ALIEN TECHNOLOGY®

ARCA

Alien RFID Custom Application

Release Notes

January, 2016

All Readers

Alien Technology®

ARCA

Release Notes

January, 2016

This details the release of the ARCA (Alien RFID Custom Application) kit, version 2016_01. The following briefly describes the changes to the the kit since the last firmware release.

Refer to the "Alien On-Reader Programming Overview, ARCA.pdf", revision B or later, for complete details.

New Features

alienconnection: Dropped Connections are Automatically Reconnected

The alienconnection API now can automatically detect when the TCP connection to the reader is broken, and attempt to reconnect to the reader automatically.

alienreader: Setting reader.gpio Can Now Take an Integer Value

The "reader.gpio=" setter method now automatically cases the gpio value to a string for you, so you can now use reader.gpio=5 instead of reader.gpio="5".

alienreader: Looks in Local Directory First for a readermethods.dat File

When alienreader starts up, it looks for a readermethods.dat file containing all of the reader commands that is should know about. This version of the alienreader API now looks first in the local directory for readermethods.dat, and then looks in the folder where the API files are.

By using a custom version of readermethods.dat for each application, you can strip out any references to reader commands that you aren't using, and this can significantly speed up the launch speed and general performance of your app, by simplifying the complexity of the alienreader class.

alientag: New Access to All Tag Fields in the TagList Data

The alientag API could parse text-formatted taglist data and stuff instance variables with the data that it parsed. If there was data that didn't correspond with an instance variable, it was lost. This new alientag API exposes a hash named "field", which contains each of the tag data fields that were received.

For example, if using a custom taglist format:

```
        reader.taglistcustomformat("Tag:%i, PC:${PCWORD}")
```

Then after parsing a line of tag data, the PC word could be accessed with:

```
        tag.field['pc']
```

The inspect method for alientag now simply prints out the contents of the new field hash, so you'll see all the data on the tag, whether it was successfully copied to an instance variable or not.

Parsing of the reader's taglist is now more lenient as well. Any amount of whitespace before or after the commas that separate the tag fields is now tolerated, and reasonable default values are now supplied to some tag data variables (such as count).

alientag: Time fields were Changed from DateTime to Time Objects

This provides consistency with other time objects elsewhere in the API.

alienconfig: Access config data using symbols instead of strings

You can now avoid using string literals, and speed up your custom application insignificantly, but referencing config keys using a symbol instead of a string.
For example:

```
	ipaddress = config[’reader_address’]
```

can now be replaced by:

```
	ipaddress = config[:reader_address]
```

All Example Scripts were Updated

Notably:

```
ex_tagserver.rb      A new demo mimicking the ALR-F800’s new TagServer feature.
ex_automode.rb:      Uses AutomodeReset instead of trying to setup automode manually.
ex_blink_led.rb:     Added cool-looking "dance" light demo!
ex_ftp_download.rb:  Cleaned up code; now points to an actual path on Alien FTP site.
ex_get_taglist.rb:   Uses AlienTaglist's "filter" and "build_tag_array" instead, and
                     no longer filters out only tags starting with ”0000”.
ex_io.rb:            Cleaned up code significantly.
ex_io_ant.rb:        Queries reader's maxantennas instead of hardcoded "3".
ex_tcp_listener.rb:  Major code cleanup - it actually works reliably now!
ex_command_line_interface.rb:  Cleaned up code.
ex_taglist_with_tcp_notify.rb: Uses AlienTaglist.filter & build_tag_array instead.
ex_taglist_with_udp_notify.rb: Uses AlienTaglist.filter & build_tag_array instead.
```

Also, config.dat was renamed to config.ini (and all references to it), config.fetch() is used (with appropriate default values) when possible, and double-quoted string literals were replaced by single-quoted strings, for faster performance.

Quieted StdOut and StdErr on the Serial Console

Spurious output from running ARCA application was appearing on the serial console, with could interfere with normal reader communications on the serial port. All output from ARCA apps is now directed to /dev/null.

RAI Updated to Support ALR-F800 Readers

RAI Registered Apps are now Kept Alive on the Reader

When a custom application is loaded on a reader, registered as a service, and then started by the reader, it is now “kept alive”. If your ruby app should happen to quit unexpectedly, the reader will automatically restart it.

Fixed Bugs

Prevented an App from Starting Multiple Times Concurrently

When a custom application is loaded, registered, then started as a reader service, subsequent attempts to start the same application/service again will just report that the app is started, without starting a second instance of the app.

RAI Utility Updated to Support Ruby v2.1+

A syntax irregularity present in older versions of Ruby was corrected in Ruby v2.1. This version of RAI is consistent with the new preferred syntax.

Added Missing “require 'rubygems'” Line to RAI

The previous version of RAI was missing a “require” statement that prevented it from running on certain Ruby installations.

alientag: Fixed a Bug Which Prevented Sorting Tags with nils

The comparison between tags wasn’t checking to make sure the tag wasn’t nil first. This was fixed with a simple text in the alientag’s “spaceship” comparison operator, <=>