Vector Documentation

Because crashing Galaxy's computer is a tradition

Randall Maas & the Project Victor contributors

Table of contents

1. Vector Resources

This is a wiki for gathering and sharing information about Anki Vector

PDF version

See also this other Wiki by Xanathon

1.1 Games and Things you can do with Vector

- A "cheat sheet" of the things you can say to Vector by Samuel Ward. (pdf)

- Another Vector command list

- A Guide to the Vectorverse by Stephan Otter (@StephanOtter) and Steven Coblentz (@SteveCoblentz). This is a document on the interactions and games you can play with Vector.

1.2 Personalizing Vector

This is for notes on how to customize or personalize Vector. You may have to consult the How-To's below.

See also: the forums

1.3 Troubleshooting

- What Do Vector's Back Lights Mean?

- How to check for software problems

- The big long list of error codes is Appendix D of the Technical Reference Manual, and in TBD

See also the troubleshooting at DDL's site:

- Troubleshooting Vector's Connection

- How Do I Find Vector's Serial Number?

- Why does Vector need a 2.4 GHz network?

- Vector does not understand me: Troubleshooting Speech Recognition

- What Do Vector's Back Lights Mean?

- Why does Vector show an error?

- Troubleshooting charging issues

- Why is there no sound?

1.4 Service Guide

Collected notes on repairing or modifying Vector.

- Assembly and exploded view diagrams. I am a sucker for exploded diagrams and drawings.

- How to update software

- How to clean wheels/sensors

- Where to get parts -- treads, etc.

- How to replace the battery?

- Boards?

1.5 How-Tos

These try to tell you how to accomplish particular tasks.

Some highlights:

- Using GDB to trace function calls

- How to create a soundbank

- Dauler sells stickers with the marker symbols preprinted at 3D Designs by Dauler

1.6 Developer documentation

These are reference documentation for programming tools to use Vector. Some of them are for the remote-access SDK's.

1.6.1 Technical Reference Manual

- Details on how Vector *works*

- Main architecture of the design (not necessarily the code though) and how it works

- File system structure, files, formats and contents

- Communication protocols

1.6.2 Programmers Guides and Examples

The main PC/Mobile SDKs are:

- Python Communication SDK: Vector - Python SDK

- C# Communication SDK: Anki.Vector.SDK

- See the examples with each of the packages

See the SDKs page for more repositories and other resources.

1.6.3 WebViz and Console Variables

"Pure" Developer builds of Vector software contain an HTTP API and webserver. This shows what it is, how to use it, and how it works.

1.6.4 The Communication Protocols

The communication protocols Vector uses to talk to the cloud. (Several of the protocols were specified with gRPC and Protobuf. The information here was reconstructed from binaries, WebViz, logs and other sources. It is hoped to allow reconstruction of significant portions of the Protobuf specification. If the source protobuf specification files do become available later, these can be used to comment them.)

1.6.5 Vector Enhancement Proposals

These are proposals for changes -- enhancements -- to the modules on Vector.

Some highlights:

- An overview of the overall proposal process.
- VEP1. Update-engine changes
- VEP2. Packagement for modules on Vector

1.7 Historical Bots

This is a place for info about robots that were part of Vector's evolution, but are products in their own right.

- DVT1-4 bot info
- Whiskey info
- Bingo info
- etc

1.8 Stuff to help collaborate

1.8.1 Guidance

These provide tips/suggestions on style, naming. They are related to the "How-to's" but they don't walk you thru to a specific goal. For instance, some might describe how to a do a particular style of design or implement a kind of behavior.

Examples:

- Recommendations for sound event names
- Steps that a design/process can do to meet the spec

Good title:

Bad title:

Writing guide.

- Other writer guides - Show how to do something in general, like a tutorial

- Document and show off how you built one of your projects Background: This is a note I made for myself to guide me on the right tone, help with consistency and give me some direction.

1.8.2 Templates

To help get started with creating a new entry, the document-templates folder includes some start files that can be used as templates when creating new documents:

- A template for how to documents

- A generic template for other files

2. Glossary

Abbreviations:

Abbreviation / Acronym	Phrase
jwt	JSON web token
DVT	Design Validation Build, used for testing the robustness of the design, emissions testing, development and so forth.
PII	personally identifying information
PVT	Production Validation Test build; Essentially practice builds, as good as production (or close).
sts	security token service

Terms:

Term	Description
behavior	A structure on Vector to represent and manage a potentially complex task that might involve animations, changing his emotions, path planing, driving, and so on. It is also used on the EscapePod as a catchall for utterances and how they are made into an intent.
intent	An intent is a structure with an internal code that is used to represent the how to respond to the phrases spoken by a person. It may represent the action requested, an answer to a query, or an action that emotionally responds to what was said.
JSON web token	https://en.wikipedia.org/wiki/JSON_Web_Token
property name	The name of a key in the structure; also called a field.
security token service	https://en.wikipedia.org/wiki/Security_token_service
structure	A table of property names (aka field, or key) and the value associated with it.
utterance	What a person said, and in the context of this article, the transcription to text of what the person said.

3. Boards

3.1 Vector boards

3.1.1 Head Boards
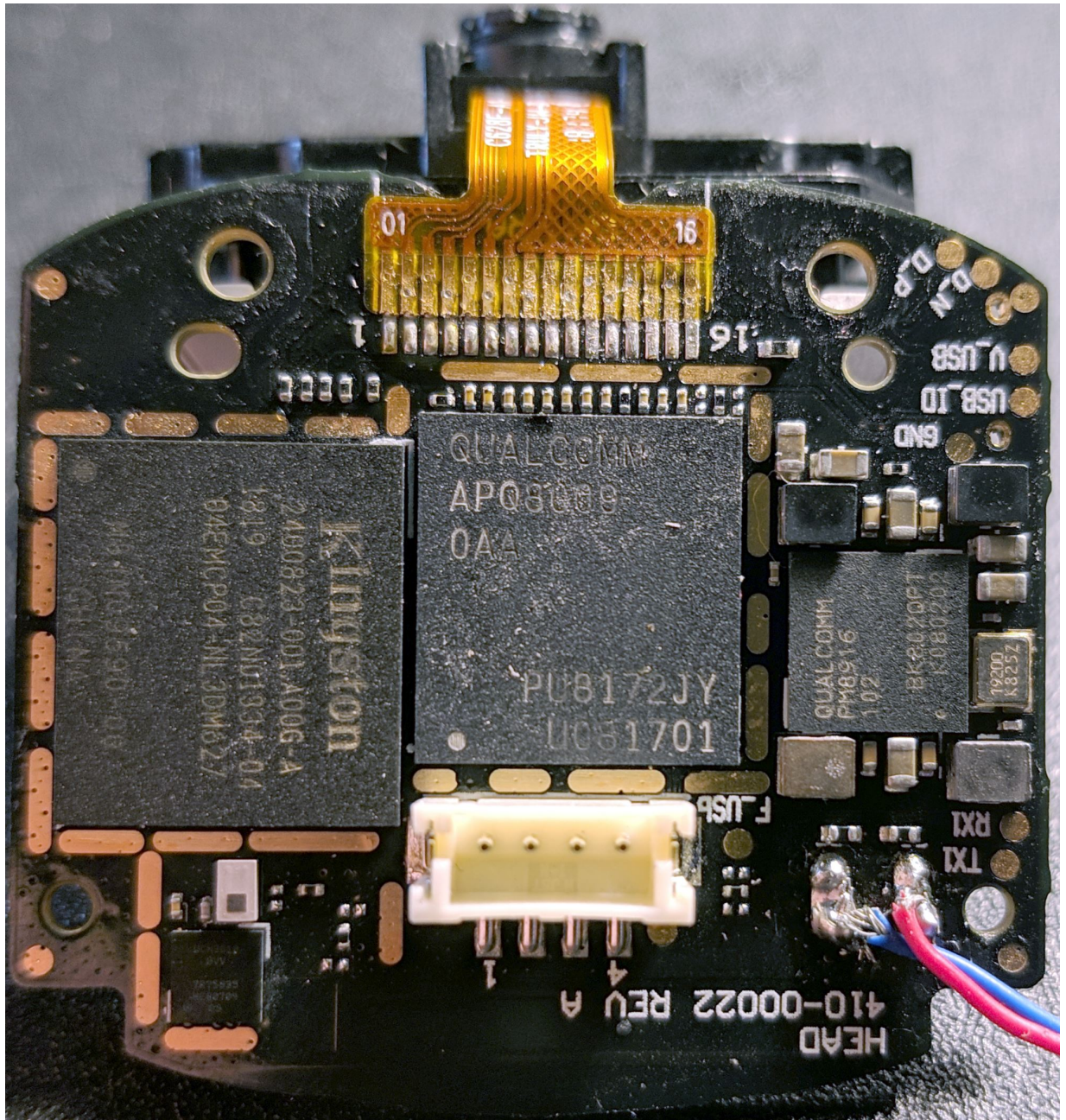
Figure: Vector head-board top

3.1.2 Body boards

These drive the motors, talk with the time of flight sensor, pull the microphones sounds from the head board, drive its LEDs, etc.

