

URG04 使用说明

林靖宇 linjy02@hotmail.com 2019 年 9 月 27 日

1. 概述

本模块用于控制二维激光扫描仪（URG-04LX-UG01）和读取扫描数据。

模块分为 C++版（URG04Kit_x64.dll）和 C#版（URG04Control.dll，需要 C++版模块 URG04Kit_x64.dll）。

2. C++编程

用 C++编程时注意设置项目属性，包括引用包含文件 URG04Kit.hpp 和库文件 URG04Kit_x64.dll，设置系统路径指向 dll 文件。

首先应用程序可以调用静态函数 SearchURG04Device()搜索安装激光扫描仪的端口。拔插设备后需要再次调用此函数更新端口列表。调用此函数后就可以调用静态函数 GetURG04Total()和 GetURG04Port()获取设备总数、端口号和设备名称。如果知道扫描仪所在端口，可省去此步骤。

然后步骤如下：

- ①创建一个 URG04Device 对象。
- ②调用 StartScan()选择设备端口并启动激光扫描仪。调用 Close()关闭启动的激光扫描仪。URG04Device 对象结束时自动关闭启动的激光扫描仪。
- ③调用 ReadScanAngle()读取测距数据对应的扫描角数组。扫描角数组是不变的。
- ④定时（周期 100ms）调用 ReadRange()读取测距数据及其时戳。
- ⑤如果需要，可以调用 SetDataLog()启动记录测距数据，log 文件为 urgRange.txt；可以调用静态函数 RangeToMap()将测距数据转换为二维地图。

URG04Device 的成员函数说明见表 1。

表 1 URG04Device 的 C++函数说明

函数类别	函数原型	说明
设备查询函数	static int SearchURG04Device()	功能：查询连接的激光扫描仪。 返回值：系统中激光扫描仪的数量。 说明：拔插设备后需要再次调用此函数。有可能误将其它 USB 设备识别为激光扫描仪。

	static char* GetURG04Port(int idx, int& ComNo)	功能：读取激光扫描仪的名称和串口号。调用 SearchURG04Device()后有效。 输入：idx - 激光扫描仪的序号。 输出：ComNo - 串口端口号（即 COMx 中的 x）。无效串口的端口号为-1。 返回值：名称字符串指针。NULL 表示序号超过设备总数。
	static int GetURG04Total()	功能：获得系统中激光扫描仪的总数。如果未调用过 SearchURG04Device()则返回-1。无激光扫描仪则返回 0。
设备管理函数	int StartScan(int ComNo)	功能：启动指定串口上的 URG 激光扫描仪。成功后持续采集测距数据，每组数据采集时间为 100ms。用 ReadRange()可以随时读取测距数据。 输入：ComNo - 激光扫描仪的串口号。 输出：无。 返回值：1 表示成功启动激光扫描仪。0 表示非激光扫描仪。
	void Close()	功能：关闭 URG 激光扫描仪，停止采集测距数据。 说明：URG04Device 对象结束时自动调用此函数。
数据获取函数	int ReadScanAngle(double* theta);	功能：读取测距数据对应的扫描角（rad）及其数量 szTheta。如果 theta 为 NULL 则仅返回扫描角的数量 szTheta。 输入：theta[] - 扫描角缓冲区，大小为 szTheta。 输出：theta[] - 扫描角数组。 返回值：扫描角的数量 szTheta（即测距数据的数量），等效于扫描的范围。 说明：扫描仪将一周（360°）分为 1024 步，因此扫描角的单位为 $360^{\circ}/1024=0.3516^{\circ}$ 。扫描仪左右对称扫描。定义扫描仪正前方的扫描角为 0，逆时针方向为正向。因此 szTheta=0 是最右边，szTheta/2-1 是正前方，szTheta-1 是最左边。
	int ReadRange(double* range, int* timeStamp, double* range0 = NULL, int* timeStamp0 = NULL)	功能：读取测距数据（m）及其时戳和数量 szRange。如果缓冲区为 NULL 则不返回相应数据。 输入：range[], range0[] - 测距数据缓冲区，大小为 szRange。timeStamp, timeStamp0 - 用于获取时戳。 输出：range[], range0[] - 最新与上次的测距数据，每个扫描角对应一个距离值。*timeStamp, *timeStamp0 - 测距数据的时戳。 返回值：测距数据的数量 szRange，等于扫描角的数量。 说明：扫描仪将一周（360°）分为 1024 步，因此扫描角的单位为 $360^{\circ}/1024=0.3516^{\circ}$ 。扫描仪左右对称扫描。定义正前方的扫描角为 0，逆时针方向为正向。因此 szRange=0 是最右边，szRange/2 是正前方，szRange-1 是最左边。
设备信息	int IsActive()	功能：判断设备是否启动，即关联的串口对象是否打开。 返回值：返回 1 表示已启动，0 表示已关闭。

