7bot 机械臂固件函数库

中文手册

Arm2.0 版本系列

张文超

Arm2.0 版机械臂固件函数库简介

介绍

该固件函数库是一个函数包，它由宏，类，以及外部函数共同构成。该库中包含了每一个形参的讲解和一个应用示例。通过使用本固件函数库，用户无需掌握细节，也可以轻松对机械臂进行操作，因此，该固件库可以大大减少用户的开发时间，从而降低开发成本。另外，对于学习需要的客户，使用该固件库，可以大大缩减入门时间，降低编程难度。

此库函数手册，一般分为三部分；

- 定义
- 函数库概述
- 函数库的应用描述

目录

第一章 机械臂基本函数(Arm.h)

机械臂基本函数包括了机械臂运动的基本函数，我们一般完成机械臂的运动都是利用该机械臂函数库中的函数完成。

注意 1：为了机械臂方便管理和编程，舵机统一从下而上从 0 开始命名舵机的 ID 号。具体请参照图 1。

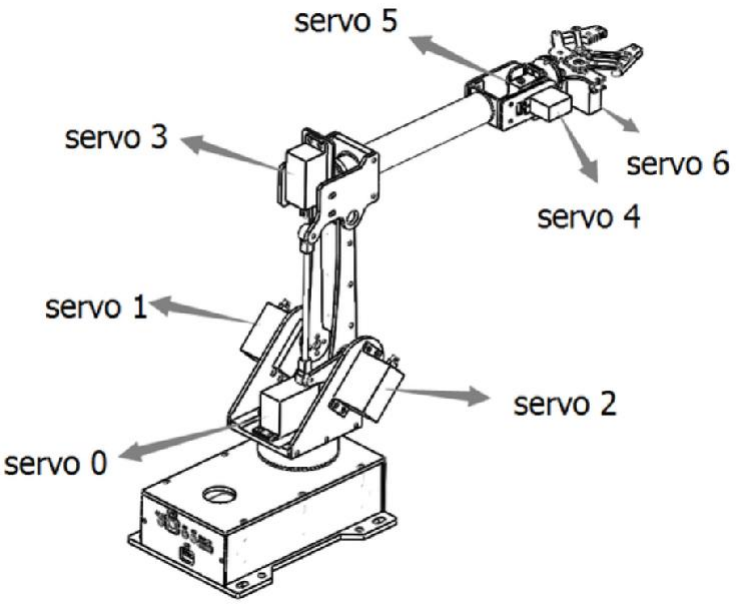

图 1

注意 2：如上图图 1 所示，在舵机的长方形底座上，我们约定，底座长方形的长边方向为 y 轴方向，短边方向为 x 轴方向，垂直于 xy 轴方向为 z 轴方向。

注意 3：在 Arm.cpp 这个源文件中我们已经定义了一个机械臂对象，因为我们的源文件中一些函数会用到，所以我们定义好了，并且在头文件中声明了，所以我们直接调用，无需再次定义声明。下图 1.0 是头文件中的声明。

```
    void Get_Offset();
    double Rad2Angle(double rad);
    double Pos2Rad(word pos);
};

extern Arm MyArm;

double mapFloat(double val, double in_min, doubl

#endif
```

图 1.0

1.0 机械臂 Arm 类中的变量

该类在 Arm.h 里定义，具体请查看该文件。

Table0．给出了 Arm 的公有变量列表

公有变量名(public)	描述
Steer_Num	机械臂中现有的舵机数量
offPos	机械臂的各个舵机偏差，得到舵机数量后，利用动态数组确定其大小
theta	机械臂的各个舵机弧度值，得到舵机数量后，利用动态数组确定其数组大小
steer	机械臂的各个舵机对象，得到舵机数量后，利用动态数组确定其数组大小

Table1．给出了 Arm 的私有变量列表

私有变量名(private)	描述
comSer	主控板与电脑通信的串口选择
pos_goal	机械臂的各个舵机目标位置，得到舵机数量后，利用动态数组确定其数组大小

1.1 Arm 类中的公有函数列表

Table2．给出了 Arm 的公有函数列表

函数名	描述
begin	机械臂配置和通信初始化
position_init	机械臂位置初始化
inverse_movement	机械臂坐标位置逆解函数
move_to_position	机械臂运动控制函数
Para_Init	机械臂参数初始化函数
Set_Arm_Torque_On	机械臂扭矩开启函数
Set_Arm_Torque_Off	机械臂扭矩关闭函数
turn_steer_345_to_positon	第 3，第 4，第 5 号舵机旋转运动函数
Get_Offset	得到机械臂的偏置函数
offset_by_pos	通过机械臂的直接数据设置机械臂偏置函数
offset_by_angle	通过机械臂的角度设置机械臂偏置函数
Rad2Angle	弧度值转角度值函数
Pos2Rad	直接位置数据转换弧度值函数

1.2.0 函数 begin

Table3．描述了函数 begin

Table3.