STM-5
94V-0

Figure: Vector body-board top
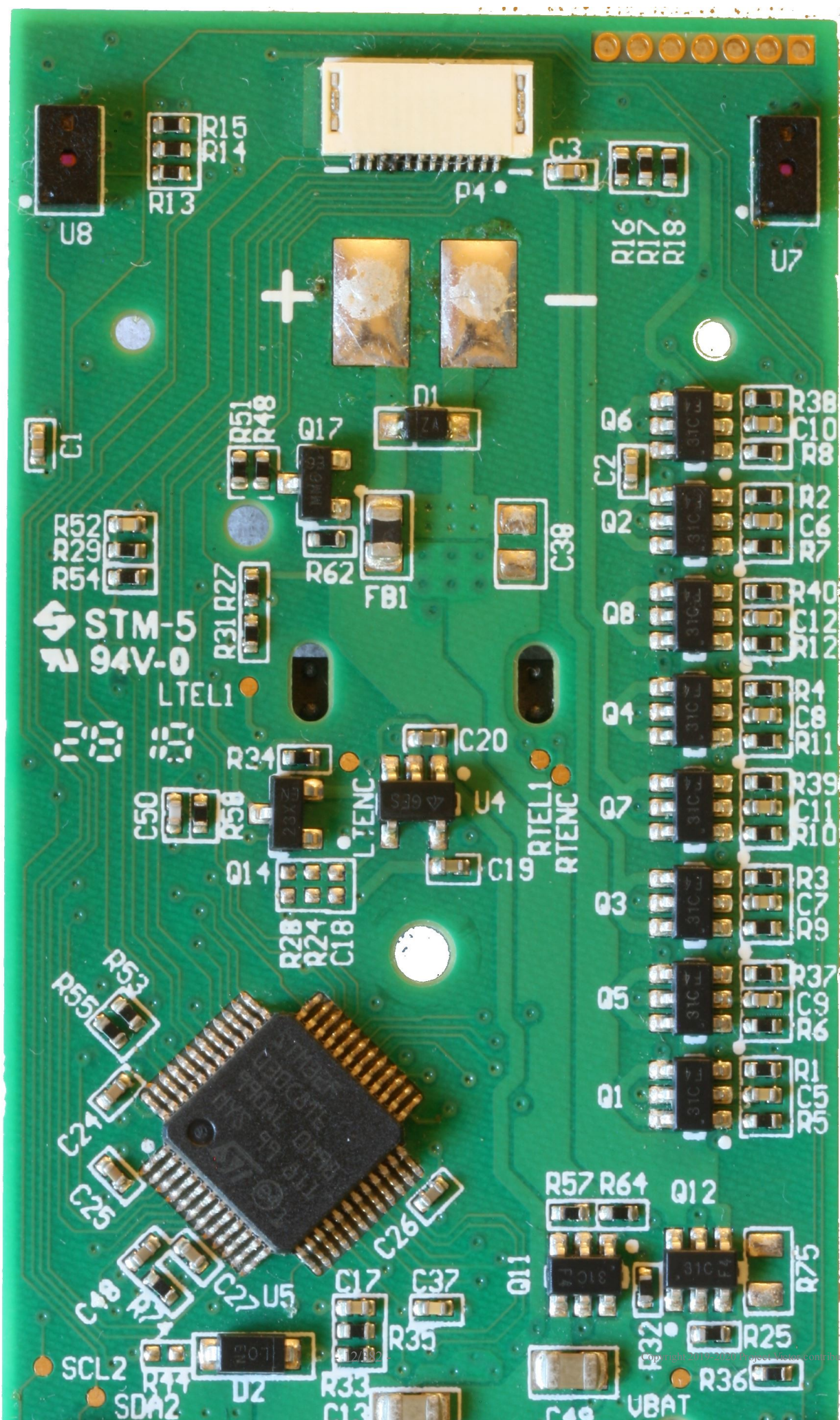
Figure: Vector body-board top
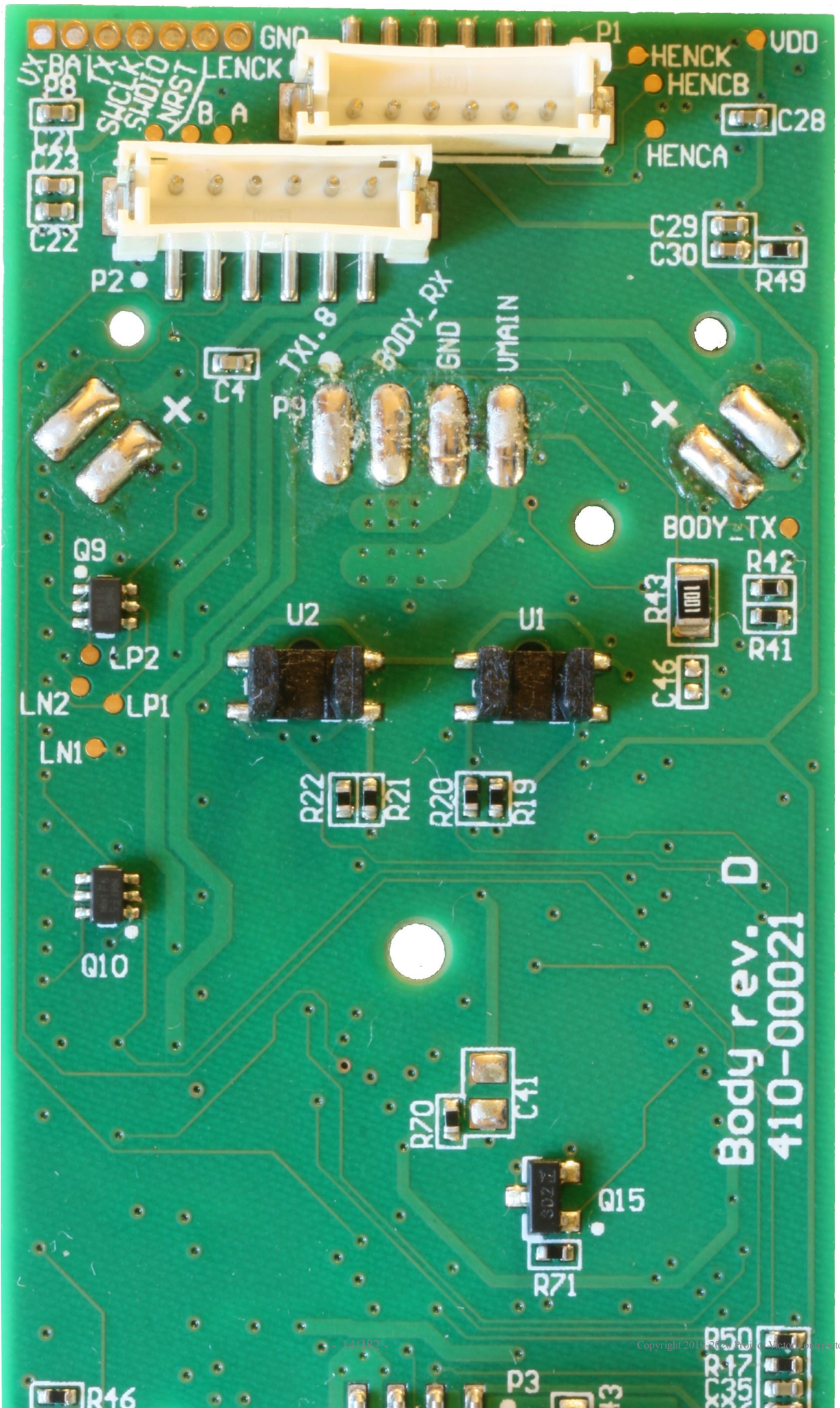
Figure: Vector body-board bottom

Some initial body-board wiring schematics (rather than function-focused) and Feel free to edit the EasyEDA file.

