



25 Valleywood Drive, Unit 20  
Markham, Ontario, L3R 5L9, Canada  
Tel: (905) 943-9572 Fax: (905) 943-9197

**WiRobot SDK**  
**Application Programming Interface (API)**  
**Supplementary for Sentinel Robot**  
**Reference Manual**  
**- (For MS Windows)**

***Dr Robot***<sup>®</sup>

**Version: 1.9**  
**SEP. 2006**

## Table of Contents

<b>I.</b>	<b>Power controller API</b> .....	<b>2</b>
I.1.	short GetVolRef( ) .....	2
I.2.	short GetBat1Vol( ) .....	2
I.3.	short GetBat1Temp( ) .....	2
I.4.	short GetBat2Vol( ) .....	2
I.5.	short GetBat2Temp( ) .....	3
I.6.	short GetDCINVol( ) .....	3
I.7.	Byte GetCHGPath( ) .....	3
I.8.	Byte GetPowerPath( ) .....	3
I.9.	Byte GetCHGStatus( ) .....	4
I.10.	void SendCmdCHG( ) .....	4
<b>II.</b>	<b>Constellation system API</b> .....	<b>6</b>
II.1.	short GetGPS01( ) .....	6
II.2.	short GetGPS02 ( ) .....	6
II.3.	short GetGPS03 ( ) .....	6
II.4.	short GetGPS04 ( ) .....	6
II.5.	short GetGPS05 ( ) .....	7
II.6.	short GetGPS06 ( ) .....	7
II.7.	short GetGPS07 ( ) .....	7
II.8.	short GetGPS08 ( ) .....	8
II.9.	void setGPSID(Byte id1, Byte id2, Byte id3, Byte id4) .....	8
II.10.	Byte getSenID1 ( ) .....	8
II.11.	Byte getSenID2 ( ) .....	8
II.12.	Byte getSenID3 ( ) .....	9
II.13.	Byte getSenID4 ( ) .....	9
II.14.	Example of using the Constellation system .....	9

This document contains two parts, one for power controller and the other is for Constellation system.

## I. Power controller API

### I.1. short GetVolRef( )

Description:

GetVolRef returns the power controller AD reference voltage (AD value)

Syntax:                    ival = GetVolRef ();

Parameter:                void

Return value:            short    ival

Remarks:

The AD reference Voltage

$$V_{ref} = ((ival / 4095) * 3.0) * 2$$

### I.2. short GetBat1Vol( )

Description:

GetBat1Vol returns the battery1 voltage (AD value)

Syntax:                    ival = GetBat1Vol ();

Parameter:                void

Return value:            short    ival

Remarks:

The Battery1 voltage

$$VBat1 = (ival / 4095 * V_{ref}) * 8$$

### I.3. short GetBat1Temp( )

Description:

GetBat1Temp returns the battery1 temperature sensor (AD value)

Syntax:                    ival = GetBat1Temp ();

Parameter:                void

Return value:            short    ival

The value must be in the range of 1100 – 2500, if beyond this range, stop charging the battery

### I.4. short GetBat2Vol( )

Description:

GetBat2Vol returns the battery2 voltage (AD value)

Syntax:                    ival = GetBat2Vol ();

Parameter:                void

Return value:            short    ival

Remarks:

The Battery2 voltage

$$VBat2 = (ival / 4095 * V_{ref}) * 8$$

### 1.5. short GetBat2Temp( )

Description:

GetBat2Temp returns the battery2 temperature sensor (AD value)

Syntax:                    ival = GetBat2Temp ();

Parameter:                void

Return value:            short    ival

The value must be in the range of 1100 – 2500, if beyond this range, stop charging the battery

### 1.6. short GetDCINVol()

Description:

GetDCINVol returns the DCIN voltage (AD value)

Syntax:                    ival = GetDCINVol ();

Parameter:                void

Return value:            short    ival

Remarks:

The DCIN voltage

$$VBat2 = (ival / 4095 * Vref) * 8$$

### 1.7. Byte GetCHGPath( )

Description:

GetCHGPath returns the charge path

Syntax:                    ival = GetCHGPath ();

Parameter:                void

Return value:            Byte    ival

Bit	Description
0	X
1	X
2	X
3	X
4	X
5	X
6	Charge Battery 2
7	Charge Battery 1

### 1.8. Byte GetPowerPath( )

Description:

GetPowerPath returns the power path of controller

Syntax:                    ival = GetPowerPath ();

Parameter:                void

Return value:            Byte    ival

Bit	Description
0	X
1	X
2	X
3	X
4	X
5	Power by DCIN
6	Power by Battery 2
7	Power by Battery 1

### I.9. Byte GetCHGStatus( )

Description:

GetCHGStatus returns the status of power controller.

