

xapmcsSpykee

**Erector/Meccano Spykee Robot
Distributed Computing Node**

Michael McSharry
Sept 6, 2009

Table of Contents

Introduction.....	3
Setup	3
xapmcsHub Setup	3
xapmcsSpykee Setup	7
Spykee Interactive Control	8
Spykee Automated Control.....	10

Table of Figures

Figure 1 xAP Hub Activity Form	4
Figure 2 xAP Viewer	5
Figure 3 xAP Viewer Message Detail	6
Figure 4 xapmcsSpykee tray icon	7
Figure 5 xapmcsSpykee Interactive Form	7
Figure 6 xapmcsSpykee Video/Motion Form.....	8
Figure 7 Sample Command File	9
Figure 8 Command Proof File Sample	10

Table of Tables

Table 1 xapBSC.Cmd Message Example	1
Table 2 xapBSC.Event Message Example.....	11
Table 3 xapmcsSpykee End Points and Parameters	11

Introduction

This guide is intended to show the setup and use of xapmcsSpykee to achieve control of the Spykee robot in an xAP environment. xapmcsSpykee is an xAP node that contains a Window form for interactive control and uses the XapBSC schema for automated control. It communicates with Spykee via LAN.

At this time the control and display functions of Spykee related to motion, video, LED, sound bytes and status are integrated. The microphone, VOIP, and MP3 audio functions provide a control mechanism, but have not been integrated into useful functionality. No capability has yet been provided for video surveillance.

As with any xAP environment, these xAP applications will only run on computers that also have a xAP hub running. Any xAP hub can be used. The setup of hub xapmcsHub is included here.

Setup

xapmcsHub Setup

xapmcsHub can be downloaded from <http://mcsSprinklers.com/xapmcsHub.zip>. This application and each xAP application should be located in individual folders. A setup such as \xAP\xapmcsHub, \xap\xapmcsSpykee, etc. is a reasonable folder organization. When the zip file is uncompressed the folder structure within the zip file should be maintained.

All mcs xAP applications operate with access from tray icons. A tray icon will typically provide access to the setup mechanism(s), a communication window if appropriate and application level access if appropriate. There will also be two system-level entries in the tray icon and these are not functional in the distributed software.

xapmcsHub should be first xAP application started on any computer. This may be done in a controlled manner from Windows startup folder or other mechanism that achieves the desired objective. If non-mcs xAP applications are also used and they are started before an xAP hub they may prevent a hub that starts later from operating. If only mcs xAP applications are used then the order is not as critical, but could cause startup delays as the hub learns about other running applications.

xapmcsHub.exe is the executable that is run. Three checkboxes are provided for setup of xapmcsHub when setup is selected from the tray icon. The typical configuration will be as shown in Figure 1 where only xAP will be processed, an icon will appear in the tray and no backup hub function will be spawned. Once stable operations are achieved the tray icon may be removed if one desires a less-cluttered tray.