函数	int GetActiveURGComNo()	功能：获得启动的设备的串口端口号。返回-1 表示未启动。
	void SetDataLog(int bLog); int GetDataLog();	功能：访问数据记录状态。log 文件为 urgRange.txt。
辅助函数	static void RangeToMap(unsigned char *map, int mapRow, int mapCol, double r_front, double steering, double* range, int szRange = -1, double stepAngle = -1)	功能：将测距数据转换为二维地图。 输入：map[mapRow][mapCol] - 二维地图缓冲区。r_front - 地图中心到地图顶部正中的物理距离（m），用于尺度变换。steering - 转向角（弧度），地图正上方为 0（北），逆时针为正，顺时针为负。range[szRange] - 用 ReadRange() 读取的测距数据或经过处理的测距数据（m）。szRange=-1 表示用系统缺省值。stepAngle - 扫描角步长（弧度）。stepAngle=-1 表示用系统缺省值。 输出：map - 二维地图，像素逐行排列，每个像素取值 0 表示阻塞，0xFF 表示无遮挡。 说明：扫描仪位于地图中心，r_front 对应半个地图（mapRow/2）的距离。调用 ReadScanAngle() 可以获得 szRange 值，并且相邻的扫描角之差就是 stepAngle。

3. C#编程

用 C# 编程时需要引用 URG04Control.dll，命名空间为 URG04Driver，并需要 URG04Kit_x64.dll。

URG04Control.dll 包含以下两个类，均可独立使用：

- 1) URG04SetupControl: URG04 激光扫描仪管理控件。
- 2) URG04Device: 实现 URG04 激光扫描仪的基本访问功能。用户需要设计控制界面。

3.1. URG04SetupControl

URG04SetupControl 提供控制界面和数据访问功能，控制界面见图 1。

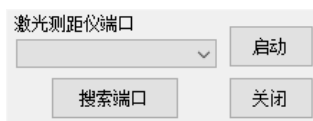


图 1 URG04SetupControl 界面

URG04SetupControl 使用方法如下：

- ①在窗体中添加一个 URG04SetupControl 控件（例如 urgSetup1）。
- ②调用 urgSetup1.InitializeControl() 初始化控件。
- ③用户在“设备端口”列表中选择设备，点击“启动”使设备开始工作，点击“关闭”停止设备。启动的设备才能输出测距数据。如果插入了新设备，点击“搜索端口”刷新设备列表。如果需要，选中“记录数据”启动记录测距数据，log 文件为 urgRange.txt。
- ④读取数据：
 - 数组 urgSetup1.theta 存储测距数据对应的扫描角。

- 定时（周期 100ms）调用 `urgSetup1.urg04Device.ReadRange()` 读取测距数据及时戳（参阅 `URG04Device`）。
 - 如果在主窗体中添加一个 `Timer` 控件（例如 `timer1`）定时读取测距数据，则应设置 `urgSetup1.data_timer = timer1`，这样点击“启动”和“关闭”也同时启动和停止定时器 `timer1`。
- ⑤ 如果需要，调用 `URG04Device.RangeToMap()` 将测距数据转换为二维地图，再调用 `URG04Device.MapToBmp()` 转换为灰度图以便显示（参阅 `URG04Device`）。
- ⑥ 可以访问以下属性：`urg04Device` 是设备对象（参阅 `URG04Device`），`theta` 是扫描角数组，`nRange` 是扫描角数组长度，即一组测距数据长度，`urg04_port` 是安装激光扫描仪的端口号列表。

3.2. URG04Device

`URG04Device` 不包含用户界面，提供设备控制和数据访问的全部功能。

首先应用程序可以调用静态函数 `URG04Device.SearchURG04Device()` 搜索安装激光扫描仪的端口。拔插设备后需要再次调用此函数更新端口列表。调用此函数后就可以调用静态函数 `URG04Device.GetURG04Total()` 和 `URG04Device.GetURG04Port()` 获取设备总数、端口号和设备名称。如果知道扫描仪所在端口，可省去此步骤。

然后步骤如下：

- ① 创建一个 `URG04Device` 对象。
- ② 用 `StartScan()` 选择设备端口并启动激光扫描仪，用 `Close()` 关闭激光扫描仪（对象结束时自动关闭）。属性 `ScannerActive` 判断设备是否启动。属性 `ActiveURGPort` 是当前启动的设备的端口号，属性 `URGPort` 是当前启动或刚关闭的设备的端口号。
- ③ 用 `ReadScanAngle()/ReadScanAngleDeg()` 获取扫描角数组以及数组大小。扫描角数组是不变的。
- ④ 定时（周期 100ms）调用 `ReadRange()` 读取测距数据及其时戳。数据未进行处理。
- ⑤ 如果需要，设置属性 `Log` 为 `true` 记录数据，`log` 文件为 `urgRange.txt`；用 `RangeToMap()` 将测距数据转换为二维地图，再用 `MapToBmp()` 转换为灰度图以便显示。