Syntax:                    ival = GetCHGStatus ();

Parameter:                void

Return value:            Byte    ival

Bit	Description
0	X
1	X
2 – CH	1 – Charging 0 – no charge
3 – PF	Power fail
4 – DC	DCDIV comparator output
5 – LP	Lower power
6 – FA	Fault(on reset, short circuit, shut down by DCDIV
7	X

### I.10. void SendCmdCHG( )

Description:

SendCmdCHG sends the command to power controller.

Syntax:                    void SendCmdCHG (byte ival);

Parameter:                byte ival

Return value:            void

In this version, Power1 is camera power, and power2 is motion controller and PDA power, power3 is servo and servo controller power.

Bit	Description
0	Turn On/Off Power1 0 – Turn Off, 1 – Turn On
1	Reset Power2 1 – power 0 – reset 200ms
2	power3 1 – Turn On 0 – Turn Off
3	Power by DCIN
4	Power by Bat2

5	Power by Bat1
6	Charge Bat2 1 – charge 0 – stop charge
7	Charge Bat1 1 – charge 0 – stop charge

## II. Constellation system API

### II.1. short GetGPS01( )

Description:

Get the distance measurement (unit: mm) from the LEFT Constellation receiver module on the car to the selected Constellation transponder (ID = ID1).

Syntax:                   ival = GetGPS01 ( );

Parameter:               void

Return value:           short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID1 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

### II.2. short GetGPS02 ( )

Description:

Get the distance measurement (unit: mm) from the LEFT Constellation receiver on the car to the selected Constellation transponder (ID = ID2).

Syntax:                   ival = GetGPS02 ( );

Parameter:               void

Return value:           short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID2 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

### II.3. short GetGPS03 ( )

Description:

Get the distance measurement (unit: mm) from the LEFT Constellation receiver on the car to the selected Constellation transponder (ID = ID3).

Syntax:                   ival = GetGPS03 ( );

Parameter:               void

Return value:           short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID3 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

### II.4. short GetGPS04 ( )

Description:

Get the distance measurement (unit: mm) from the LEFT Constellation receiver on the car to the selected Constellation transponder (ID = ID4).

Syntax:                    ival = GetGPS04 ( );

Parameter:                void

Return value:             short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID4 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

## II.5. short GetGPS05 ( )

Description:

Get the distance measurement (unit: mm) from the RIGHT Constellation receiver on the car to the selected Constellation transponder (ID = ID1).

Syntax:                    ival = GetGPS05 ( );

Parameter:                void

Return value:             short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID1 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

## II.6. short GetGPS06 ( )

Description:

Get the distance measurement (unit: mm) from the RIGHT Constellation receiver on the car to the selected Constellation transponder (ID = ID2).

Syntax:                    ival = GetGPS06 ( );

Parameter:                void

Return value:             short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID2 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

## II.7. short GetGPS07 ( )

Description:

Get the distance measurement (unit: mm) from the RIGHT Constellation receiver on the car to the selected Constellation transponder (ID = ID3).

Syntax:                    ival = GetGPS07 ( );

Parameter:                void

Return value:             short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID3 is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

## II.8. short GetGPS08 ( )

Description:

Get the distance measurement (unit: mm) from the RIGHT Constellation receiver on the car to the selected Constellation transponder (ID = ID<sub>4</sub>).

Syntax:                    ival = GetGPS08 ( );

Parameter:                void

Return value:             short ival

Remarks:

If ival = 0x7fff, it means receiver did not get the distance measurement, the transponder with ID<sub>4</sub> is out of range.

If ival = 0x7ff0, it means receiver did not get any new distance data.

## II.9. void setGPSID(Byte id1, Byte id2, Byte id3, Byte id4)

Description:

Set the ID of Constellation transponder that the robot will be using.

Syntax:                    void setGPSID(Byte id1, Byte id2, Byte id3, Byte id4)

Parameter:                Byte id1, Byte id2, Byte id3, Byte id4

Return value:             void

Remarks:

The sensor with id1, id2 is on the one Constellation transponder, and the sensor with id3, id4 is on another transponder.

## II.10. Byte getSenID1 ( )

Description:

Get the ID of Constellation transponder that the robot is using. The distance measurement that is obtained with getGPS01( ) function is from the LEFT receiver module, and distance measurement that is obtained with getGPS05( ) function is from the RIGHT receiver module.

Syntax:                    ival = getSenID1( )

Parameter:                void

Return value:             byte ival

## II.11. Byte getSenID2 ( )

Description:

Get the ID of Constellation transponder that the robot is using. The distance measurement that is obtained with getGPS02( ) function is from the LEFT receiver module, and distance measurement that is obtained with getGPS06( ) function is from the RIGHT receiver module.

Syntax:                    ival = getSenID2( )

Parameter:                void

Return value:             byte ival

## II.12. Byte getSenID3 ( )

Description:

Get the ID of Constellation transponder that the robot is using. The distance measurement that is obtained with getGPS03( ) function is from the LEFT receiver module, and distance measurement that is obtained with getGPS07( ) function is from the RIGHT receiver module.