函数名	begin
函数原型	void begin(HardwareSerial *desireSer)
功能描述	机械臂配置和通信初始化
输入参数	desireSer：串口选择(根据自己的开发板实际确定)
返回值	无

先决条件	无
被调用函数	Steer_Detect(私有); Para_Init; Get_Offset;

例:

/**初始化通信串口为 USB_SER 通信，也就是 Serial 通信，USB_SER 是&Serial 的宏定义 **/
MyArm.begin(USB_SER);

1.2.1 函数 position_init

Table4. 描述了函数 position_init

Table4.

函数名	position_init
函数原型	Void position_init(void)
功能描述	机械臂位置初始化函数
输入参数	无
返回值	无
先决条件	无
被调用函数	Set_Steer_position_runtime

例:

/**初始化通信串口为 USB_SER，并且实现机械臂位置初始化 **/
MyArm.begin(USB_SER);
MyArm. position_init();

1.2.2 函数 inverse_movement（重载 1）

注意：inverse_movement 有 2 个重载，这是第 1 个

Table5. 描述了函数 inverse_movement

Table5.

函数名	inverse_movement
函数原型	void inverse_movement(double x_ , double y_, double z_)
功能描述	机械臂坐标位置逆解函数（由末端坐标逆解出前三个舵机的角度值）
输入参数 1	x_ ：机械臂末端的 x_坐标。
输入参数 2	Y_ ：机械臂末端的 y_坐标
输入参数 3	Z_ ：机械臂末端的 z_坐标
返回值	无
先决条件	无
被调用函数	atan; acos;

例:

/**初始化通信串口为 USB_SER，并且求解(120,120, 120)的角度值**/
MyArm.begin(USB_SER);
MyArm. inverse_movement (120,120, 120);
Serial.print(….);//略

1.2.3 函数 inverse_movement（重载 2）

注意：inverse_movement 有 2 个重载，这是第 2 个

Table6. 描述了函数 inverse_movement

Table6.

函数名	inverse_movement
函数原型	void inverse_movement(PVector pt);
功能描述	机械臂坐标位置（由 pt 提供）逆解函数（由末端坐标逆解出前三个舵机的角度值）
输入参数	pt: 点向量（存储机械臂末端的点坐标）
返回值	无
先决条件	无
被调用函数	atan; acos;

例:

/**初始化通信串口为 USB_SER，并且求解(120,120, 120)的角度值**/

MyArm.begin(USB_SER);

PVector p(120,120,120)

MyArm. inverse_movement (p);

1.2.4 函数 move_to_position（重载 1）

注意：move_to_position 有 3 个重载，这是第 1 个

Table7. 描述了函数 move_to_position

Table7.

函数名	move_to_position
函数原型	void move_to_position(double x_ , double y_, double z_ , word runtime);
功能描述	(3 个参数代表机械臂末端位置)，机械臂在 runtime 时间内运行到该坐标位置。
输入参数 1	X_: 机械臂末端的 x 轴坐标位置，单位 mm
输入参数 2	Y_: 机械臂末端的 y 轴坐标位置，单位 mm
输入参数 3	Z_: 机械臂末端的 z 轴坐标位置，单位 mm
输入参数 4	runtime：运行到目标位置的时间，单位 ms
返回值	无
先决条件	无
被调用函数	inverse_movement

例:

/**初始化通信串口为 USB_SER，并且在两秒内运动到(120,120, 120)坐标位置**/

MyArm.begin(USB_SER);

MyArm. move_to_position(120,120, 120，2000);

1.2.5 函数 move_to_position（重载 2）

注意：move_to_position 有 3 个重载，这是第 2 个

Table8. 描述了函数 move_to_position

Table8.