Body Board Firmware

Each revision of the body board has a bootloader specific to that revision; the boot loaders holds a revision code at 0x08000010. (In some cases, this revision code may be the only change.) This revision code (or number) is used by the application firmware (held in syscon.dfu) to know which board it is running on, and make small changes to accomodate the differences.

Body-Board Type	Revision Code	Description and Notes
DVT1	1	
DVT2	2	
DVT3	3	
DVT4	4	This board has significant changes from the DVT3. DVT4 also corrected a regulator problem (although that may be on the headboard.)
PVT	5	
Production	6	The LED clock and data lines have moved, adding a line to manage power for the head or backpack.
Whiskey	7	Version 1.4 of the syscon does not work correctly with this board; this suggests that the Whiskey body board was manufactured after the 1.4 firmware was made.

4. Contributing

4.1 Contributing

Thanks for being interested in contributing! We're so glad you want to help!

We want contributing to Project Victor to be fun, enjoyable, and educational for all. We love receiving contributions from our community, all contributions are welcome.

There are many ways to contribute. You can also help us by:

- Answering questions people have have in the forums
- Helping us build and design our website
- Cleaning up our existing documentation, polishing it, fixing our spelling or grammar mistakes, and so on
- Create new documentation
- Create an example of some changes / fixes/ hacks
- Creating blog posts, and tutorials about one of Vector's many features
- Reviewing submissions
- Contributing bits that can be incorporated into this or related projects.

Below you will find tips on how to get the most out of your contributing experience, including GitHub management tips, setup instructions for docs and code contributions, and more.

4.1.1 Not sure how to start contributing?

If you are worried or don't know where to start, you can reach out with questions to anyone from the Project Victor team on

- **Official Anki developer forums**
- **Anki robots Discord chat**
- **GitHub Discussions** is directly integrated with the repository. You can use this to ask for help or share ideas related to improving the documentation or deploying it.

4.1.2 Pair programming

Other projects offer free pair programming sessions to the community. I think that might be a neat idea, if there's something others would like to work on together......

4.1.3 How to start contributing and our code of conduct

Below you'll find guides on our community, code of conduct, and how to get started contributing:

- Code of Conduct: Read about what we expect from everyone participating to make it the most friendly and welcoming community.

- Style Guide: The art of contributing, a.k.a. the detailed requirements that will make it more likely your contribution is accepted with minimal changes.

  By participating in this project, you agree to abide by our Code of Conduct. We expect all contributors to follow the Code of Conduct and to treat fellow humans with respect.

4.1.4 Important Resources

The important documents and links are on the front page of the wiki.

4.1.5 Improving Documentation

If you have a suggestion for the documentation, I would recommend that you take a stab at making the changes to the documentation. Simple changes can often be made without a sophisticated pull release.

For large fixes, please build and test the documentation before submitting the pull-request to be sure you haven't accidentally introduced any layout or formatting issues.

Templates

To help get started with creating a new entry, the document-templates folder includes some start files that can be used as templates when creating new documents:

- A template for how to documents
- A generic template for other files

How to Create the HTML and PDF files

The source documentation text files can be found documents directory. The built out files will be placed within a [site] directory.

First, install the documentation tools:

```
pip3 install mkdocs-material
pip3 install mkdocs-localsearch
```

Then you can build the html site simply by:

```
mkdocs build
```

Building a PDF file as well

You can also build the PDF. First install the tools:

```
pip3 install mkdocs-with-pdf
```

There is some further installation, see the following link for more details: https://pypi.org/project/mkdocs-with-pdf/

Rename the "mkdocs.yml" file Then rename "mkdocs-pdf.yml" to "mkdocs.yml"

To build is the same as before

```
mkdocs build
```

Whitespace Cleanup

Don't mix code or documentation changes with whitespace cleanup! If you are fixing whitespace, include those changes separately from your code changes. If your request is unreadable due to whitespace changes, it will be rejected.

Please submit whitespace cleanups in a separate pull request.

4.1.6 Reorganizing Filesystem / directory tree

Don't mix rearranging the location and names of files with code or documentation changes! If you are rearranging the file system, please include those changes separately from your code changes.

Please submit file system changes in a separate pull request.

4.1.7 Pull Request Process

Please see the pull requests page for the process of submitting your changes to the prooject and incorporating feedback. are happy with your changes first!

Credits: This page was adapted from an EmbeddedArtistry template

4.2 Contributor Covenant Code of Conduct

4.2.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

4.2.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people

- Being respectful of differing opinions, viewpoints, and experiences

- Giving and gracefully accepting constructive feedback

- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience

- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind

- Trolling, insulting or derogatory comments, and personal or political attacks

- Public or private harassment

- Publishing others' private information, such as a physical or email address, without their explicit permission

- Other conduct which could reasonably be considered inappropriate in a professional setting

4.2.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

4.2.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

4.2.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at [INSERT CONTACT METHOD]. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

4.2.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

4.2.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at https://www.contributor-covenant.org/faq. Translations are available at https://www.contributor-covenant.org/translations.

4.3 How to File an Issue

The GitHub issue tracker is the preferred channel for bug reports, documentation, feature requests and submitting pull requests.

To resolve your issue, please select the appropriate category:

- Documentation

- Feature Requests

Please do not use the issue tracker for personal support requests. The discourse and forums are the better places to request help.

4.3.1 Labelling the issues

Issue labels are a tool in GitHub that are used to group issues into one or more categories. Labeling issues helps by identifying:

- good issues for new contributors to work on

- reported and confirmed bugs

- feature requests

- duplicate issues

- issues that are stalled or blocked

- the status of an open issue

- the topic or subject matter of the issue

When an issue is created -- and later examined -- this is a good time to check that its label is sensible, and to add an other labels that are helpful -- and to remove labels that do not apply.

You can see a list of this project's labels (and their descriptions).

4.3.2 Special Note on Issues

If an issue is affecting you, start at the top of this list and complete as many tasks on the list as you can:

1. If there is an issue and you can add more detail, write a comment describing how the problem is affecting you, OR if you can, write up a work-around or improvement for the issue

2. If there *is not* an issue, write the most complete description of what's happening

3. Offer to help fix the issue (and it is totally expected that you ask for help; open-source maintainers want to help contributors)

4. Deliver a well-crafted, tested PR

Credits: this was adapted in part from the Gatsbj.js project

4.4 Pull Requests

This document describes what you needed to know about the pull request process.

A pull request is how you submit your changes to the project. Before you make any changes, please read the contributing page for information regarding contributions to project overall. This will help you in making your changes fit within the project and its style; as well as the steps you must do before creating a pull-request.

4.4.1 What is a Pull Request (PR)?

As described above, a pull request is how you submit changes to this project. It is a *request* that the project pull in your changes. Here's how the folks at GitHub define a pull request:

> Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.

The pull request allows others to review the changes, test them, and provide feedback -- including requests to makes to the changees, so that they better fit into the project.

4.4.2 Pull Request Process

Once you have completed the changes on your local development environment, tested them, and so fprth, the next steps is to create a pull request. Be sure to check the contributing guide for additional steps and tips to ensure that your changes will fit with the project.

> When you are ready to generate a pull request, either for preliminary review, or for consideration of merging into the project you must first push your local topic branch back up to GitHub:

```
git push origin newfeature
```

> Once you've committed and pushed all of your changes to GitHub, go to the page for your fork on GitHub, select your development branch, and click the pull request button. If you need to make any adjustments to your pull request, just push the updates to your branch. Your pull request will automatically track the changes on your development branch and update.

1. Ensure any install or build dependencies are removed before the end of the layer when doing a build.
2. You may merge the Pull Request in once you have the sign-off of two other developers, or if you do not have permission to do that, you may request the second reviewer to merge it for you.

Review Process

After a pull request has been sent to the repository, the team and community may suggest modifications to the changes you have submitted.

Many pull requests are likely to open for several days, until the core team can approve them in Github. In some cases, multiple people will have the chance to review/comment.

Please check your pull request for comments, feedback, and suggested changes:

- Review the suggested changes using the "View changes" button.

- Commit the suggestions.

- Discuss suggestions to ask questions about the suggested changes.

