONT, 1);</pre>		
}		
		~

图 5.2 Arduino IDE 界面

打开Arduino IDE软件后,需在"Tools > Borad"选择"Arduino/Genuino Mega or Mega 2560","Tools > Processor"选择"ATmega2560 (Mega 2560)",在"Tools > Port"选择相应的串口。

用户可在"File > Examples > AiStarter"选择样例,单击 为将样例上传至AI-Starter,使 AI-Starter按样例运行。也可参考章节7调用API编写程序后上传至AI-Starter,使AI-Starter按 自己编写的程序指令运行。



# 6. Al-Starter Demo

# 6.1 自动巡线 Demo

### 6.1.1 介绍

该Demo实现了AI-Starter在地图上自动巡线功能。

### 6.1.2 操作步骤

- 步骤1 开机后,电机停止。
- 步骤2 按下"启动|停止"按键,发出"滴一声"后开始巡线。
- 步骤3 再次按下"启动|停止"按键,发出"滴一声"后停止巡线。

### 6.1.3 **关键代码说明**

1) 获取当前巡线传感器值。

#### 程序 6.1 获取传感器数值

void getCurrentIRState(int \*irstate)

{

\*irstate = 0;
for (int i = 0; i < IR\_NUM; i++) {
 \*irstate |= AIStarter\_SmartBotGetIRModuleValue(i) << i;
}</pre>

2) 获取当前小车姿态。

### 程序 6.2 获取小车姿态

float getCurrentPos(const int irstate)

```
{
    const float coeff = 0.7;
    const int irPos[] = {-30, -18, -6, 6, 18, 30};//设置各个红外对管权重值
    static float lastPos;
    float curPos;
    float readPos;
    int total = 0;
    int irOffCnt = 0;
    //calculate the car position offset
    文档版本 V1.1.0 (2019-01-07) 用户手册 版权所有 © 越疆科技有限公司
```

# 🕖 DOBOT

```
for (int i = 0; i < IR_NUM; i++) {
     if (irstate & (1 << i)) {
          total += irPos[i];
          irOffCnt++;
     }
}
if (irOffCnt) {
     readPos = total / irOffCnt;
}
else {
     readPos = lastPos;
}
//calculate the current position
curPos = (1 - coeff) * lastPos + coeff * readPos;
lastPos = curPos;
return curPos;
```

```
3) 设置小车速度。
```

```
程序 6.3 设置小车速度
```

void setCarSpeed(const float curPos)

#### {

const int baseSpeed = 50; //rpm const float kp = 1;

const float ki = 0.06;

const float kd = 0.0;

const float errorsumLimit = 50;

```
float error = curPos;
```

static float lastError;

static float errorsum;

float errorChange;

int speedLeftWheel;

int speedRightWheel;

int speedOffset;

文档版本 V1.1.0 (2019-01-07)



```
//pid
errorsum += error;
if (errorsum> errorsumLimit) {
    errorsum = errorsumLimit;
}
else if (errorsum < -errorsumLimit){
    errorsum = -errorsumLimit;
}
errorChange = error - lastError;
speedOffset = kp * error + ki * errorsum + kd * errorChange;
lastError = error;
//calculate the wheel speed
speedLeftWheel = baseSpeed + speedOffset;
speedRightWheel = baseSpeed - speedOffset;
AIStarter_SmartBotSetMotor(MOTORL, speedLeftWheel);
AIStarter_SmartBotSetMotor(MOTORR, speedRightWheel);
Serial.println("go ahead");
```

```
}
```

# 6.2 自动避障 Demo

### 6.2.1 介绍

该Demo实现了AI-Starter自主避障功能。

# 6.2.2 操作步骤

- 步骤1 开机后,电机停止。
- 步骤2 按下【启动|停止】,开始后开始避障。
- **步骤3** 小车遇到障碍物后退一段距离,根据超声波传感器进行判断左转或右转,然后继续前进。
- 步骤4 再次按下【启动|停止】按键,发出"滴一声"响后停止巡线。