函数名	move_to_position

函数原型	void move_to_position(PVector pt , word runtime);
功能描述	(3 个参数代表机械臂末端位置)，机械臂在 runtime 时间内运行到该坐标位置。
输入参数 1	Pt：机械臂末端坐标位置（由 pt 提供）
输入参数 2	runtime：运行到目标位置的时间，单位 ms
返回值	无
先决条件	无
被调用函数	inverse_movement

例:

/**初始化通信串口为 USB_SER，并且在两秒内运动到(120,120, 120)坐标位置**/

MyArm.begin(USB_SER);

PVector pt(120,120,120);

MyArm. move_to_position(pt, 2000);

1.2.6 函数 move_to_position（重载 3）

注意：move_to_position 有 3 个重载，这是第 3 个

Table9. 描述了函数 move_to_position

Table9.

函数名	move_to_position
函数原型	void move_to_position(word pos0 , word pos1, word pos2 , word runtime);
功能描述	(3 个参数代表机械臂末端位置)，机械臂在 runtime 时间内运行到该坐标位置。
输入参数 1	pos0：机械臂的舵机 0 的直接控制数据（0~4095）
输入参数 2	Pos1：机械臂的舵机 1 的直接控制数据（0~4095）
输入参数 3	pos2：机械臂的舵机 2 的直接控制数据（0~4095）
输入参数 2	runtime：运行到目标位置的时间，单位 ms
返回值	无
先决条件	无
被调用函数	inverse_movement

例:

/**初始化通信串口为 USB_SER，并且在两秒内运动到各个舵机直角位置**/

MyArm.begin(USB_SER);

MyArm. move_to_position(2047,2047,2047 , 2000);

1.2.7 函数 offset_by_pos

Table10. 描述了函数 offset_by_pos

Table10.

函数名	offset_by_pos
函数原型	void offset_by_pos(byte id, short offset);
功能描述	通过机械臂的直接位置，设置机械臂偏置
输入参数 1	Id: 舵机的 ID 号
输入参数 2	Offset: 设置舵机的偏置，值的范围（-2046 ~ +2046）

返回值	无
先决条件	无
被调用函数	无

例:

/**初始化通信串口为 USB_SER，并且给舵机 1 的偏置设为 200**/

MyArm.begin(USB_SER);

MyArm. offset_by_pos(1,200);

1.2.8 函数 offset_by_angle

Table11. 描述了函数 offset_by_angle

Table11.

函数名	offset_by_angle
函数原型	void offset_by_angle(byte id, double angle);
功能描述	通过机械臂的角度，设置机械臂偏置
输入参数 1	Id: 舵机的 ID 号
输入参数 2	angle: 设置舵机的偏置角度，范围（-90~+90）
返回值	无
先决条件	无
被调用函数	无

例:

/**初始化通信串口为 USB_SER，并且给舵机 1 的偏置设为 10° **/

MyArm.begin(USB_SER);

MyArm. offset_by_ angle (1,10);

1.2.9 函数 Set_Arm_Torque_Off

Table12. 描述了函数 Set_Arm_Torque_Off

Table12.

函数名	Set_Arm_Torque_Off
函数原型	void Set_Arm_Torque_Off (void);
功能描述	机械臂扭矩关闭函数 设置机械臂的扭矩为关：使机械臂失去扭矩
输入参数	void
返回值	无
先决条件	无
被调用函数	无

例:

/**初始化通信串口为 USB_SER，并且使机械臂失去扭矩**/

MyArm.begin(USB_SER);

MyArm. Set_Arm_Torque_Off ();

1.2.10 函数 Set_Arm_Torque_On

Table13. 描述了函数 Set_Arm_Torque_On

Table13.

函数名	Set_Arm_Torque_On
函数原型	void Set_Arm_Torque_On(void);
功能描述	机械臂扭矩开启函数　设置机械臂的扭矩为开：使机械臂恢复扭矩
输入参数	void
返回值	无
先决条件	无
被调用函数	无

例:

/**初始化通信串口为 USB_SER，并且使机械臂恢复扭矩**/

MyArm.begin(USB_SER);

MyArm. Set_Arm_Torque_On ();

1.2.11 函数 turn_steer_345_to_positon(重载 1)

注意：该函数有两个重载，这是第 1 个

Table14. 描述了函数 turn_steer_345_to_positon

Table14.