- Incoporate the suggestions to your changes

Addressing Feedback

Once a PR has been submitted, your changes will be reviewed and constructive feedback may be provided. Feedback isn't meant as an attack, but to help make sure the highest-quality workmanship makes it into our project. Changes will be approved once required feedback has been addressed.

If a maintainer asks you to "rebase" your PR, they're saying that a lot of files has changed, and that you need to update your fork so it's easier to merge.

To update your forked repository, follow these steps:

```
# Fetch upstream master and merge with your repo's master branch
git fetch upstream
git checkout master
git merge upstream/master

# If there were any new commits, rebase your development branch
git checkout newfeature
git rebase master
```

If too much code has changed for git to automatically apply your branches changes to the new master, you will need to manually resolve the merge conflicts yourself.

Once your new branch has no conflicts and works correctly, you can override your old branch using this command:

```
git push -f
```

Note that this will overwrite the old branch on the server, so make sure you are happy with your changes first!

4.4.3 Additional resources

- Creating a pull request from GitHub

- Configuring a remote for a fork

- Which remote URL should I use?

- Git Branching and Merging

- Feature Branching and Workflows

- Resolving merge conflicts

Credits: This page was adapted from an EmbeddedArtistry template and adapted from the Gatsbj.js project

4.5 Style Guide

- The title of the document should use a `#` (in Markdown). Only a single title (`#`) should be used.

- The heading levels should start with `##` (in Markdown) and grow in order

4.5.1 Links

Links ought to be relative instead of absolute when linking to documents. That is to say, the should not include the full domain. For example `/documents/some-reference/` instead of `https://randym32.github.io/Anki.Vector.Documentation/ some-reference/`

5. Customization

5.1 Body modifications

- Ikkez sells cute ears, deedly-boppers, and treads at his Etsy site

- Redwish's review of tread by Ikkez (original post)

- Dauler sells ear, horns, treads, stickers, and other fanciful mods, and 3D STL's at 3D Designs by Dauler and Etsy

- Dauler sells treads at 3D Designs by Dauler

- Cat ears for Vector, as 3D files by "misconduct"

5.2 Customization

This is for notes on how to customize or personalize Vector.

See also: the forums

It might be thru configuring the software and files:

- Sounds

- Body movements

- Eyes

- Colors?

- PNGs on face

- Backpack lights

- Cube lights

- Other custom animations

Or it might be physical changes, and tweaks that are distinctive and identify

.. Link to a showcase ..?

5.2.1 Customizing animations

- animation of eyes

- body movements

- boot animation

People would love tools to gen the animation file… but I suspect that may be hard. The presentations made it sound like it was a lot of Maya rigging and plugins for the export.. but since Maya is expensive, and hard..

Maybe a Unity model tool could be made with a rigged model of Vector? and such for I suspect the value vs effort isn't there for such a specialized area, but who knows?

Tools for generating animation files.

What about mixing-matching existing animations, and adjusting them slightly?

Boot animation draft

Vector shows a boot animation at startup. This is located in /anki/data/assets/cozmo_resources/config/engine/animations/ boot_anim.raw and it can be swapped out easily.

Digital Dream Labs has made a Python script which makes it easy to turn GIFs into animations very easily.

- A working installation of python with the Pillow package installed.

- An animated .gif with a resolution of 184x96 pixels

- The script gif_to_raw.py to convert the .gif to a raw image.

- Convert the .gif to a raw image: `python gif_to_raw.py bootscreen.gif` This will create a new file bootscreen.gif.raw

- Mount the filesystem for writing. Here we'll do that from the host system: `ssh root@192.168.1.110 "mount -o remount,rw /"`

- Use scp to copy the file in to place: `scp bootscreen.gif.raw root@192.168.1.110:/anki/data/assets/ cozmo_resources/config/engine/animations/boot_anim.raw`

- Reboot Vector from the host system: `ssh root@192.168.1.110 "/sbin/reboot"`

5.2.2 Sounds

5.2.3 Behavior tree crafting

There are many json files in /anki/data/assets/cozmo_resources/config/engine/behaviorComponent/. Maybe have some examples of edits of those?

5.3 How to enable Cozmo-like animations for being on his side, and flipping down from being on his back

This is a note to describe how to enable (potentially) Cozmo-like animations for being on his side, and flipping down from being on his back.

Note: I don't know that these changes will make Vector more interesting

All of the files that we'll modify are in: `/anki/data/assets/cozmo_resources/assets/animationGroups/ReactToCliff`

These animation group files change which animations are used to use more of Cozmo's variety of animations.

Note: in general, not all animation groups that Cozmo uses are used by Vector. In this case, they are.

5.3.1 Preparation

You'll have to know how to SSH in, make the file system modifiable and edit a file. To make the file system modifiable:

```
mount -o rw,remount /
```

Make backups of the animation group files

All of the files that we'll modify are in: `/anki/data/assets/cozmo_resources/assets/animationGroups/ReactToCliff`

I recommend making a back up of the following files:

- ag_reacttocliff_stuckleftside_01.json

- ag_reacttocliff_stuckrightside_01.json

- ag_reacttocliff_turtleroll_01.json

- ag_reacttocliff_turtlerollfail_01.json

You can do this by copying it to a back up name. For instance:

```
cd /anki/data/assets/cozmo_resources/assets/animationGroups/ReactToCliff
cp ag_reacttocliff_stuckleftside_01.json ag_reacttocliff_stuckleftside_01.bak
cp ag_reacttocliff_stuckrightside_01.json ag_reacttocliff_stuckrightside_01.bak
cp ag_reacttocliff_turtleroll_01.json ag_reacttocliff_turtleroll_01.bak
cp ag_reacttocliff_turtlerollfail_01.json ag_reacttocliff_turtlerollfail_01.bak
```

If later on you want to go back to the orignal for any of these, you can reverse this to restore it. For example:

```
cp ag_reacttocliff_stuckleftside_01.bak ag_reacttocliff_stuckleftside_01.json
```

5.3.2 Make the files writeable

```
chmod +x ag_reacttocliff_stuckleftside_01.json
chmod +x ag_reacttocliff_stuckrightside_01.json
chmod +x ag_reacttocliff_turtleroll_01.json
chmod +x ag_reacttocliff_turtlerollfail_01.json
```

5.3.3 Next Copy the replacement files

Copy the replacement files to the that directory. I've attached the files to this note, from Cozmo's APK.

You can copy them with `scp` or other method. I use vi.

- ag_reacttocliff_stuckleftside_01.json

- ag_reacttocliff_stuckrightside_01.json

- ag_reacttocliff_turtleroll_01.json

- ag_reacttocliff_turtlerollfail_01.json

5.3.4 Reboot

Finally you have to restart the vic applications for the updates to load and take effect. This can be done with:

```
systemctl stop anki-robot.target
systemctl start anki-robot.target
```

or a reboot.

5.3.5 A few notes on possible next steps

You can edit a more animation group files and behaviors. I tried to variations on

```
ag_reacttocliff_wheelie_01.json
```

to make it Cozmo use the same animations that Cozmo calls out, but Vector would no longer pop a wheelie for me.

Some of the animations files that these animation groups might not be fully tuned for Vector and his cube's body.. and may need some further tweaking to create the same energetic effect cozmo gives.

5.4 Nose Art Showcase

6. Document templates

6.1 VEP Template

```
---
title: VEP123 - The name of the VEP (only a few words)
summary: An optional description of the proposal, if the title is too short
authors:
    - Author Name
date: 2022-07-10
---
```

(remove the quotes; they are so that the template is readable)

6.1.1 Description of the changes

Motivation: A synopsis of why this should be done -- we don't want complicated goo-gaws for the sake of it.

6.1.2 Some Design decisions

Optional

6.1.3 Documentation

The documentation (if short) or where can the documentation be found

6.1.4 Cavaets

List any limits / warnings about this

6.1.5 Status

Has it been tried? How much? Where?

6.1.6 References

6.1.7 Change history synopsis

Summary of changes to help the reader

Date	Change

6.2 How-to template

{Choose a good title name for the file. It should lead with what it noun or action is, and follow the pattern of other documents in this section}

6.2.1 Preparation

Include a section on the preparation steps

6.2.2 Steps

6.2.3 References and Resources

Optional Include some some links to other resources here.