### 6.2.3 **关键代码说明**

1) 初始化 AI-Starter。

文档版本 V1.1.0(2019-01-07)	用户手册	版



#### 程序 6.4 初始化小车

AIStarter\_SmartBotInit();

2) 将 AI-Starter 超声波模块获取到的数据传入数组"dis"中。

### 程序 6.5 保存超声波数据

dis[0] = AIStarter\_SmartBotGetSonar(SONAR1);

dis[1] = AIStarter\_SmartBotGetSonar(SONAR2);

dis[2] = AIStarter\_SmartBotGetSonar(SONAR3);

3) 根据超声波模块检测到障碍物的距离设置转向模式。

#### 程序 6.6 设置转向模式

```
if(dis[0] > 2*DIS){
         motorStatus = AHEAD;
    else if(dis[0] > DIS \& dis[0] < 2*DIS ){
         motorStatus = motorStatus;
    else if(dis[0] > 0 & dis[0] < DIS )
         motorStatus = BACKRIGHT;
         break;
    }
    if(dis[1] > 2*DIS){
         motorStatus = AHEAD;
    else if(dis[1] > DIS \& dis[1] < 2*DIS ){
         motorStatus = motorStatus;
    else if(dis[1] > 0 & dis[1] < DIS )
         motorStatus = BACKRIGHT;
         break;
if(dis[2] > 2*DIS ){
         motorStatus = AHEAD;
    else if(dis[2] > DIS & dis[2] < 2*DIS ){
         motorStatus = motorStatus;
    else if(dis[2] > 0 & dis[2] < DIS )
         motorStatus = BACKLEFT;
         break;
```

文档版本 V1.1.0 (2019-01-07)

}



}

4) 根据超声波检测到障碍物的位置进行相应的转向, BACKRIGHT 为向右转,BACKLEFT 为向左转, AHEAD 为直行。

程序	6.7	根据障碍物位置进行转向
作主/丁	0.7	似加厚时加卫且处门积凹

SW	itch(motorStatus){
	case BACKRIGHT:
	AIStarter_SmartBotSetMotor(MOTORR,BACKSPEED);
	AIStarter_SmartBotSetMotor(MOTORL,BACKSPEED);
	delay(BACKTIME);
	AIStarter_SmartBotSetMotor(MOTORR,DIFSPEED);
	AIStarter_SmartBotSetMotor(MOTORL,FRONTSPEED);
	delay(SWERVETIME);
	break;
	case BACKLEFT:
	AIStarter_SmartBotSetMotor(MOTORR,BACKSPEED);
	AIStarter_SmartBotSetMotor(MOTORL,BACKSPEED);
	delay(BACKTIME);
	AIStarter_SmartBotSetMotor(MOTORR,FRONTSPEED);
	AIStarter_SmartBotSetMotor(MOTORL,DIFSPEED);
	delay(SWERVETIME);
	break;
	case AHEAD:
	AIStarter_SmartBotSetMotor(MOTORR,FRONTSPEED);
	AIStarter_SmartBotSetMotor(MOTORL,FRONTSPEED);
	break;
	default:
	break;

# 6.3 **白平衡校准 Demo**

### 6.3.1 介绍

}

AI-Starter颜色传感器校准白平衡。

文档版本 V1.1.0(2019-01-07)



### 6.3.2 操作步骤

- 步骤1 将AI-Starter放在A4白纸上。
- 步骤2 按下"启动|停止"开关,自动校准白平衡值。

#### 6.3.3 **关键代码说明**

1) 初始化AI-Starter。

程序 6.8 初始化 AI-Starter

AIStarter\_SmartBotInit();

1) 校准白平衡。

#### 程序 6.9 校准白平衡

AIStarter\_SmartBotSetColorWB(COLORSENOR1);

AIStarter\_SmartBotSetColorWB(COLORSENOR2);