函数名	turn_steer_345_to_positon
函数原型	boolean turn_steer_345_to_positon(word pos3 , word pos4, word pos5 , word runtime);
功能描述	第 3，第 4，第 5 号舵机旋转运动函数 重要：舵机 345 的状态设置函数
输入参数 1	pos3：舵机 3 的位置，范围（0~4095），表示（0~360°） 注意:：如果开启保护，以保护中的位限为主
输入参数 2	Pos4：舵机 4 的位置，范围（0~4095），表示（0~360°） 注意:：如果开启保护，以保护中的位限为主
输入参数 3	Pos5：舵机 5 的位置，范围（0~4095），表示（0~360°） 注意:：如果开启保护，以保护中的位限为主
输入参数 4	runtime: 机械臂由当前位置运行到指定位置所花的时间，单位是毫秒
返回值	无
先决条件	无
被调用函数	Set_Steer_position_runtime

例:

/**初始化通信串口为 USB_SER，并且使机械臂的 345 号舵机在两秒内运行到正中位置**/

MyArm.begin(USB_SER);

MyArm. Set_Steer_position_runtime (2047,2047,2047, 2000);

1.2.12 函数 turn_steer_345_to_positon(重载 2)

注意：该函数有两个重载，这是第 2 个

Table15. 描述了函数 turn_steer_345_to_positon

Table15.

函数名	turn_steer_345_to_positon

函数原型	Boolean turn_steer_345_to_positon(double angle3 , double angle4, double angle5 , word runtime);
功能描述	第 3，第 4，第 5 号舵机旋转运动函数 重要：舵机 345 的状态设置函数
输入参数 1	angle3: 舵机 3 的角度，范围（0~360） 注意:：如果开启保护，以保护中的位限为主
输入参数 2	angle4:舵机 4 的角度，范围（0~360） 注意:：如果开启保护，以保护中的位限为主
输入参数 3	angle5: 舵机 5 的角度，范围（0~360） 注意:：如果开启保护，以保护中的位限为主
输入参数 4	runtime: 机械臂由当前位置运行到指定位置所花的时间，单位是毫秒
返回值	无
先决条件	无
被调用函数	Set_Steer_position_runtime

例:

/**初始化通信串口为 USB_SER，并且使机械臂的 345 号舵机在两秒内运行到正中位置**/

MyArm.begin(USB_SER);

MyArm. Set_Steer_position_runtime (90,90,90, 2000);

1.2.13 函数 Get_Offset

Table16. 描述了函数 Get_Offset

Table16.

函数名	Get_Offset
函数原型	void Get_Offset()
功能描述	得到机械臂的偏置函数
返回值	无
先决条件	无
被调用函数	Get
解释	得到的数据存在动态数组 offset 里

例:

/*初始化通信串口为 USB_SER，并且得到机械臂各个舵机的偏置（舵机位置的直接数据）*/

MyArm.begin(USB_SER);

MyArm.Get_Offset();

1.2.14 函数 Rad2Angle

Table17. 描述了函数 Rad2Angle

Table17.

函数名	Rad2Angle
函数原型	double Rad2Angle(double rad);
功能描述	弧度值转角度值函数
返回值	double：返回角度值
先决条件	无

被调用函数	mapFloat

例:

/*初始化通信串口为 USB_SER，并且输入弧度为 2，测试输出弧度*/

MyArm.begin(USB_SER);

MyArm. Rad2Angle(2);

1.2.15 函数 Pos2Rad

Table18. 描述了函数 Pos2Rad

Table18.

函数名	Pos2Rad
函数原型	double Pos2Rad (double pos);
功能描述	直接位置数据转换弧度值函数
返回值	double：返回弧度值
先决条件	无
被调用函数	mapFloat

例:

/*初始化通信串口为 USB_SER，并且输入直接位置数据 2047，测试输出弧度值*/

MyArm.begin(USB_SER);

MyArm. Pos2Rad (2047);

第二章 点向量基本函数(PVector.h)

2.0 机械臂 PVector 类中的变量

该类在 PVector.h 里定义，具体请查看该文件。

Table19．给出了 PVector 的公有变量列表

公有变量名(public)	描述
x, y, z;	机械臂末端的坐标位置

Table20．给出了 PVector 的私有变量列表

私有变量名(private)	描述
无	无

2.1 PVector 类中的公有函数列表

Table2．给出了 PVector 的公有函数列表

函数名	描述
PVector	构造函数（有两个重载）
set_xyz	机械臂位置初始化
add	机体坐标自加函数
sub	机体坐标相减函数（注意：不是自减）
normalize	归一化函数
dot	点乘函数
dist	点距函数

2.2.0 构造函数 PVector（重载 1）

注意：该函数有两个重载，这是重载 1

Table21. 描述了函数 PVector

Table21.

