

Asterisk Interface

(c) 2013-2015 My Device.
www.mydevice.com.au

This driver provides 2-way feedback and events from the Asterisk PBX system.

Whilst due caution has been taken whilst developing this driver, it may contain bugs and should not be used where it may potentially provide unlawful entry to premises.

Here are a few usage scenarios:

- 1) A guest presses the intercom button, you answer the call and recognize the person. You type *1 on your phone's keypad which triggers an RTI macro to unlock the front gate and turn on the pathway lighting leading to the front door...
- 2) You have an incoming call (external or intercom) and the phones throughout the house are ringing... but you're sitting in the cinema watching a movie at a volume level that isn't going to let you hear it. No problem, because the Asterisk driver has noticed the incoming call and has triggered an RTI macro to pause the movie and fade the cinema lights to 20%.
- 3) You want a doorbell sound to play throughout the house when the intercom is rang. Use the Asterisk driver to detect the incoming call, which triggers a macro to route a doorbell sound to the whole house audio system (Sonos, AD8x etc). Once you pickup a phone extension the driver detects the call has been answers and triggers an RTI macro to mute the doorbell sound.

These are only a few ideas that this powerful driver will allow you to create.

Note:

Please report any bugs found to bugs@mydevice.com.au. Include driver version number and steps to reproduce the issue where possible.

Asterisk Configuration

In this section I will describe two ways to configure the Asterisk setup, via the FreePBX web interface and directly by modifying Asterisk config files.

FreePBX

Login to the FreePBX web interface.

The Asterisk Manager Interface is no longer included as standard, it must be added manually.

Go to Admin -> Module Admin.

Click the "Check Online" button and wait for it to check for updates/module availability.

Scroll down to the "Settings" section and expand the "Asterisk Manager Users" module.

If not already installed/updated, install it now by selecting the Install action.

At the bottom of the page select the "Process" button. This will install the module. Make sure you hit the "Apply Config" button at the top of the page (it'll be in red and hard to miss).

We can now grant the RTI driver access to the system.

Go to the Settings -> Asterisk Manage Users menu.

Add a new Manager.

Manager name: rti - this must match the one set in the driver config.

Manager secret: password - this must match the one set in the driver config.

Deny: 0.0.0.0/0.0.0.0

Permit: <XP IP Address>/255.255.255.0

Set Read permissions for "call" and "dtmf".

Do not give the driver Write permission for anything, that way it can't destroy your system.

Submit changes and apply.

Make sure you set the Permit value correctly, otherwise your XP will be refused access.
Your system is now ready for testing.

Manager: rti

[Delete Manager rti](#)

[Edit Manager](#)

Manager name:

Manager secret:

Deny:

Permit:

[Rights](#)

	Read	Write
system	<input type="checkbox"/>	<input type="checkbox"/>
call	<input checked="" type="checkbox"/>	<input type="checkbox"/>
log	<input type="checkbox"/>	<input type="checkbox"/>
verbose	<input type="checkbox"/>	<input type="checkbox"/>
command	<input type="checkbox"/>	<input type="checkbox"/>
agent	<input type="checkbox"/>	<input type="checkbox"/>
user	<input type="checkbox"/>	<input type="checkbox"/>
config	<input type="checkbox"/>	<input type="checkbox"/>
dtmf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
reporting	<input type="checkbox"/>	<input type="checkbox"/>
cdr	<input type="checkbox"/>	<input type="checkbox"/>
dialplan	<input type="checkbox"/>	<input type="checkbox"/>
originate	<input type="checkbox"/>	<input type="checkbox"/>
message	<input type="checkbox"/>	<input type="checkbox"/>
ALL	<input type="checkbox"/>	<input type="checkbox"/>

Asterisk

For the more adventurous you can modify the config files directly.

First, make sure you have added the AMI module.

Then using your favorite editor open the `/etc/asterisk/manager.conf` file.

Make sure the general settings are as follows:

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0
```

Close and save the file.

Next, edit the `/etc/asterisk/manager_additional.conf` file.

```
[rti]
secret = <password>
deny=0.0.0.0/0.0.0.0
permit=<XP IP Address>/255.255.255.0
read = call,dtmf
write =
```

Substitute `<password>` and `<XP IP Address>` for the correct values.

Exit and save the file.

Do not give the driver Write permission for anything, that way it can't destroy your system.

Connect to the Asterisk CLI and get it to reload:

```
root@raspbx:~# asterisk -r
raspbx*CLI> core reload
```

Your system is now ready for testing.

Driver Configuration

Connection Settings

Asterisk IP Address – the IP address of your Asterisk server. Make sure this is a static IP.

Asterisk AMI Port – this is the port the AMI module listens for commands. Default is port 5038.

Manager Name – username (no spaces allowed). This must match the value set in Asterisk.

Manager Secret – password for the username above. This must match the value set in Asterisk.

Events

Number of DTMF Events – this is the number of touch-tone button press events you wish to monitor.

Number of Dial Events – this is the number of connections you wish to monitor.

DTMF Event

Name – purely for your reference.

Extension – enter the extension(s) that can trigger the event. You can use a single extension, a range, or comma separated items. For example: 100-105,110,201-205 would result in an event triggering if the source was extension 100,101,102,103,104,105,110,201,202,203,204 or 205.

Trigger Code – this is the DTMF sequence that must be pressed to trigger the event. Valid codes are 0-9 and *. You cannot use #, as this key is used to clear the buffer if a mistake is made.

Dial Event

Name – purely for your reference

Extension - enter the extension(s) that can trigger the event. You can use a single extension, a range, or comma separated items. For example: 100-105,110,201-205 would result in an event triggering if the source was extension 100,101,102,103,104,105,110,201,202,203,204 or 205.

Summary

With only two kinds of events you can create a large number of scenarios. If you want to provide the phone user a method to perform a task (open a lock, turn lights on etc) use a DTMF Event. If you want to detect an incoming call is starting/ending/answered then use a Dial Event.

Trouble Shooting

One of the most important things to watch out for is setting a trigger code that matches a dialplan.

Imagine a setup where the dialplan is set to "1.." which allows any number from 100 to 199 to be dialed.

If you were to then use a trigger code of 123, the dialplan would consume the button presses having mistook it for a local extension.

Also watch out for reusing an existing * feature code. For example *69 is the last received call information, so you can't use this as a trigger code.

Security

The driver determines which side of a conversation initiated the call. This means someone can't use their mobile phone to send the DTMF tones from the external intercom down the wire and fool the driver into triggering a macro.

Make sure that you don't allow the intercom extension to be allowed to trigger an event.

Imagine I have a home with extensions 101,102,103,104,105 and the intercom is extension 106.

I would create a DTMF Event that allows extensions 101-105 to be able to access trigger code *123. As 106 is not in this list, any DTMF coming from the intercom side of the conversation is ignored.

If the security permissions are set incorrectly it is possible for a remote device to damage your Asterisk setup, make calls or execute commands at root level on your hardware. Never grant more privileges than required.

All commands are sent across the network in plain text, so this driver should only be used on a trusted network.

Examples

Gate Release

DTMF Event – this will generate an event when *321 is typed from extensions 101,102,103,104 or 105 during a call.

Name: Open Gate

Extension: 101-105

Trigger Code: *321

Incoming Intercom Call

Dial Event – this will generate an event when a call is detected originating from extension 100.

Name: Incoming Call

Extension: 100