### 6.4 颜色识别与自动巡线 Demo

#### 6.4.1 介绍

该Demo实现了AI-Starter颜色识别结合巡线使用的功能。

### 6.4.2 操作步骤

- 步骤1 按下"启动|停止"按键,发出"滴一声"响后进行巡线。
- 步骤2 巡线过程中,检测到黑色停止线后小车停止运动,并开始检测颜色。
- 步骤3 识别颜色为红色时,小车停留3秒,同时"滴滴滴"响三声。
- 步骤4 识别颜色为绿色时,小车停留3秒,蜂鸣器长鸣3秒。
- 步骤5 完成上述任务后,继续前进。
- 步骤6 再次按下"启动|停止"按键,发出"滴一声"响后停止巡线。

#### 6.4.3 **关键代码说明**

1) 初始化 AI-Starter。

程序 6.10 初始化小车

#### AIStarter\_SmartBotInit();

2) 检测颜色。

文档版本 V1.1.0(2019-01-07)



#### 程序 6.11 检测颜色

if(AIStarter\_SmartBotGetColorSenor(COLORSENOR1,RCOLOR)

-AIStarter\_SmartBotGetColorSenor(COLORSENOR1,GCOLOR) > 30 &&

```
AIStarter_SmartBotGetColorSenor(COLORSENOR1,RCOLOR)
```

-AIStarter\_SmartBotGetColorSenor(COLORSENOR1,BCOLOR) > 30) {

colorState = RLINE;

3) 根据颜色状态进行相应动作。

程序	6.12	根据颜色执行动作
	-	

switch(colorState) { case OTHERLINE: //colorRec = false; lineState = LINEPATROL; break; case RLINE: delay(3000); //colorRec = false; lineState = LINEPATROL; break; case GLINE: delay(3000); //colorRec = false; lineState = LINEPATROL; break; default: break; }

# 6.5 机械臂协作 Demo

### 6.5.1 介绍

该Demo实现AI-Starter与Magician协作。

- 1) AI-Starter到达绿线停止一分钟。
- 2) 机械臂通过Pixy摄像头检测到小车到达搬运点后开始执行搬运动作。
- 3) 机械臂从方块区吸取方块放置到小车上。
- 4) 一分钟后AI-Starter装载完毕重新启动巡线。

文档版本 V1.1.0 (2019-01-07)



#### 6.5.2 操作步骤

#### **AI-Starter**

- 步骤1 按下"启动|停止"按键,发出"滴一声"响后进行巡线。
- 步骤2 巡线过程中,检测到黑色停止线后小车停止运动,并开始检测颜色。
- 步骤3 识别颜色为红色时,小车停留3秒,同时"滴滴滴"响三声。
- 步骤4 识别颜色为绿色时,发出"滴一声",小车停留1分钟。
- 步骤5 完成上述任务后,继续前进。
- 步骤6 再次按下"启动|停止"按键,发出"滴一声"响后停止巡线。

#### Magician

- **步骤1** 打开Magician Cooperation文件夹下的Magician文件夹,上传Magician固件至 Arduino拓展板。
- 步骤2 当机械臂检测到AI-Starter停至装载区时,开始执行装载方块任务。
- 步骤3 AI-Starter继续巡线。

### 6.5.3 关键代码说明

### **AI-Starter**

1) 初始化 AI-Starter。

#### 程序 6.13 初始化小车

#### AIStarter\_SmartBotInit();

2) 检测颜色。

#### 程序 6.14 检测颜色

if(AIStarter\_SmartBotGetColorSenor(COLORSENOR1,RCOLOR)

-AIStarter\_SmartBotGetColorSenor(COLORSENOR1,GCOLOR) > 30 &&

AIStarter\_SmartBotGetColorSenor(COLORSENOR1,RCOLOR)

-AIStarter\_SmartBotGetColorSenor(COLORSENOR1,BCOLOR) > 30) {

colorState = RLINE;