函数名	PVector
函数原型	Void position_init(void)
功能描述	构造函数：初始化机体坐标
输入参数	无
返回值	无
先决条件	无
被调用函数	无

例:

/**创建并初始化一个点向量对象**/

PVector pt();

2.2.1 构造函数 PVector（重载 2）

注意：该函数有两个重载，这是重载 2

Table22. 描述了函数 PVector

Table22.

函数名	PVector
函数原型	Void position_init(double _x, double _y, double _z)
功能描述	构造函数：初始化机体坐标
输入参数 1	_x: 点向量的 x 方向的坐标
输入参数 2	_y: 点向量的 y 方向的坐标
输入参数 3	_z: 点向量的 z 方向的坐标
返回值	无
先决条件	无
被调用函数	无

例:

```
/**创建并初始化一个点向量对象,坐标为(120,120,100)**/
PVector pt(120,120,100);
```

2.2.2 函数 set_xyz

Table23. 描述了函数 set_xyz

Table23.

函数名	set_xyz
函数原型	Void set_xyz(double _x, double _y, double _z)
功能描述	设置机体坐标
输入参数 1	_x: 点向量的 x 方向的坐标
输入参数 2	_y: 点向量的 y 方向的坐标
输入参数 3	_z: 点向量的 z 方向的坐标
返回值	无
先决条件	初始化一个点向量(PVector)对象
被调用函数	无

例:

```
/**创建并初始化一个点向量对象,坐标设为(120,120,100)**/
PVector pt();
pt.set_xyz(120,120,100);
```

2.2.3 函数 add

Table24. 描述了函数 add

Table24.

函数名	add
函数原型	void add(PVector p)
功能描述	机体坐标自加函数（表示给机体的一个坐标增量）
输入参数	P: PVector 的一个对象，具体作用是作为机体的增量值存在
返回值	无
先决条件	初始化一个点向量(PVector)对象
被调用函数	无

例:

```
/**创建并初始化一个点向量对象,坐标设为(120,120,100),其自增坐标为(10,20,10)**/
```

```
PVector pt(120, 120, 100);
PVector p(10, 20, 10);
pt.add(p);
```

2.2.4 函数 sub

Table25. 描述了函数 sub

Table25.

函数名	sub
函数原型	PVector sub (PVector p)
功能描述	机体坐标相减函数（注意：不是自减）,返回结果点向量
输入参数	P: PVector 的一个对象，具体作用是作为一个被减掉的坐标，最后把结果返回，原点向量的值并没有变化
返回值	PVector：PVector 的一个对象，作为回传相减后的值
先决条件	初始化一个点向量(PVector)对象
被调用函数	无

例:

```
/**创建并初始化两个点向量对象,一个坐标设为(120,120,100),另一个坐标为(10,20,10)，返回第一个点向量减去第二个点向量的坐标值**/
PVector pt(120, 120, 100);
PVector p(10, 20, 10);
PVector p_tmp();                    //存储相减后的结果
P_tmp = pt.sub(p);
```

2.2.5 函数 normalize

Table26. 描述了函数 normalize

Table26.

函数名	normalize
函数原型	Void normalize()
功能描述	归一化函数
输入参数	无
返回值	无
先决条件	初始化一个点向量对象
被调用函数	无

例:

```
/**创建并初始化一个点向量对象,坐标设为(120,120,100)，然后在归一化**/
PVector pt(120, 120, 100);
pt.normalize();
```

2.2.6 函数 dot

Table27. 描述了函数 dot

Table27.

函数名	dot

函数原型	double dot(PVector p)
功能描述	点乘函数
输入参数	无
返回值	无
先决条件	初始化一个点向量对象
被调用函数	无

例:

```
/**创建并初始化 2 个点向量对象,坐标设为(20,20,10)，另一个坐标设为(6,6,10) **/
PVector pt(20, 20, 10);
PVctor p(6,6,10);
pt.dot(p);
```

2.2.6 函数 dist

Table28. 描述了函数 dist

Table28.