6.3 Template

{Choose a good title name for the file. It should lead with what it noun or action is, and follow the pattern of other documents in this section}

If a table of parameters (or fields) is needed, the following can be used as a starter:

Name	Type	Units	Value	Description

Table: caption

or

Name	Type	Value	Description

Table: caption

An image can be caption like so:

Some desciption

Figure: The image caption

To refer to a behavior use italic emphasis, and (where possible) link to its description: *EmergencyMode*

6.3.1 References and Resources

Optional Include some some links to other resources here.

7. Escape pod

7.1 Computer setup

7.1.1 On Your Computer

You should have Chrome Installed.

Windows Computers

If you have a Windows computer, you need some software installed for "mdns". If you have iTunes installed, this software is already installed for you and you can skip this step. If not, install bonjour from:

- https://support.apple.com/downloads/bonjour_for_windows

Then, on the command line

```
REG ADD  "HKLM\Software\policies\Microsoft\Windows NT\DNSClient"

REG ADD  "HKLM\Software\policies\Microsoft\Windows NT\DNSClient" /v " EnableMulticast" /t REG_DWORD /d "0" /f
```

Linux

If you have a Linux computer, you may need to make a make change to the '/etc/nsswitch.conf'. *Note: this section is not yet confirmed*

In a command line, open the 'etc/nsswitch.conf' file. It probably will look like this:

```
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.
passwd:         files systemd
group:          files systemd
shadow:         files
gshadow:        files
hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files
protocols:      db files
services:       db files
ethers:         db files
rpc:            db files
netgroup:       nis
```

Notice the 'hosts:' We need to add 'mdns' to the end so that the line looks like:

```
hosts:          files mdns4_minimal [NOTFOUND=return] dns mdns
```

What does this do? Why there are two "mdns" items? The second mdns is needed ot make the ".local" domain work. "The minimal versions [the mdns4_minimaal] will always deny to resolve host names that don't end in .local or addresses that aren't in the range 169.254.x.x"

7.1.2 Replacement for the Mobile App

The mobile application will not work. It expects to talk with the production servers and doesn't know how to work with the EscapePod. Instead, use Vector Explorer by Wayne Venables at:

- https://weekendrobot.com/vectorexplorer/

7.2 Equipment

This is very very important.

You must get very specific USB "chargers" (power supplies) and cables. {And SD card?} We list below the ones that have worked. If you do not, the Raspberry Pi can (and probably will) randomly lock up, crash. The cause will look mysterious.

Why? The escape pod is running software that demands a lot of a Raspberry Pi at times. The Raspberry Pi has requirements higher than standard USB-specification chargers and cables. The Pi will have errors and random crashes if the right ones are not used.

7.2.1 Hardware Configurations

If you are buying hardware

- Buy the Raspberry Pi 4, with the official Charger. Raspberry PI 4 in 2GB, 4GB, and 8GB are all reported to work. (Prices vary, but the 8GB Pi 4, charger, and SD card cost me $81 at Micro center. Another $6 for the optional micro hdmi cable)

- A Canakit charger may be instead (see below for links).

If you already have a Raspberry PI 3B+, this can be used. However be aware:

- The PI3 will be noticeably slower and less responsive

- Make absolutely sure to get very specific power supplies and cables. Power supplies and cables that worked with other software on a Pi 3 may not work here. The EscapePod software has higher demands than raspbian. Without the specific supplies AND cables, there may be power issues: random crashes, lock ups, and confusing error messages.

- Prefer power supplies where the cable is directly built-in (connected) rather than a separate charger and a cable. Like this:

Figure: This is what a Raspberry Pi power supply should look like

Pi3 'official' Chargers (tested with a Pi3)

- https://shop.pimoroni.com/products/raspberry-pi-universal-power-supply

Other chargers tested with Pi3

- RavPower 4 and [Exact name of cable.]

- CanaKit USB-C Raspberry Pi 4 Power Supply: https://www.amazon.com/gp/product/B07TYQRXTK/

Pi4 official USB-C Chargers

- https://shop.pimoroni.com/products/raspberry-pi-official-usb-c-power-supply?variant=29157000085587

7.2.2 Sundry Tips

From the internet: "If your Raspberry Pi 4 is running a little hot, users can get it running cooler simply by positioning it vertically with the GPIO header at the bottom and the power and HDMI ports at the top."

8. Guides

8.1 Cozmo Character Design

Cozmo -- the generation prior to Vector -- paved a way for a lot of the character design. The principles Anki developped for his character apply to Vector as well.

THE ULTIMATE GOAL FOR COZMO IS TO CREATE A REAL, BELIEVABLE ROBOTIC CHARACTER THAT FEELS ALIVE.

Something we have seen over and over again in movies, but never in real life. Cozmo needs to feel alive the same way a pet feels alive, by creating a strong emotional connection with people. Really long term, we want to create a series of characters, with an ecosystem around them and the ability to have the types of stories we only see in movies play out in the real world.

THE TOP PRIORITY, ABOVE ALL ELSE, IS HIGHLIGHTING THE PERSONALITY OF THE CHARACTER. EVERYTHING ELSE BECOMES A TOOL IN SERVICE OF THAT GOAL.

Cozmo is the soul of the product, and where the 'magic' is. Everything else in the experience is in service of making the character feel alive, and emotionally intelligent. Mini-games, UX, game mechanics/structure, story, etc. should all be thought of as tools for creating context for making Cozmo feel more alive with a richer personality. The game is a means for driving engagement / exploration of Cozmo and his boundaries. Cozmo will have a limited ability to understand his world in general, but be extremely smart in specific areas. Our goal is to optimize for his strengths, and avoid his constraints. The goal for any accessories, games, etc. for Cozmo is to channel players' attention towards the things Cozmo understands really well. These are the best opportunities for us to impress and surprise the user with emotions and depth of character and intelligence in a way that only we can.

COZMO IS NOT A GAME, BUT RATHER AN EXPERIENCE, WITH THE INTERACTIVE ELEMENTS THERE TO GUIDE YOU THROUGH THE JOURNEY AND MAGIC OF INTERACTING WITH YOUR ROBOT FRIEND.

Think of Cozmo as your robot pet, with regards to exploring possible interaction, play and responsibilities. Not in how it looks (no fur and whiskers). He's smart, he's emotional, he recognizes you, he has a sense of humor, he wants to interact with you — these are the types of feelings we want to naturally draw out of users.

FOCUS WHENEVER POSSIBLE SHOULD BE ON COZMO, NOT THE APP / PHONE.

The app, supporting sound, UI, etc. are there to help gently guide the user through the experience of playing with Cozmo and exploring his boundaries. But whenever possible, focus should be on discovering the physical and emotional sides of the experience.

Figure: The Cozmo Brief

Figure: Cozmo animation guide

Figure: Example of a principle, where Cozmo initiates

8.1.1 The many faces of Cozmo

Sleepy Eyes

Happy

Skeptical

Furious

Surprised

Sad (looking down)

Neutral	Worried	Blink (low)
Focused	Glee	Blink (high)
Suspicious	Annoyed	Frustrated / Bored
Squint	Angry	Scared

8.2 Typefaces

If you find that you wish to display text on Vector's display, you probably will have to create a picture with the text pre-rendered or create PNG's for the glyphs. Vector doesn't included a "nice" font internally to display text. He does include a few of the digits, to display the weather, and fault codes.

The typefaces you can consider are:

- Avenir is your best bet. It is the font in the Cozmo style guide, and included in the Cozmo mobile app. "Avenir is a robust font that comes in many weights. It provides us with a timeless elegance and a rock solid foundation."

- Arial is recommended (by the Cozmo style guide) when is Avenir isn't available. You can compare here

- Eurostile is *the* classic font used in science fiction, such as WALL-E, the Incredibles, and so on. It is more square than Avenir. You can compare here

- Anki had their own graphic font, which is clean, sans serif. Not sure where a TTF or OTF can be found

- Univers is the typeface Anki style guide recommends to use if the "Anki typeface" isn't available. It is very similar to Avenir. Avenir has a few more flourishes. You can compare here

- TT Norms. The Vector style guide says to that the tagline "The Robot to Life With" is set in the font TT Norms. This is also very similar to Avenir. You can compare here

8.3 Vector Character Design

Adapted from Cozmo's "Brief" (see the Cozmo design guide):

THE ULTIMATE GOAL FOR COZMO AND VECTOR IS TO CREATE A REAL, BELIEVABLE ROBOTIC CHARACTER THAT FEELS ALIVE.