}

3) 根据颜色状态进行相应动作。

#### 程序 6.15 根据颜色执行动作

文档版本 V1.1.0(2019-01-07)	用户手册	版权所有 © 越疆科技有限公司
case OTHERLINE:		
<pre>switch(colorState) {</pre>		



//colorRec = false; lineState = LINEPATROL; break; case RLINE: delay(3000); //colorRec = false; lineState = LINEPATROL; break; case GLINE: delay(3000); //colorRec = false; lineState = LINEPATROL; break; default: break;

#### Magician

1) 设置方块点位。

```
程序 6.16 设置方块点位
```

```
float AreaPoint[4][3] = {
{137.05, -206.94, -39},
{137.05, -244.31, -39},
{100.50, -206.94, -39},
{100.50, -244.31, -39}
```

};

2) 设置AI-Starter装载点位。

```
程序 6.17 设置装载点位
```

```
float trayPoint[4][3] = {
    {308.12, 25.92, 28},
    {308.12, -15.92, 28},
    {258.12, 25.92, 28},
    {258.12, -15.92, 28}
};
```

3) 吸取方块,放置AI-Starter装载点位。

```
文档版本 V1.1.0(2019-01-07)
```



#### 程序 6.18 吸取方块

void AreaToAIStarter()
{
for(uint8_t i=0; i<4; i++){
Dobot_SetPTPCmdEx(JUMP_XYZ, AreaPoint[i][0], AreaPoint[i][1], AreaPoint[i][2], 0);
Dobot_SetEndEffectorSuctionCupEx(true);
Dobot_SetPTPCmdEx(MOVL_XYZ, AreaPoint[i][0], AreaPoint[i][1], AreaPoint[i][2]+70, 0);
Dobot_SetPTPCmdEx(JUMP_XYZ, trayPoint[i][0], trayPoint[i][1], trayPoint[i][2], 0);
Dobot_SetEndEffectorSuctionCupEx(false);
Dobot_SetPTPCmdEx(MOVL_XYZ, trayPoint[i][0], trayPoint[i][1], trayPoint[i][2]+30, 0);
}
Dobot_SetPTPCmdEx(MOVJ_XYZ, InitPositionX, InitPositionY, InitPositionZ, InitPositionR);
}

4) 初始化摄像头和机械臂。

#### 程序 6.19 初始化摄像头和机械臂

#### pixy.init();

Dobot\_Init();

5) 检测物块数量。

### 程序 6.20 检测物块数量

pixy.ccc.getBlocks();

# 6.6 舵机驱动 Demo

### 6.6.1 介绍

AI-Starter 舵机驱动示例。

### 6.6.2 操作步骤

**步骤1** 将舵机接入AI-Starter数字 I/O 7。I/O 7说明请参照章节3.1.1概述。 **步骤2** 按下"启动|停止"开关, 舵机自动往复旋转180°。

### 6.6.3 **关键代码说明**

1) 初始化AI-Starter。

### 程序 6.21 初始化 AI-Starter

立档版本 V1 1.0(2019-01-07)	田户手冊	版权所有 ◎ 越疆科技有限公司
入间版本 V1.1.0(2019-01-07)		版状/// 书 ⑤ 透麵件1文件版公书



AIStarter\_SmartBotServoAttach(SERVO1);

2) 舵机做 180° 往复运动。

### 程序 6.22 舵机往复运动

```
for (pmwServo = 0; pmwServo <= 180; pmwServo += 1) {
```

```
AIStarter_Servo.write(pmwServo);
```

delay(15);

}

```
for (pmwServo = 180; pmwServo >= 0; pmwServo -= 1) {
```

AIStarter\_Servo.write(pmwServo);

delay(15);

# 6.7 Xbee 通信 Demo

### 6.7.1 介绍

AI-Starter与PC端Xbee通信示例。

### 6.7.2 操作步骤

- **步骤1** 在XCTU官网https://www.digi.com/support/productdetail?pid=3352&type=utilities 下载XCTU软件。
- 步骤2 使用XCTU软件烧录XbeeMatch.xpro固件。
- 步骤3 烧录完成后,将Xbee装回AI-Starter串口2上,串口2请参考章节3.1.1概述。
- 步骤4 按下"启动|停止"开关,使用XCTU给AI-Starter发送控制命令。
- 步骤5 根据PC端发送的命令,AI-Starter执行相应的动作。