函数名	dist
函数原型	double dist (PVector p)
功能描述	点距函数
输入参数	无
返回值	无
先决条件	初始化一个点向量对象
被调用函数	无

例:

```
/**创建并初始化 2 个点向量对象,坐标设为(20,20,10)，另一个坐标设为(6,6,10) **/
PVector pt(20, 20, 10);
PVctor p(6,6,10);
double ds = 0;
ds = pt. dist (p);
```

第三章 机械臂学习功能函数(Arm_learn.h)

3.0 机械臂 Arm_learn 类中的变量

该类在 Arm_learn.h 里定义，具体请查看该文件。

Table29．给出了 Arm_learn 的公有变量列表

公有变量名(public)	描述
button0_time_cnt	记录按键 0 经过了几个 10ms
button1_time_cnt	记录按键 1 经过了几个 10ms
rom_offset	记录学习数据的地址偏置
record_point_num	记录学习的位点个数（最后写在 eeprom）
start_learn_flag	开始学习标志位
record_flag	记录标志位

Table1．给出了 PVector 的私有变量列表

私有变量名(private)	描述
无	无

3.1 Arm_learn 类中的公有函数列表

Table30．给出了 Arm_learn 的公有函数列表

函数名	描述
Arm_learn	构造函数
button_learn_detect	按键检测函数
record_one_point	位点记录函数
start_learn	机械臂学习程序
reappear_learn	机械臂学习姿态重现函数
get_steer_positon	得到机械臂位姿（采用了三次均值滤波）

3.2.0 构造函数 Arm_learn

Table31．描述了函数 Arm_learn

Table31.

函数名	Arm_learn
函数原型	Arm_learn (void)
功能描述	构造函数：初始化机体坐标
输入参数	无
返回值	无
先决条件	无
被调用函数	无

例:

```
/**创建并初始化一个机械臂学习的对象**/
Arm_learn MyArm_learn;
```

3.2.1 函数 button_learn_detect

Table32. 描述了函数 button_learn_detect

Table32.

函数名	button_learn_detect
函数原型	void button_learn_detect(void)
功能描述	按键检测函数
输入参数	无
返回值	无
先决条件	初始化一个机械臂学习对象
被调用函数	无
说明	该函数获取的值存储在按键时间侦测的公有变量里

例:

/**创建并初始化一个机械臂学习的对象，然后去侦测各个按键的按键时长**/

Arm_learn MyArm_learn;

MyArm_learn. button_learn_detect();

3.2.2 函数 record_one_point(一般不需要用户去调用)

Table33. 描述了函数 record_one_point

Table33.

函数名	record_one_point
函数原型	void record_one_point(void)
功能描述	位点记录函数
输入参数	无
返回值	无
先决条件	初始化一个机械臂学习对象
被调用函数	无
说明	该函数一般放在 start_learn 中调用

3.2.3 函数 start_learn

Table34. 描述了函数 start_learn

Table34.

函数名	start_learn
函数原型	void start_learn(void)
功能描述	机械臂学习程序
输入参数	无
返回值	无
先决条件	初始化一个机械臂学习对象
被调用函数	button_learn_detect; record_one_point

例:

/**创建并初始化一个机械臂学习的对象，然后去侦测各个按键的按键时长**/

Arm_learn MyArm_learn;

MyArm_learn. start_learn();

3.2.4 函数 reappear_learn

Table35. 描述了函数 reappear_learn

Table35.

函数名	reappear_learn
函数原型	void reappear_learn()
功能描述	机械臂学习姿态重现函数
输入参数	无
返回值	无
先决条件	初始化一个机械臂学习对象
被调用函数	move_to_position; turn_steer_345_to_positon

例:

/**创建并初始化一个机械臂学习的对象，然后去重现学习到的参数**/

Arm_learn MyArm_learn;

MyArm_learn. button_learn_detect();

3.2.5 函数 get_steer_positon

Table36. 描述了函数 get_steer_positon

Table36.

函数名	get_steer_positon
函数原型	word get_steer_positon(byte i)
功能描述	得到机械臂位姿（采用了三次均值滤波）
输入参数	无
返回值	无
先决条件	初始化一个机械臂学习对象
被调用函数	Get_Steer_Position_Current_Inf;

例:

/**创建并初始化一个机械臂学习的对象，然后去得到舵机 1 的当前位姿**/

Arm_learn MyArm_learn;

MyArm_learn. get_steer_positon (3);