`URG04Device` 的成员函数说明见表 2。

表 2 `URG04Device` 的 C# 函数说明

类别	方法 / 属性	说明
设备查询	<code>static int SearchURG04Device()</code>	功能：查询连接的激光扫描仪。 返回值：系统中激光扫描仪的数量。 说明：拔插设备后需要再次调用此函数。有可能误将其它 USB 设备识别为激光扫描仪。

	static string GetURG04Port(int idx, out int ComNo)	功能：读取激光扫描仪的名称和串口号。调用 SearchURG04Device()后有效。 输入：idx - 激光扫描仪的序号。 输出：ComNo - 串口端口号（即 COMx 中的 x）。无效串口的 端口号为-1。 返回值：名称字符串指针。NULL 表示序号超过设备总数。
	static int GetURG04Total()	功能：获得系统中激光扫描仪的总数。如果未调用过 SearchURG04Device()则返回-1。无激光扫描仪则返回 0。
设备 管理	int StartScan(int ComNo)	功能：启动指定串口上的 URG 激光扫描仪。成功后持续采 集测距数据，每组数据采集时间为 100ms。用 ReadRange() 可以随时读取测距数据。 输入：ComNo - 激光扫描仪的串口号。 输出：无。 返回值：1 表示成功启动激光扫描仪。0 表示非激光扫描 仪。
	void Close()	功能：关闭 URG 激光扫描仪，停止采集测距数据。 说明：URG04Device 对象结束时自动调用此函数。
数据 获取	int ReadScanAngle (double[] theta) int ReadScanAngleDeg (double[] thetaDeg)	功能：读取测距数据对应的扫描角（rad / deg）及其数量 szTheta。如果 theta 为 NULL 则仅返回扫描角的数量 szTheta。 输出：theta - 扫描角数组。 返回值：扫描角的数量（即测距数据的数量），等效于扫 描的范围。 说明：扫描仪将一周（360°）分为 1024 步，因此扫描角 的单位为 360° /1024=0.3516°。扫描仪左右对称扫描。 定义扫描仪正前方的扫描角为 0，逆时针方向为正向。因 此 szTheta=0 是最右边，szTheta/2-1 是正前方，szTheta-1 是最左边。
	int ReadRange(double[] range, ref int timeStamp, double[] range0, ref int timeStamp0)	功能：读取测距数据（m）及其时戳和数量 szRange。如果 缓冲区为 NULL 则不返回相应数据。 输出：range, range0 - 最新与上次的测距数据，每个扫描 角对应一个距离值。timeStamp, timeStamp0 - 测距数据的 时戳。 返回值：测距数据的数量，等于扫描角的数量。 说明：扫描仪将一周（360°）分为 1024 步，因此扫描角 的单位为 360° /1024=0.3516°。扫描仪左右对称扫描。 定义正前方的扫描角为 0，逆时针方向为正向。因此 szRange=0 是最右边，szRange/2 是正前方，szRange-1 是最 左边。
设备 信息	bool ScannerActive	功能：判断设备是否启动，即关联的串口对象是否打开。
	int ActiveURGPort	功能：启动的设备的端口号。-1 表示设备关闭。
	int URGPort	功能：当前或之前启动的设备的端口号。-1 表示未曾启动。
	bool Log	功能：访问数据记录状态。log 文件为 urgRange.txt。

辅助函数	static void RangeToMap (byte[,] map, double r_front, double steering, double[] range, double stepAngle)	<p>功能：将测距数据转换为二维地图。</p> <p>输入：map - 空二维地图数组，大小为 width*height。 r_front - 地图中心到地图顶部正中的物理距离（m），用于尺度变换。steering - 转向角（弧度），地图正上方为 0（北），逆时针为正，顺时针为负。range - 用 ReadRange() 读取的测距数据（m）。stepAngle - 扫描角步长（弧度），缺省值为 $2 \times 3.1415927 / 1024$。</p> <p>输出：map - 填充的二维地图，像素逐行排列，每个像素取值 0 表示阻塞，0xFF 表示无遮挡。。</p> <p>说明：扫描仪位于地图中心，r_front 对应半个地图（mapRow/2）的距离。调用 ReadScanAngle() 可以获得 szRange 值，并且相邻的扫描角之差就是 stepAngle。</p>
	static Bitmap MapToBmp (byte[,] map)	<p>功能：将二维地图转换为 BMP 图像。</p> <p>输入：map - 填充的二维地图，像素逐行排列，每个像素取值 0 表示阻塞，0xFF 表示无遮挡。</p> <p>返回值：二维地图灰度图像。</p>
	static double RoundedAngle (double theta)	功能：角度规整为 $(-\pi \pi)$ 。
	static double RoundedAngleDeg (double theta)	功能：角度规整为 $(-180.0 \ 180.0]$