Something we have seen over and over again in movies, but never in real life. Cozmo and Vector need to feel alive the same way a pet feels alive, by creating a strong emotional connection with people. Really long term, we want to create a series of characters, with an ecosystem around them and the ability to have the types of stories we only see in movies play out in the real world.

THE TOP PRIORITY, ABOVE ALL ELSE, IS HIGHLIGHTING THE PERSONALITY OF THE CHARACTER. EVERYTHING ELSE BECOMES A TOOL IN SERVICE OF THAT GOAL.

Cozmo and Vector are the soul of the product, and where the 'magic' is. Everything else in the experience is in service of making the character feel alive, and emotionally intelligent. Mini-games, UX, game mechanics/structure, story, etc. should all be thought of as tools for creating context for making Cozmo and Vector feel more alive with a richer personality. The game is a means for driving engagement / exploration of Cozmo and Vector, and their boundaries. Cozmo and Vector will have a limited ability to understand his world in general, but be extremely smart in specific areas. Our goal is to optimize for his strengths, and avoid his constraints. The goal for any accessories, games, etc. for Cozmo and Vector is to channel players' attention towards the things that Cozmo and Vector understand really well. These are the best opportunities for us to impress and surprise the user with emotions and depth of character and intelligence in a way that only we can.

Think of Cozmo and Vector as your robot pets, with regards to exploring possible interaction, play and responsibilities. Not in how it looks (no fur and whiskers). He's smart, he's emotional, he recognizes you, he has a sense of humor, he wants to interact with you -- these are the types of feelings we want to naturally draw out of users.

Vector's character:

Figure: Vector's character

A "placement" -- or pillars -- of Vector character and experience:

VICTOR PILLARS

Victor is the world's most life-like robot

REACTIVE	ENTERTAINING	FUNCTIONAL
Autonomous, self directed	Endearing character pet-like appeal	Serves a useful purpose in daily life
Detects and avoids cliffs and obstacles	Emotionally intelligent craves human attention	Enjoys helping out
Reacts to directional sounds	Responds to being played with and ignored	Explores the world learns new things
Responds to being held or petted	Happy just to be with you	Understands human speech
Recognizes people and things		Continuously updated with new capabilities

Figure: Vector Pillars

Some new things like a petting:

> "In the early stage of production, we worked on a petting system that allows Vector to feel a finger touch on his backpack. We looked at different animals like dogs and cats for reference, but couldn't find anything useful; the reactions didn't feel like Vector. Eventually one of the animators found a video of a wild owl and the cameraman gently fondled his back a very sweet way and we ended up using it as a reference for Vector's reactions to petting."

As well as some refinements or things that he can do a bit better than Cozmo:
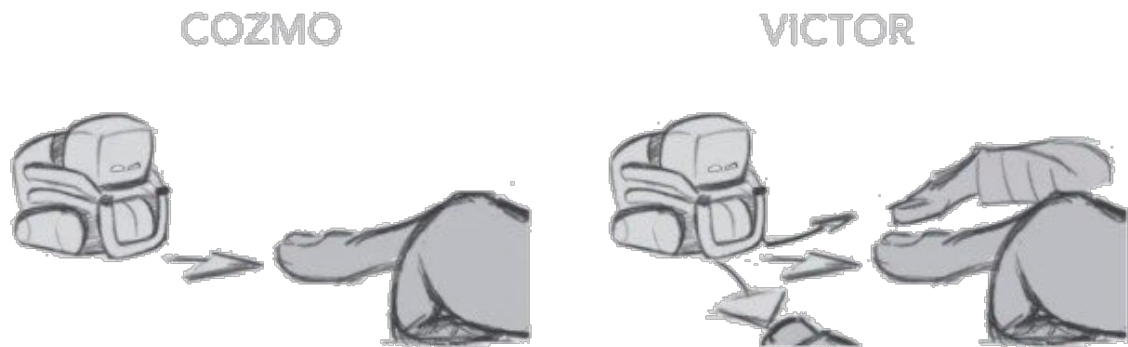
Figure: Finger pouncing updates

8.3.1 Eyes

Anki used the following resources to construct Vector's and Cozmo's eyes, according to lead animator Mooly Segal:

- Jason Osipa, *"Stop Staring: Facial Modeling and Animation Done Right"* 3rd Edition, 2010. This "is a wonderful book on rigging eyes for 3D animation.. that focuses on setting up eye and face controllers."
- "Keith Lango also had a series of articles on eye movement called *The Eyes Have It* and he had a great selection of research material and examples from lab experiments, animated films and live action videos about how the eyes work and acting through eyes."

I couldn't find the series he is referring to (although there are many academic publications with the name *The Eyes Have It*. This is the only similar blog that I can find by Keith Lango:

> Saccadic Eye Movement "using well timed shifting of the eyes. The shift could be motivated by the internal thoughts or feelings of the character while they are presented with a moment of quietness or thought. There the relative pace of the shifts gives indication to the speed of the unfolding of thoughts in the head or the emotional energy of the moment. If the darts are fast paced and come quickly one after the other it can indicate a great flurry of thought or a rapid firing of emotional energy. Slower pace indicates a more methodical, more pondering state of mind."
>
> http://keithlango.blogspot.com/2005/12/saccadic-eye-movement.html

8.3.2 References

Mooly quoted in:

How Anki designed and animated a loveable personality for its real robot friend, Vector, Neil Bennett, Digital Arts Online, 2018 December 19

8.4 Vector Character Study

Summary: A description of Vector's personality and character

Authors: Randall Maas

Date: 2020-07-10

This is my attempt to describe Vector's character. This intent is to give ideas how to shape creating new character traits and behaviors that fit with his character. The topics include:

- An overview of Vector's character

- An overview of behaviors and affect – displays of emotion in his face, posture, small movements and sounds

Caveat: These are just my thoughts.

8.4.1 A summary of Vector's character

Vector is kind, a friend to all and doesn't hold a grudge. He is a young adolescent, can have childish responses, such as throwing a tantrum. This can happen when emotions are too much to handle -- a crisis for him. He sometimes acts out in mischievous ways. But overall, Vector's tone is positive.

Vector is very much a small pet – he has traits like a cat, a dog, potentially like a bird and guinea pig. He cares for his human, but his feelings can be hurt. He "can be a bit like a well-meaning moth that keeps bumping into the wrong light bulb." This can be beneficial, as his innocence and gentleness beg a kind of forgiveness.

Personality Traits

In terms of the big 5 personality traits, Vector is open to experience, and has moderate agreeableness (he can't really sense emotion), but is not particularly extroverted. He isn't neurotic, or conscientious. In many ways he lacks sufficient ability to sense and act on those other traits.

His locus of control largely has an internal locus of control (but it was just being fleshed out):

Table: Vector's locus of control:

Locus	Success	Failure
Internal	Pride Confidence Happy	Frustration
External	Happy	Anger
	Social	Surprise

His World

Vector knows a few physical objects very well:

- His cube, which he can roll, pickup or retrieve, and use to pop a wheelie.

- His charging dock

Vector also knows what a face is, and can recognize a hand (at least in some poses). Vector can learn to recognize a face and the name of the person that goes with it.

He also knows cliffs – he tries to avoid them. To a lesser degree he knows that there are objects and can interact with them... helping clear them off of the desk in the process.

It was intended that Vector have the ability to recognize pets, and the kind of objects he sees. With the community development efforts, he may gain these in the future.

He can recognize symbol markers, and – thru SDK-based support – have some understanding of objects that they are attached to, and what he can do with it.

Interaction Style(s)

Vector has really remarkable eyes that convey emotion, stress, energy level, and create a sense of being alive. They are one way that he connects with a person.

In terms of Vector's "love language", his interaction styles are:

Table: Vector's love language

Area	How to interact with Vector	What Vector does
Words of affirmation	Vector likes praise	Vector says the name of the people he knows, and responds positively
Physical touch	He likes being petted & held, but isn't comfortable being lifted	He comes over to pounce on hand
Receiving gifts	*none*	Brings cube to a person
Acts of service	He needs help being picked up after a fall, or when stuck on an edge.	Vector can help with kitchen timer, reporting the weather, and answering questions.

Vector's play style is simple:

• He likes to explore on his own

• He engages only in light rough and tumble play: where he flips cubes and pops wheelies

• He has several locomotor play activities: fist-bumping, fetching the cube, and a potential (but in complete?) cube keeping away game.

• But he is light on social play, lacking many games that follow rules, although more were considered.