Syntax:                    ival = getSenID3( )

Parameter:                void

Return value:            byte ival

## II.13. Byte getSenID4 ( )

Description:

Get the ID of Constellation transponder that the robot is using. The distance measurement that is obtained with getGPS04( ) function is from the LEFT receiver module, and distance measurement that is obtained with getGPS08( ) function is from the RIGHT receiver module.

Syntax:                    ival = getSenID4( )

Parameter:                void

Return value:            byte ival

## II.14. Example of using the Constellation system

Here is the example for calculation the robot position based on the distance measurement from the Constellation system.

### Step 1:

Set the transponder ID to 1,2,3,4:

```
setGPSID(1,2,3,4);
```

### Step 2:

Get the distance in the custom sensor event.

```
//First the program should confirm the transponder ID is right.  
ID1 = getSenID1( );  
ID2 = getSenID2( );  
ID3 = getSenID3( );  
ID4 = getSenID4( );  
//then get the distance unit :mm  
If ( ID1 = 1 ) and ( ID2 = 2 ) or ( ID3 = 3 ) or ( ID4 = 4 ) {  
    //get distance here  
Dis_ID1_L = getGPS01 ( );  
Dis_ID2_L = getGPS02 ( );  
Dis_ID3_L = getGPS03 ( );  
Dis_ID4_L = getGPS04 ( );  
Dis_ID1_R = getGPS05 ( );
```

```

Dis_ID2_R = getGPS06 ( );
Dis_ID3_R = getGPS07 ( );
Dis_ID4_R = getGPS08 ( );
}

```

### Step 3:

Calculate the position based on the information of distance measurement.  
 Here is the example for transponder 1( with sensor ID = 1, sensor ID = 2)

```

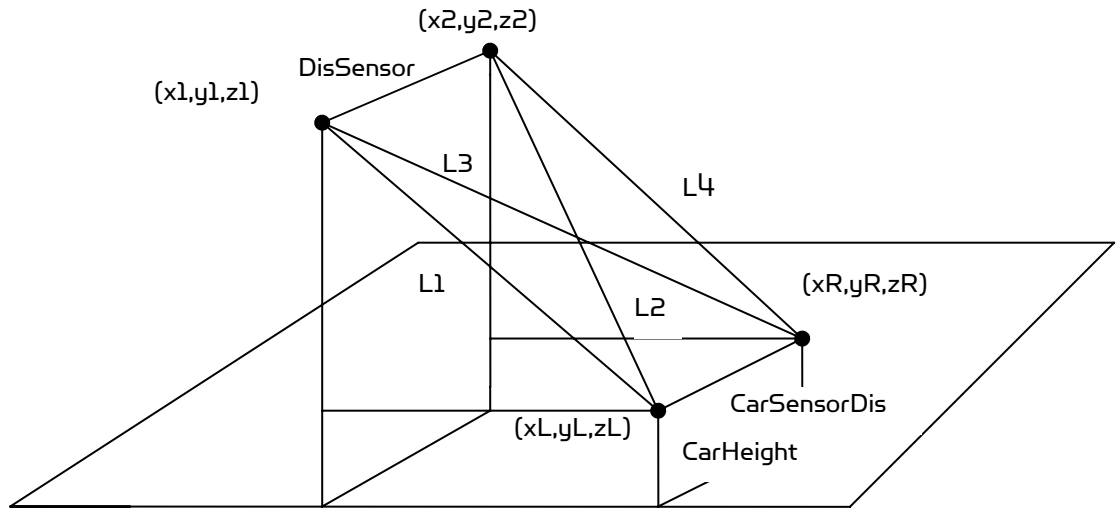
Sensor ID1 position (X1,Y1,Z1);
Sensor ID2 position (X2,Y2,Z2);
CarHeight = 225; //unit: mm the height of Constellation receiver on the robot
CarSensorDis = 300; // unit: mm the distance between the left receiver and right
// receiver on the robot
DisSensor = sqrt( (X1- X2) & 2 + (Y1 - Y2 ) & 2); //the distance between two
//sensors on the transponder
L1 = Dis_ID1_L , L2 = Dis_ID2_L. L3 = Dis_ID1_R, L4 = Dis_ID2_R.

```

All the data must meet the following conditions

1.  $abs(L1 - L2) < DisSensor$   
and  $abs(L1+L2) > DisSensor$
2.  $abs(L3 - L4) < DisSensor$   
and  $abs(L3+L4) > DisSensor$
3.  $abs(L1 - L3) < CarSensorDis$   
and  $abs(L1+ L3) > CarSensorDis$
4.  $abs(L2- L4) < CarSensorDis$   
and  $abs(L2+L4) > CarSensorDis$

If all the data are effective, then we should be able to calculate the position of left receiver (xL,yL,zL) and right receiver (xR,yR,zR) position by simple triangle functions.



The the robot position  $X = (xL + xR) / 2$   
 $Y = (yL + yR) / 2$   
 $Z = zL = zR = z1 - \text{CarHeight} = z2 - \text{CarHeight}$