External Integration with the Alarm Octopus - API

There is an API for interfacing to the board, and therefore to the underlying alarm system. The functionality is quite basic at the moment. Further functionality can be added if needed.

With the initial release it is assumed that most interactions will be to inject functionality into the attached control panel of the alarm system. This document is intended to be read in conjunction with the `Instruction Manual`.

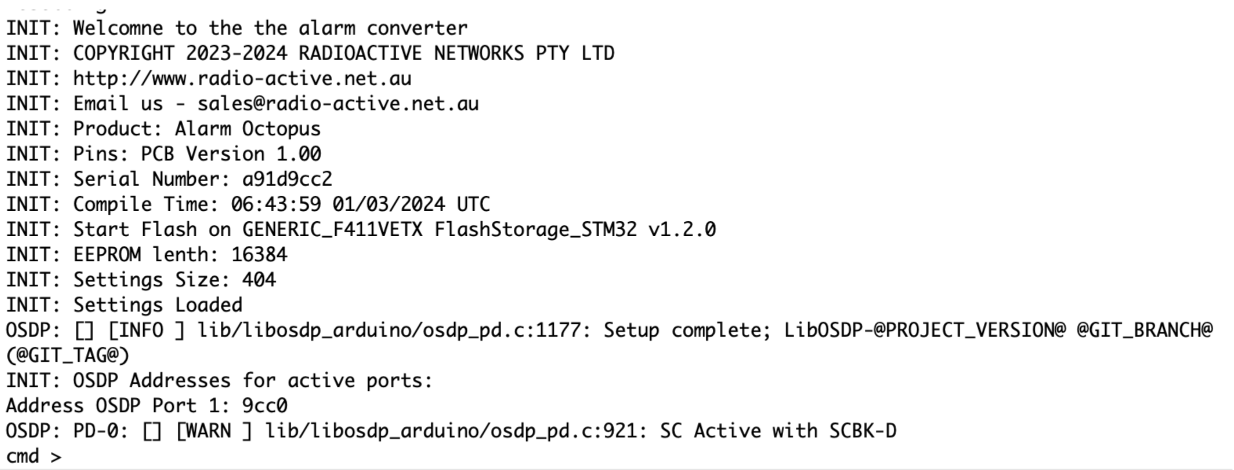
The available functionality is:

- Inject RFID into the alarm system
- Send pre-set RFID cards into the alarm system

Further functionality can be added with a firmware update if required. Other commands from the `instruction manual` may provide useful functionality as well.

Interface

The interface is via a USB connection which emulates a serial port. Com port parametrs are 115200 8N1



```
INIT: Welcomne to the the alarm converter
INIT: COPYRIGHT 2023-2024 RADIOACTIVE NETWORKS PTY LTD
INIT: http://www.radio-active.net.au
INIT: Email us - sales@radio-active.net.au
INIT: Product: Alarm Octopus
INIT: Pins: PCB Version 1.00
INIT: Serial Number: a91d9cc2
INIT: Compile Time: 06:43:59 01/03/2024 UTC
INIT: Start Flash on GENERIC_F411VETX FlashStorage_STM32 v1.2.0
INIT: EEPROM lenth: 16384
INIT: Settings Size: 404
INIT: Settings Loaded
OSDP: [] [INFO ] lib/libosdp_arduino/osdp_pd.c:1177: Setup complete; LibOSDP-@PROJECT_VERSION@ @GIT_BRANCH@
(@GIT_TAG@)
INIT: OSDP Addresses for active ports:
Address OSDP Port 1: 9cc0
OSDP: PD-0: [] [WARN ] lib/libosdp_arduino/osdp_pd.c:921: SC Active with SCBK-D
cmd >
```

On power up, the command console will be locked. It can be unlocked with the `password` command, followed by the appropriate password. The password can be removed with the command `password set blank`. Once the console is unlocked, it will lock again after an appropriate delay, and will require entering the password again. Obviously, if the password is set to blank, the console will not auto-lock.

The console can be locked with the use of the `password` command by itself.

The console will indicate that it is ready for commands by displaying `cmd >`. This will be the case regardless of if the console is locked or unlocked. If an event comes in causing information to be displayed, the prompt and any entered characters will be removed, the information displayed, and the prompt and typed characters will be restored.

Commands can generally be sent blind, provided the console is unlocked, or the command to unlock is included in the sent characters.

Commands

Injecting RFID Cards

RFID Cards can be injected onto a specific OSDP port. The code will be sent as-is including any parity errors. The format of the command is:

`inject [OSDP Port] [RFID CARD]`

The OSDP Port would generally be between 1 and 8. The RFID Card number will be either eight or ten characters long, consistingt of hex digits.

`inject 3 1234ABCD56`

Sendind Pre-Configured Cards

The `fake` command allows you to store a card to be sent when the button attached to a wiegand port is pressed. This functionality can used programatically as well. The `card` command has been created to emulate that button being pressed. It can be used on any of the Wiegand ports. The `instruction manual` contains more information on the `fake` command

`card [port]`

`card 3`

Monitoring for Events

Various events will be displayed on the console. Since some of this information may be sensitve, the `protect` command can be used to stop most information from being displayed unless logged in. Setting `protect 0` will allow all events to be monitored even when the console is locked. Setting `protect 1` will require a password to be entered for most information to be displayed.