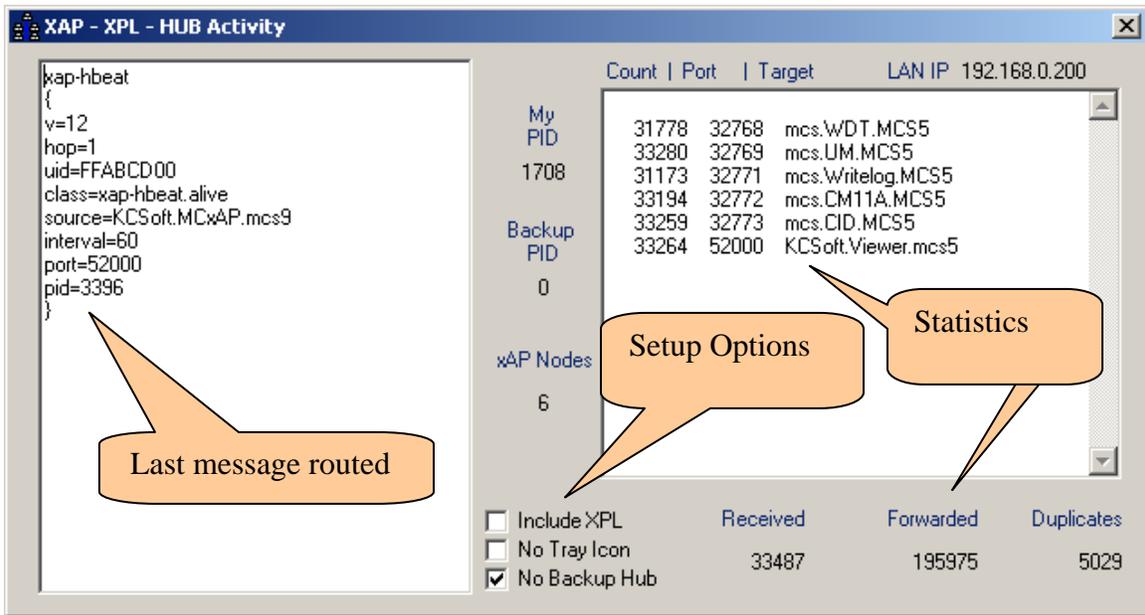


Figure 1 xAP Hub Activity Form

There are a variety of xAP hubs available and any should be able to be used. The xAP Viewer from xapFramework.net is a useful tool to observe xAP message traffic and it can also serve the dual role as an xAP hub. While the following link can be used at the time of this writing to download the xAP Viewer.

http://www.edjo.pwp.blueyonder.co.uk/edward/xAP/xFx/viewer_main.htm

The xAP viewer is an essential tool when working in a xAP environment to understand the communication that is occurring over the LAN.

Figure 2 and Figure 3 show a typical display for messages summaries and message detail that is available from a Window's GUI. When using the xAP viewer and a dedicated hub one needs to be careful to start the dedicated hub before the viewer otherwise the hub will not be able to gain access to the xAP port on the primary interface.