# 6.7.3 关键代码说明

1) 初始化AI-Starter。

### 程序 6.23 初始化 AI-Starter

AIStarter\_SmartBotInit();

2) 读取 Xbee 信息。

### 程序 6.24 读取 Xbee 信息

```
strCommand = AIStarter_SmartBotXbeeRead();
```

```
文档版本 V1.1.0(2019-01-07)
```



```
3) 根据Xbee信息执行相应动作
```

程序 6.25 根据 Xbee 信息执行相应动作

```
if(!AIStarter_SmartBotXbeeCompare(strCommand, "Ahead")) {
    rOffSet = 1;
    lOffSet = 1;
} else if(!AIStarter_SmartBotXbeeCompare(strCommand, "Back")) {
    rOffSet = -1;
    lOffSet = -1;
} else if(!AIStarter_SmartBotXbeeCompare(strCommand, "TurnLeft")) {
    rOffSet = 1;
    lOffSet = 0.5;
} else if(!AIStarter_SmartBotXbeeCompare(strCommand, "TurnRight")) {
    rOffSet = 0.5;
    lOffSet = 1;
} else if(!AIStarter_SmartBotXbeeCompare(strCommand, "Stop")) {
    rOffSet = 0;
    lOffSet = 0;
}
```

```
4) 清除Xbee信息
```

```
程序 6.26 清除 Xbee 信息
```

AIStarter\_SmartBotXbeeClear();



# 7. API 接口

# 7.1 初始化

表 7.1 初始化

原型	int AIStarter_SmartBotInit ()		
描述	初始化		
参数	无		
返回	无		

# 7.2 设置小车速度

### 表 7.2 设置小车速度

原型	int AIStarter_SmartBotSetMovment (int dir, int speed)	
描述	设置小车速度和方向	
参数	dir: 设置小车方向	
	enum{	
	FRONT,	
	BACK,	
	RIGHT,	
	LEFT	
	};	
	speed: 设置占空比, 范围: 0~255	
返回	无	

# 7.3 设置小车运动方向/速度/时间

|--|

原型	int AIStarter_SmartBotSetMovmentTime (int dir, int speed,float time)	
描述	设置小车运动方向、速度和时间	
参数	dir: 设置小车方向	
	enum {	
	FRONT,	
	BACK,	
	RIGHT,	

文档版本 V1.1.0(2019-01-07)



	LEFT	
	};	
	speed: 设置占空比,范围: 0~255	
	time: 设置小车运动时间(单位: s)	
返回	无	

# 7.4 设置电机转速

### 表 7.4 设置电机转速

原型	int AIStarter_SmartBotSetMotor(int port, int speed)	
描述	设置电机转速	
参数	port: 选择电机	
	enum {	
	MOTORR,	
	MOTORL	
	};	
	speed: 设置电机转速。范围: 0~200rpm	
返回	无	

# 7.5 **设置电机参数**

### 表 7.5 设置电机参数

原型	int AIStarter_SmartBotSetMotorPI (float KP, float KI)	
描述	设置电机参数	
参数	KP: 比例因子。取值范围: 0.5~2.5	
	KI: 积分因子。取值范围: 0.05~0.5	
返回	无	

# 7.6 获取电机位姿

### 表 7.6 获取电机位姿

原型	float AIStarter_SmartBotGetMotorPose (int port)		
描述	获取电机位姿		
参数	port:选择电机		
	enum {		
	MOTORR,		
文档版本	V1.1.0 (2019-01-07)	用户手册	版权所有 © 越疆科技有限公司