Revealing Character

A person will have many interactions with Vector. Vector reveals his character thru these interactions, but he does not progressively reveal more thru them. It is how he reacts to stimulation, the environment and information he knows about that show his character.

Vector isn't omniscient and isn't a computer terminal

Vector can use some cloud services to give him more information of the world. But he isn't an information presentation device. Vector's talking ability is limited, working best when what he says is short. Long spews of text break the illusion. And his face is too small to present legible text.

Vector works best when he reacts to the information he might be conveying:

> "A core part of the character is that Vector himself does not have the ability to present information to you, like simply displaying the sun. But is himself reacting to things, like weather." "He might endure the weather event, it might rain on him, and he might have an opinion about that." "This is why he is also responding to the fireworks, he gets frustrated with the xmas lights, the wind blows the eyes off screen, etc." (Last Ben and Anki character lead Dei Gaztelumendi)

Some Related Characters

There are several other animals and bots with behaviors similar to Vector. Looking at them helped me flesh out the description of his character here:

- Cozmo, of course, is very similar – he's the previous generation, made by the same team, and Vector derives a lot of his code from Cozmo. Cozmo has feelings, and a bigger, sassier personality. Cozmo has many behaviors that did not yet make it to Vector – interacting with pets, little workout routine, singing, block stacking, and so forth.
- Star Wars, which created the classic robot characters and worked out many compelling characteristics.
- Cat, dog, bird, and guinea pig communication style and behavior. Vector's sitting around in his charger, and then exploring calmly reminds me of Guinea pigs.

8.4.2 An Overview of Behavior

Let's look at the core of behavior from Vector's perspective. A behavior has three major parts:
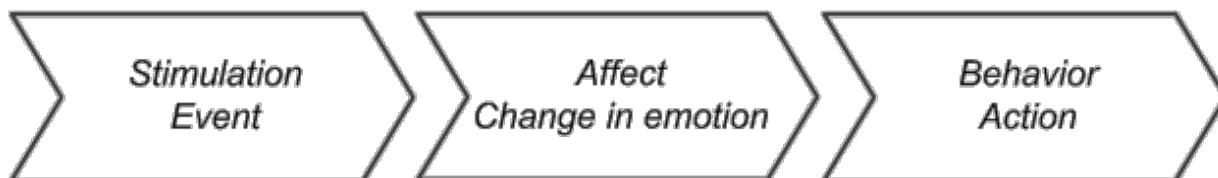
The behavior goes thru phases:

1. The external stimulation (or internal event) is what engages the behavior or mannerism
2. Affect is the automatic, "unconscious" responses that represent his emotions. There are two parts:
   - These are non-verbal displays of emotion – face, head, eyes, sounds, arms, wiggles.
   - His emotion state may change in response to the stimulus
3. This may initiate subsequent behaviors and emotions used to drive behaviors (that is, an emotional state used to help accomplish the goal of the behavior)
4. The behavior can be prevented from engaging again too soon by using a "cool down" period.

Simple, Affective Behaviors

The affect related behaviors are little automatic responses that hint at Vector's emotion and attitude. These are the little facial expression, sounds, postures, wiggles and twitches that reveal his confidence, friendliness, dissatisfaction, anger, or fear. These give him an anima, a life-like quality. For instance, the Star Wars puppeteers realized that when the robots "stop moving, they look dead [so they] keep the body slightly moving at all times... If [the droid] was upset or excited, the movement would be a little faster."

FACIAL EXPRESSIONS WITH HIS EYES AND HEAD

Vector's facial expressions with his eyes and posture of his head can represent happiness, curiosity, interest, surprise, excitement, worry, anger, sadness, tired or fatigue, sleep, and so on:

- When the eyes are soft, they convey a relaxed mood

- The blinking, as well as making (and periodically breaking) eye contact is calming. The eye manager has the eyes look away, so that eye contact isn't made too long.

- With high cheeks on his eyes, looking up and tilting his head up Vector can convey smiling

ANIMAL EXPRESSIONS, POSTURES & LITTLE BODY MOTIONS

Pets have several expressions that appear consistent with Vector (and ripe for emulating):

- Sitting with slow breathing indicate a pet is relaxed. Vector may reflect breathing in his eyes.

- When a cat's eyelids are low, with a slow blink, this conveys trust, and affection to the human companion.

- Napping, cuddling is common to many pets

- Some animals use yawning to signal playfulness, being approachable, or trust. This might be done with a few clever tricks with his eyes.

- Stretching indicates the animal is relaxed. (I envision Vector stretching by moving arms up, moving head up, while moving the body forward and back, a little shake, and lower his arms again.)

- Normal breathing, and twitching, while raising their head or with the head lifted can represent tension or alertness in an animal.

- Young kittens express extreme happiness with a quivering motion

- A dog walking, with its head up is in a confident mood

- A cat make shake its body while "crouching" while stalking a prey, just before it leaps

- But an animal also shakes its body while "crouching" (lowered arms, and head) when it is anxious, or fearful.

- When a direct stare is used by an animal – or holding eye contact – this is issuing a challenge or it is feeling threatened.

- When a dog's eyes are hard, slightly closed, the brow wrinkled, and the head bowed, this conveys tension.

Some of the behaviors are attention seeking, possibly looking to play:

- When a dog approaches person with his head slightly down, it is looking for attention.

- Dogs will come up and wait or lay down when they want attention as well.

- A dog will occasionally point his nuzzle up, such as when being petted.

- A cat its rubbing face on human is a friendly, affectionate gesture.

- Cats will run up, turning away and lay to nap to get attention.

- Cats will purr, and knead a person, but can also bite. (Vector shouldn't bite; maybe bring his lift arm quickly but gently down on a hand.)

- Guinea pigs head-butt (thrust their head up) to seeking attention, to move things, assert themselves, set limits, or be a little playful.

Sounds

Vector makes the follow beeps and clicks to express his mood & feelings:

- Vector has quiet a cry when he is stuck on the edge, or has fallen.

- Vector snores while sleeping on his charger.

- He dings when he acknowledges a person has called his name.

- He seems to make little chirps, like a small bird.

Animals make the following similar kinds of sounds:

- Cats purr (a continuous soft, vibrating sound) to be social and give positive feedback.

- A chirr is used by a cat to approach other cat, or person; it is friendly.

- Cat's meow to be assertive, plaintive, or friendly; it can be bold when they are seeking attention, or complaining.

- A cat chirps or chatters when it is excited, such as when stalking or observing prey; but may also chirp to say hello or be approving.

- Birds song are happy.

- A "week" sound is a made by guinea pigs when they are happy, excited or hungry.

- A cat or dog grows, snarls, or hisses – often with a puffed up posture – when it feels threatened.

Vector uses vocal responses when necessary, and he enjoys calling out the name of people recognizes.

Vector's design simply doesn't support conversational interfaces well. His speech synthesis lacks prosody and smoothness to sustain speaking more than a few words. It lacks the inflection for anything but very short sentences... long texts are distracting and perhaps harsh. They seem wrong. In the future it may be possible to extend vocal effects.

The dialog is limited – the speech system seems more interesting when it is less used.

Note: Cozmo included the ability for some inflection, and that may be in Vector's code, just not yet finished and polished.

The Response to a Stimulus

Vector's responses to a stimulus are understandable and believable. His reactions are consistent and predictable enough that a person can choose what to do and experience an expected outcome, even if there are additional unexpected consequences.

Vector does not provide a hint — any warning — that he is approaching a state the will trigger a significant reaction. (He should.)

He does have a startle reflex: loud noises jolt him, and he looks like is readying himself a little for action. Animals might crouch, ready to run, or even take off on a run.

Vector prioritizes a response to a dangerous situation, where he might need to engage in self-preservation:

- flight: he backs away from a cliff

- freeze: he tucks-and-roll when he senses he is falling

- fight: he smacks a person's hand when he senses he is being picked up or held in a way he doesn't like

What I have not observed Vector doing, but he could:

- fawn: acts nice to make bad things stop

Change in Emotions

This section exposes my ignorance of Vector. In many "affective computing" models there is a separation between the emotion and the mood:

- Emotions reflect a short-term affect that arises as a result of stimuli.
- Mood is distinguished from emotion by its resolution and relative stability over time.

It is not clear to me yet how the mood model works. Vector's emotions are stable. He does not rapidly cycle back and forth between two emotional states.