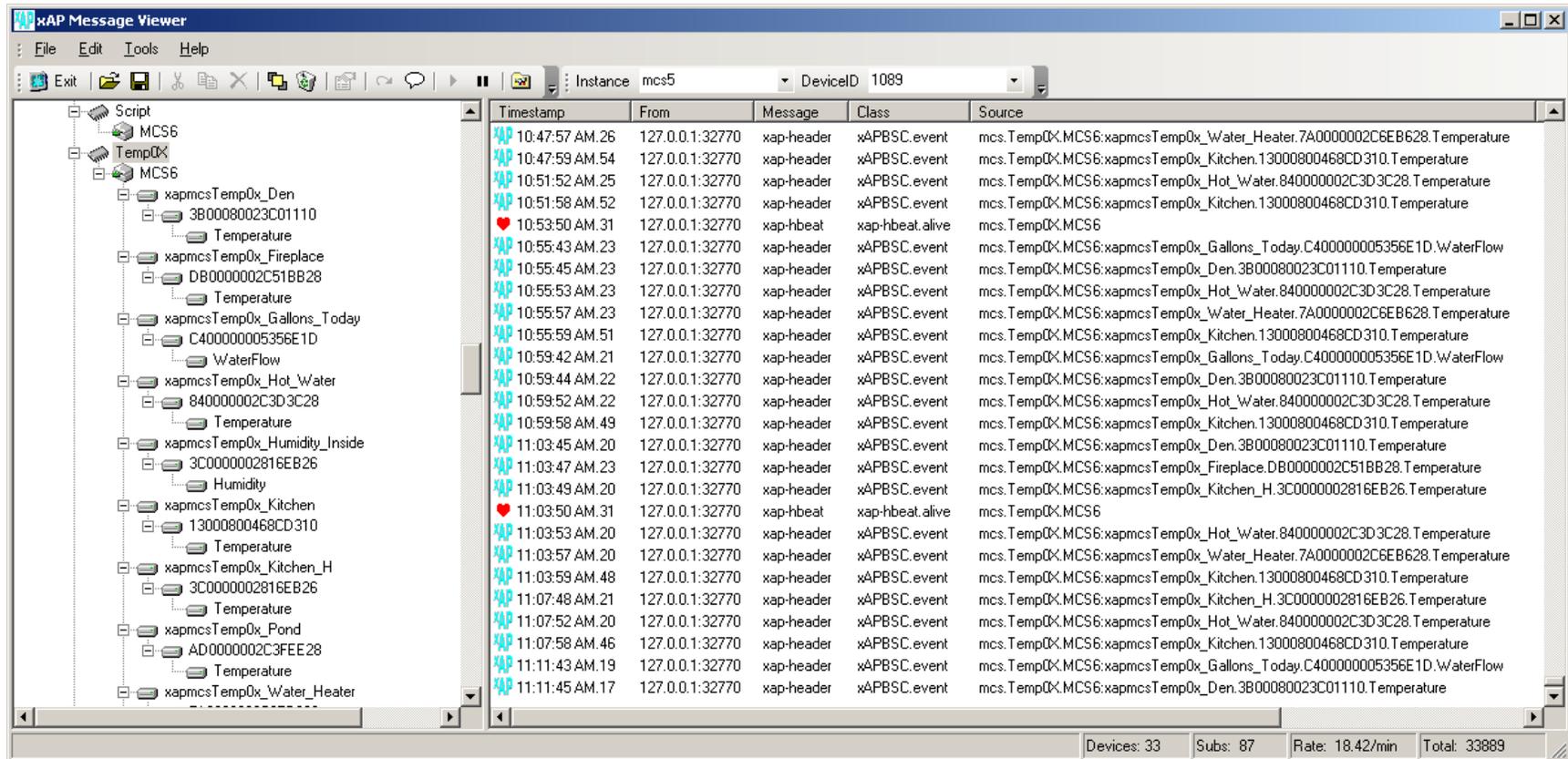


Figure 2 xAP Viewer

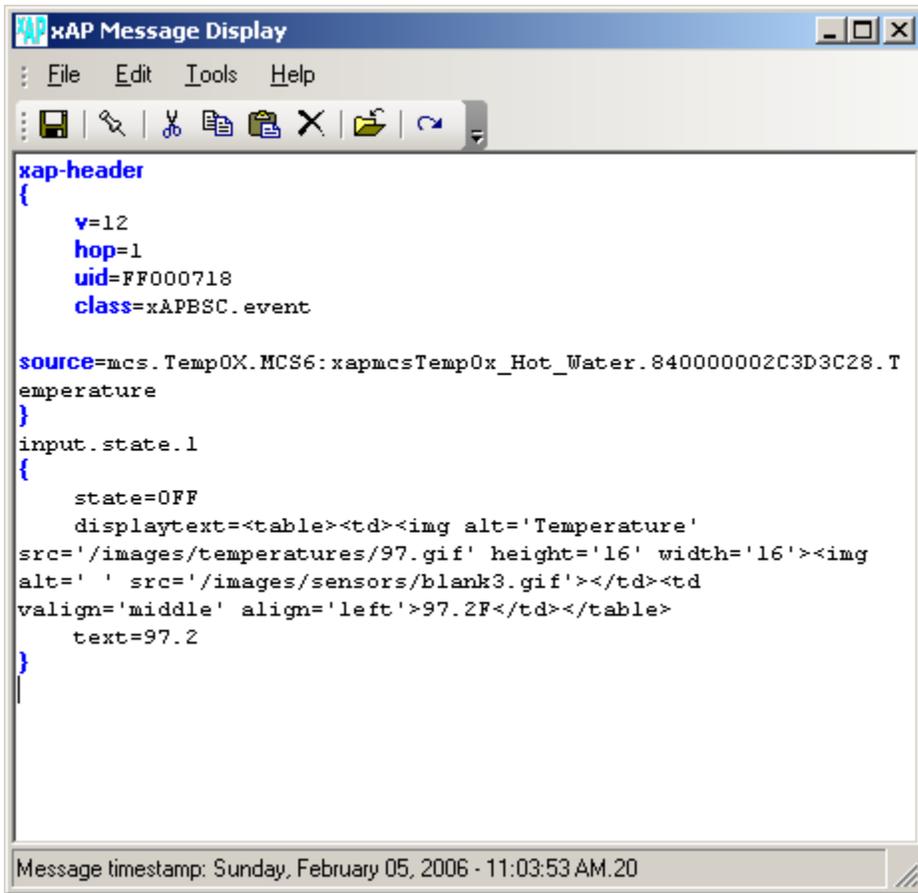


Figure 3 xAP Viewer Message Detail

xpmcsSpykee Setup

The contents of the zip package should be unzipped into a convenient location such as C:\Program Files\xAP\xpmcsSpykee. The executable program is xpmcsSpykee.exe.