	MOTORL
	};
返回	电机编码器读数

# 7.7 初始化超声波传感器

### 表 7.7 初始化超声波传感器

原型	int AIStarter_SmartBotSetSonar (int port)	
描述	初始化超声波传感器	
参数	port: 选择超声波	
	enum {	
	SONAR1,	
	SONAR2,	
	SONAR3	
	};	
返回	无	

# 7.8 获取超声波探测距离

### 表 7.8 获取超声波探测距离

原型	float AIStarter_SmartBotGetSonar (int port)	
描述	获取超声波传感器探测的距离	
参数	port: 选择超声波传感器	
	enum {	
	SONAR1,	
	SONAR2,	
	SONAR3	
	};	
返回	超声波探测到障碍物距小车的距离(单位: cm)	

# 7.9 探测障碍物

# 表 7.9 探测障碍物

原型	bool AIStarter_SmartBotGetBarrier (int port)		
描述	探测前方是否有障碍物		
文档版本	V1.1.0 (2019-01-07)	用户手册	版权所有 © 越疆科技有限公司



参数	port: 选择超声波传感器			
	enum {			
	SONAR1,			
	SONAR2,			
	SONAR3			
	};			
返回	1: 检测到障碍物; 0: 未检测到障碍物			

# 7.10 获取巡线数据

原型	int AIStarter_SmartBotGetIRModuleValue (int port)			
描述	获取巡线数据			
参数	port: 选择巡线传感器端口			
	enum {			
	IR1,			
	IR2,			
	IR3,			
	IR4,			
	IR5,			
	IR6			
	};			
返回	1: 检测到黑线; 0: 未检测到黑线			

### 表 7.10 获取巡线数据

# 7.11 获取地磁角度

#### 表 7.11 获取地磁角度

原型	float AIStarter_SmartBotGetCompass ()	
描述	获取地磁角度	
参数	无	
返回	获取地磁角度	

# 7.12 地磁校准

表 7.12 地磁校准

原型	void AIStarter_SmartBo	otSetCompassCalibration()	
文档版本	V1.1.0 (2019-01-07)	用户手册	版权所有 © 越疆科技有限公司



描述	校准方法:开机后按下最左侧按钮开始校准,校准过程中使小车在空间内分别绕XYZ三 轴方向旋转360°以上,旋转完成后按下最左侧按钮结束校准
参数	无
返回	地磁校准

# 7.13 设置颜色白平衡

#### 表 7.13 设置颜色白平衡

原型	int AIStarter_SmartBotSetColorWB (int port)		
描述	设置颜色白平衡		
参数	port: 选择颜色传感器		
	enum{		
	COLORSENOR1,		
	COLORSENOR2		
	};		
返回	无		

# 7.14 启停颜色传感器

### 表 7.14 启停颜色传感器

原型	int AIStarter_SmartBotSetColorSenor (int port,bool ison)		
描述	控制颜色传感器的启停		
参数	port: 选择颜色传感器		
	enum{		
	COLORSENOR1,		
	COLORSENOR2		
	};		
	ison: true 开启; false 关闭		
返回	无		

# 7.15 检测颜色

### 表 7.15 检测颜色

原型	bool AIStarter_SmartBotDetColorSenor (int port,int color)	
描述	检测所选的对应颜色即红绿蓝是否存在	

文档版本 V1.1.0(2019-01-07)



参数	port: 选择颜色传感器		
	enum{		
	COLORSENOR1,		
	COLORSENOR2		
	};		
	color: 选择所检测的颜色		
	enum{		
	RCOLOR,		
	GCOLOR,		
	BCOLOR,		
	};		
返回	1: 检测到颜色; 0: 未检测到颜色		

# 7.16 获取 RGB 色值

表	7.16	获取 RGB 色值	
x	1.10	ふ 坂 い し し 但	

原型	int AIStarter_SmartBotGetColorSenor (int port, int color)		
描述	获取RGB色值		
参数	port: 选择颜色传感器		
	enum{		
	COLORSENOR1,		
	COLORSENOR2		
};			
	color: 选择颜色滤波器		
	enum{		
	RCOLOR,		
	GCOLOR,		
	BCOLOR,		
	};		
返回	返回RGB色值,范围: 0~255		