xpmcsSpykee is started by running xpmcsSpykee.exe. It will present itself as a tray icon such as is shown in Figure 4. Clicking on the Icon will bring up a pair of forms. One similar to Figure 5 provides the interactive controls. The second form similar to Figure 6 contains the video image provided by Spykee. These forms are not intended to be replacements for the Spykee desktop application provided with Spykee, but are a conduit by which automation with Spykee can be obtained.



Figure 4 xpmcsSpykee tray icon

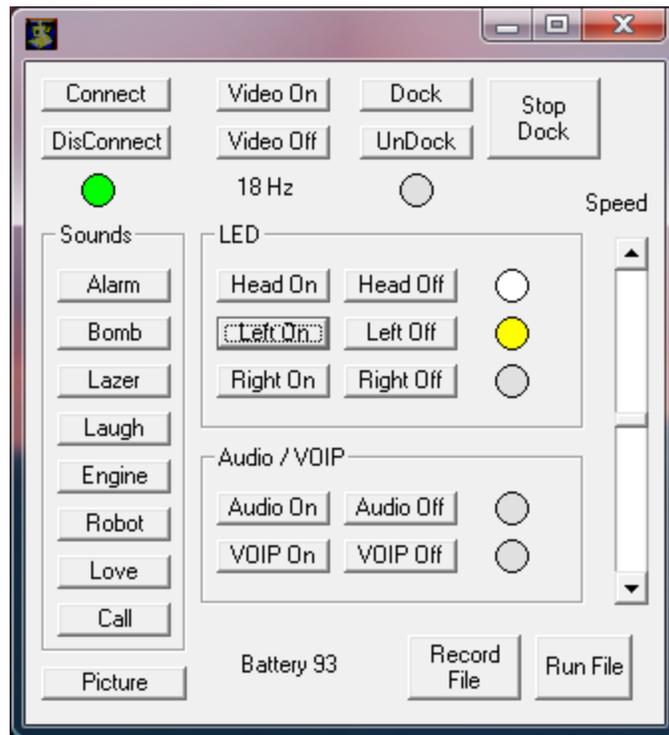


Figure 5 xpmcsSpykee Interactive Form

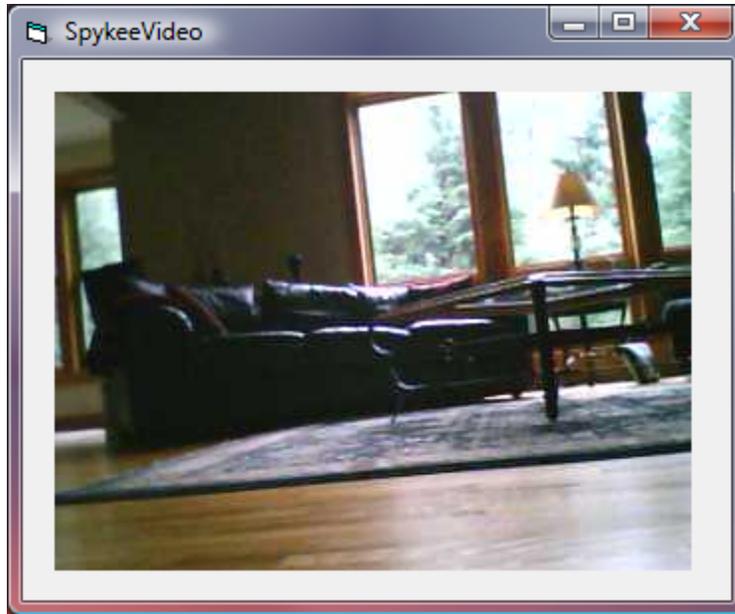


Figure 6 xapmcsSpykee Video/Motion Form

Spykee Interactive Control

The control interface to Spykee is setup to maximize the use of discrete commands with no or very limited parameters. The interactive form provides most of the control with as a button and those that also have a state associated with them contain a round graphic that becomes illuminated when the item is active/on.

When xapmcsSpykee starts it will automatically attempt to login to spykee. The success of the IP connection is shown with a green status in the upper left of the form. The Connect and Disconnect buttons are used to control this connection. On the initial execution of xapmcsSpykee the user login credentials will default to admin/admin. These can be changed. The next time xapmcsSpykee starts the password will be shown as a * for security, but the last entered password will be sent to Spykee during the login.

Following login, xapmcsSpykee will turn on the Video, set the Volume to 50% and turn off the Audio Mic/VOIP. Control buttons are provided to change these states. When Spykee docks it will turn off all LEDs and Video. Undocking will turn the Video on.

Sound bytes can be requested with the set of button in the Sounds group. The Picture button will produce a camera shutter sound and otherwise has no effect. The Love and Call buttons are the two custom one for which audio needs to be initially provided. xapmcsSpykee does not provide the audio associated with these buttons.

Battery status is shown as a percentage number at the bottom of the form. This is a status-only indication. There is still some uncertainty as to the reporting range of the battery from Spykee so this value may not be totally correct. The reported status of the Battery will be ON when the level is above 10% and OFF otherwise. When an OFF status is received then it is appropriate to dock Spykee.

Motion is controlled interactively in two way. The basic speed is set with the slider on the interactive form. When the window focus is changed to the SpykeeVideo form then the four direction arrow keys on the keyboard can be used to request Spykee to move. xapmcsSpykee support eight motion directions where four are achieved with use of the up, down, left, and right keys. The other four are achieved when two keys are held simultaneously such as up/left.

The video feed from Spykee can achieve rates over 15Hz. The current rate is shown on the interactive form. The update rate on the SpykeeVideo form has been limited to 3 Hz to reduce processing requirements. If video display components are discovered that are more efficient then this limit can be removed.

The two button on the lower right of the interactive form are used to record a sequence of commands to Spykee (Record File) and then later replay them (Run File). The Record File button will place the command sequence in a text file selected by user. A segment of one is shown in Figure 7 . This file can be later edited before replaying on Spykee. When recording or playing the command file xapmcsSpykee elevates it process priority to HIGH_PRIORITY to capture and then reproduce the timing aspects of the command being sent. The motion commands are very sensitive to their duration. The Figure 7 sample shows a command in the first column and a delay before next command is given in the second column. A third column can contain a parameter is one is associated with the command. A tab is used to delimit columns.

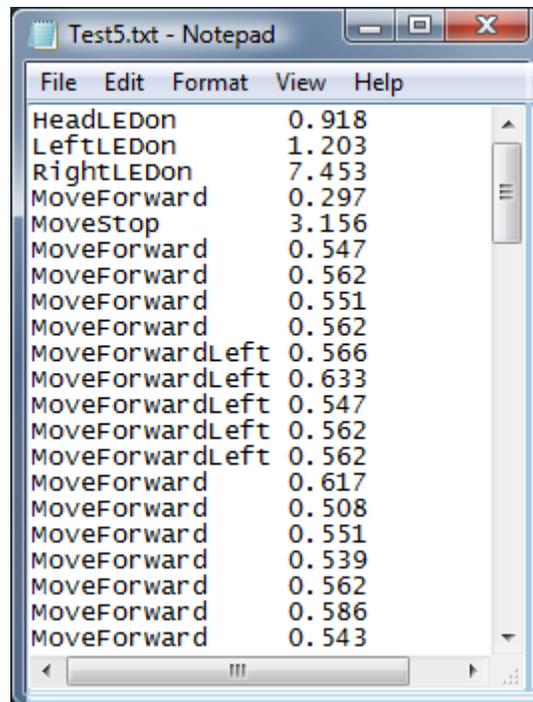


Figure 7 Sample Command File

Experience has shown that Spykee have very poor motor control repeatability. Even though commands are delivered on well timed intervals, the amount of motion changes. Whenever a command file is played a second file is produced that shows the commands provided and the relative time that each was sent to Spykee. This file has the same name as the recorded file, but with the text “_proof” appended. A sample is shown in Figure 8. This can be used to see how well xapmcsSpykee was able to deliver the commands at the intended time. The left column is a running time from the start. The second the second column is the command delivered. The third column is the delta time between commands that will correlate with the requested delay times in the command file.