# 7.17 初始化按键

表 7.17 初始化按钮

原型	int MobilePlatform_SmartBotSetKeyInit()		
功能	初始化按钮		
文档版	在V1.1.0(2019-01-07)	用户手册	版权所有 © 越疆科技有限公司



参数	无
返回	无

# 7.18 获取按键状态

表 7.18 设置按键

原型	int AIStarter_SmartBotGetKeyValue (int key)
描述	设置按键
参数	key: 选择按键
	enum{
	SW1,
	SW2,
	SW3
	};
返回	1: 按键按下; 0: 按键释放

# 7.19 设置灯光状态

### 表 7.19 设置 LED 灯

原型	int MobilePlatform_SmartBotSetLED(int port, int state)
描述	设置LED灯
参数	port:选择 LED 灯。
	enum{
	LED1,
	LED2
	}:
	state: 设置状态。
	enum{
	ON,
	OFF,
	BLINK
	}:
返回	无



# 7.20 获取光敏数值

表 7.20 获取光敏传感器数值

原型	int AIStarter_SmartBotGetLightAnalog ()
描述	获取光敏传感器数值
参数	无
返回	光敏传感器数值

# 7.21 设置超声波传感器检测距离

表 7.21 设置超声波传感器检测距
--------------------

原型	int AIStarter_SmartBotSetSonarThreshold (int dis)
描述	设置超声波传感器检测距离
参数	dis: 设置检测距离。(单位: cm)
返回	无

# 7.22 连接舵机

#### 表 7.22 连接舵机

原型	int AIStarter_SmartBotServoAttach(int servo)
描述	连接舵机
参数	servo:选择舵机
	enum{
	SERVO1,
	SERVO2
	};
返回	无

# 7.23 设置舵机角度

表 7.23 设置舵机角度

原型	int AIStarter_SmartBotServoWrite(int servo, int value)
描述	设置舵机角度

文档版本 V1.1.0(2019-01-07)



参数	servo: 选择舵机
	enum{
	SERVO1,
	SERVO2
	};
	value: 设置舵机角度, 取值范围: 0°~180°
返回	无

# 7.24 断开舵机连接

### 表 7.24 断开舵机连接

原型	int AIStarter_SmartBotServoDetach(int servo)
描述	断开舵机连接
参数	选择舵机:
	enum{
	SERVO1,
	SERVO2
	}:
返回	无

# 7.25 开启后台定时器任务

### 表 7.25 开启后台定时器任务

原型	int AIStarter_SmartBotTimerTaskAttach()
描述	开启定时器任务
参数	无
返回	无

# 7.26 关闭后台定时器任务

### 表 7.26 关闭后台定时器任务

原型	int AIStarter_SmartBotTimerTaskDetach()
描述	关闭定时器任务
参数	无
返回	无

文档版本 V1.1.0(2019-01-07)



# 7.27 Xbee 数据读取

原型	String& AIStarter_SmartBotXbeeRead()
描述	读取Xbee数据
参数	无
返回	无

表 7.27 Xbee 数据读取

# 7.28 Xbee 数据写入

### 表 7.28 Xbee 数据写入

原型	String& AIStarter_SmartBotXbeeWrite()
描述	Xbee数据写入
参数	无
返回	无

# 7.29 Xbee 数据对比

### 表 7.29 Xbee 数据对比

原型	int AIStarter_SmartBotXbeeCompare(const String &str1, const String &str2)
描述	对比两个字符串
参数	无
返回	返回0表示字符串相同
	返回非0表示字符串不相同

# 7.30 Xbee 数据清除

表 7.30 Xbee 数据清除

原型	<pre>int AIStarter_SmartBotXbeeClear()</pre>
描述	清除Xbee数据
参数	无
返回	无

文档版本 V1.1.0(2019-01-07)