Running Time	Command	Delta Time
0.086	File 0	G:\VB6\Xa
0.09	HeadLEDon	0.918
1.008	LeftLEDon	1.203
2.25	RightLEDon	7.453
9.781	MoveForward	0.297
9.961	MoveStop	3.156
13.16	MoveForward	0.547
13.664	MoveForward	0.562
14.227	MoveForward	0.551
14.777	MoveForward	0.562
15.34	MoveForwardLeft	0.566
15.906	MoveForwardLeft	0.633
16.543	MoveForwardLeft	0.547
17.09	MoveForwardLeft	0.562
17.652	MoveForwardLeft	0.562
18.215	MoveForward	0.617
18.832	MoveForward	0.508
19.34	MoveForward	0.551
19.891	MoveForward	0.539
20.43	MoveForward	0.562
20.992	MoveForward	0.586
21.582	MoveForward	0.543

Figure 8 Command Proof File Sample

Spykee Automated Control

Automation is achieved with the xAP protocol and use of the XapBSC schema. When using the mcsXap Homeseer plugin then this control is achieved with Homeseer devices with the primary control being the ON/OFF command/status. For example a command to turn on the Left arm LED is sent in the Table 1 and then xapmcsSpykee returns the acknowledge event in Table 2. From a Homeseer environment this would be done by setting the device associated with the Left arm LED to ON (e.g. click on the ON button from the Homeseer Status page) and the event response from xapmcsSpykee would update the device in Homeseer to ON.

Table 1 xapBSC.Cmd Message Example

```

xap-header
{
v=13
hop=1
uid=FF.0080:00
class=xapBSC.Cmd
source=mcs.mcsXap.mcs6
target=mcs.Spykee.MCS5:skypee.LeftLED.Relay
}
output.state.1
{
ID=1B
State=ON
}

```

Table 2 xapBSC.Event Message Example

```

xap-header
{
v=13
hop=1
uid=FF.0065:1B
class=xapBSC.Event
source=mcs.Spykee.MCS5:skypee.LeftLED.Relay
}
output.state
{
State=ON
Text=0
}

```

The set of devices (end points) recognized by xapmcsSpykee is shown in Table 3. Those that have a state associated with them will use the State=XX value of XX for the new state. These are the end points in the list from HeadLED to Connected. Those that have a parameter associated with them will use the Text=YY value of YY for the parameter. These are end points Volume through File.

Table 3 xapmcsSpykee End Points and Parameters

End Point Name	UID	Cmd	State	Text
LoginSequence	01			
MoveForward	02			Duration of motion (seconds)
MoveBackward	03			Duration of motion (seconds)

MoveRight	04			Duration of motion (seconds)
MoveLeft	05			Duration of motion (seconds)
MoveForwardRight	06			Duration of motion (seconds)
MoveBackwardRight	07			Duration of motion (seconds)
MoveForwardLeft	08			Duration of motion (seconds)
MoveBackwardLeft	09			Duration of motion (seconds)
MoveStop	0A			
Movement	0B	No	State=ON when moving	Event/Info =Direction of Movement See *1
SoundCamera	0C			
SoundAlarm	0D			
SoundBomb	0E			
SoundLazer	0F			
SoundLaugh	10			
SoundEngine	11			
SoundRobot	12			
SoundLove	13			
SoundCall	14			
StopDock	15			
Battery	16	No	State=ON when > 10	0 to 99 status
Volume	17			0 to 100 volume magnitude
Speed	18		State=ON when > 75	1 to 99 speed magnitude
File	19			File path of command file to run
HeadLED	1A		ON/OFF	
LeftLED	1B		ON/OFF	
RightLED	1C		ON/OFF	
Docked	1D		ON/OFF	
VOIP	1E		ON/OFF	
Audio	1F		ON/OFF	
Video	20		ON/OFF	Video Rate (Hz)
Turbo	21		ON/OFF	
Connected	22		ON/OFF	

*1 Movement Direction

Direction	Text=
MoveStop	00
MoveForward	01
MoveBackward	02
MoveRight	03
MoveLeft	04

MoveForwardRight	05
	06
MoveBackwardRight	
MoveForwardLeft	07
MoveBackwardLeft	08