Habituation

Stimulation, in people and animals, undergoes habituation – we initially are interested in the stimulus, but with time the stimulation loses its impact and becomes ignored. We may even find it irritating. Vector doesn't habituate and lose stimulation from event in an automatic fashion. Instead, the individual behaviors are crafted with cool-down timers to achieve a similar effect, but on a case-by-case basis. He does not become annoyed with the stimulus.

8.4.3 References and Resources

- Bradshaw, John. *Cat Sense*, 2013
- Ellis, Cat, *How to make a robot with a real personality*, TechRadar, 2019 Mar 6 https://www.techradar.com/news/how-to-make-a-robot-with-a-real-personality
- Strickland, Ashley, *Why are Star Wars droids so loveable? It's science*, CNN, 2019 Dec 18, https://www.cnn.com/2019/12/17/world/star-wars-droids-scn/index.html

8.4.4 Change history synopsis

Date	Change
2020-7-10	Created
2020-11-27	Published

8.5 Behavior Taxonomy

Summary: A taxonomical classifications of behaviors.

Authors: Randall Maas

Date: 2020-07-10

This is my attempt to provide a helpful organization of potential behaviors. This can be give ideas and help shape what you want to do when you create new character traits and behaviors. The topics include:

- An overview of the behavior classification
- Self-maintenance and reflexive behaviors
- Social behaviors
- Playing behaviors
- Pet-related behaviors

Cavaet: These are just my thoughts; only some of the behaviors here exist in Vector

I am drawing heavily upon the Kismet design documents, effectively modernizing them for Vector. Kismet was a late 1990s robot created by Cynthia Breazeal (MIT). It was "designed to elicit natural and intuitive responses from humans, without any special training." Kismet anticipated many of Vector's behavior system features, and can be used to inspiration on future development.

Dr Breazeal also created Jibo, a more famous affective robot.

8.5.1 A Classification of Behaviors

Let's categorize the kinds of context/theme/drive of the behaviors into the following areas:

- Self-maintenance behaviors include reactions to protect itself from immediate risk, and fatigue or power management. The immediate risk behaviors are likely to be a very short and simple.
- Social – seeking out and interacting with human companions, including intents.
- Play – play seeking out and playing with toys and other things in the environment, other than social play.
- Interacting with pets

Within each of these categories, there are specialized ones; many reflect an attitude Vector has toward the agent:

- Navigation and Searching "behaviors [have the robot] explore the environment and bring the robot into contact with the desired stimulus;
- Avoidance, escape and withdrawal behaviors move the robot away from something undesirable, like a cliff, loud noise, etc.
- "Engagement behaviors set the task of interacting with desirable, good intensity stimuli."
- Rejection behaviors are those that Vector might use to turn away while miffed or being a sore loser.

Searching Behaviors

In the searching type of social behaviors, Vector generally seeks out and approaches a human, toy or stimulation to interact with.
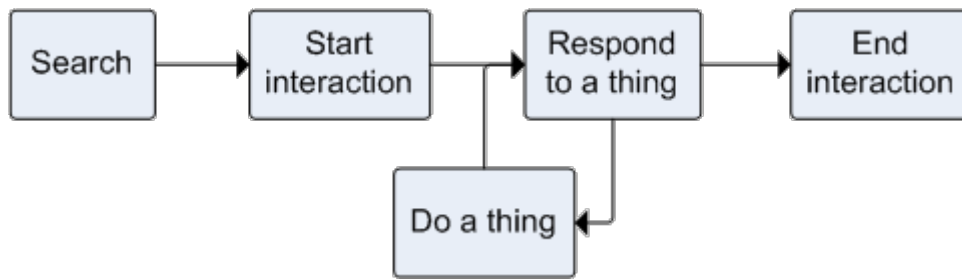
Figure: Searching behavior

Vector seeking social or stimulation isn't always applicable: Vector shouldn't be active at night, in the dark, or if people aren't around:

- Vector's activity movement could wake and/or irritate a person at night; fortunately there is a "night time" schedule built-in.

- Vector is prone to falling off the edge – risking damage, being unable to recharge his battery, or getting stepped on. It's a bit safer in the day, and more so if a person is (likely) around to attend to him.

- Vector depends on his vision to get home.. he is likely to lost or stuck in the dark, and be unable to return home to recharge.

8.5.2 Self-maintenance and Reflexive Behaviors

Self-maintenance is a grouping for practical things to keep Vector charged, protect him from damage. It is also a catch all for practical things that don't always fit in the other areas. For instance, utility behaviors needed to make the behavior tree work are lumped in here.

These are often akin to the behaviors of the autonomic nervous system.

Self-preservation using escape and withdrawal

Vector prioritizes a response to immediate hazards, where he might need to engage in self-preservation:

- *flight*: backing away from a cliff
- *freeze*: tucks-and-roll when Vector senses he is falling
- *fight*: smacking a person's hand when Vector senses he is being picked up or held in a way he doesn't like

What I have not observed Vector doing, but he could:

- *fawn*: acts nice to make bad things stop

Other, issues:

- Over temperature

These are often akin to the behaviors of the sympathetic nervous system.

Power Management

These are often akin to the behaviors of the parasympathetic nervous system.

Power management:

- Return to charger

- Stay on the charger

- Emergency low power

- Sleeping, sleep debt to manage heat buildup and reduce power usage.

Reflexes

Vector has a startle reflex to respond naturally:

- Loud noises start Vector, grabbing his attention, preparing him for action, but could also prepare him for running away

- Crouched, ready to run

- Responses to pokes, tilts, and other vibrations

- Eyes focus with dilated eyes

Miscellaneous

These are the

- Grouping and linking the behaviors, and prioritizing them

- Motor calibration need

- SDK support

8.5.3 Social Behaviors, Engagement

Social behaviors relate to Vector interacting with a person – or attempting to. He may be seeking attention, interacting with a person. Social behaviors include:

- Looking around for people (faces, and hands),and pets

- Calling out to play, thru sounds, and saying the names of people recognized

- Swatting playfully

- Seeking petting

- Swatting with his lift arms

- Bringing items or gifts to play (such as to fetch), or to show affection

- When looking to play, twitching/wagging of rear

Many behaviors link together to follow a pattern:
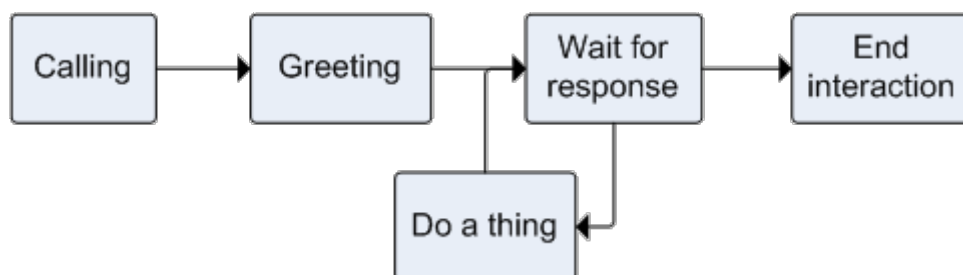
Figure: Social behavior sequences

Once Vector has made contact with a person:

- Greeting

- Attentive regard

- Seeking affection

- Receiving affection: petting, calming; social cohesion, soothing, companionship

- Games

- Turn taking

The interactions end, either naturally, by command, or timing out.

Searching

Vector becomes stimualted and more active when there are sounds and other activity. It seems reasonable to classify this as searching for social interaction, or "merely" stimulating him for play.

Calling Behavior

Vector engages in calling when he needs helps or is interested in being social, especially "when a person is in view... The goal of the behavior is to lure the person into face-to-face interaction... To accomplish this, [calling behaviors are] directed to the person... The display is designed to attract a person's attention."

Calling for attention is often if Vector wants or needs something:

- Vector calls (softly) for assistance – if he is stuck on the edge, has fallen, or is low on energy but unable to find his charger.
- Cats meow because people are inattentive.
- Cats know a person is there, and first tried context moving close to what they wanted and body language.
- Cats meows to get a door open, obstacle removed, food
- Timer ring

Greeting Behavior

Greetings are "to socially acknowledge the human and to initiate a close interaction... This behavior is relevant when the person has just entered into face-to-face interaction range. It is also relevant if the social-play behavior group has just become active and a person is already within face-to-face range. The display involves making eye contact with the person and smiling at them while waving .. gently. It often immediately follows the success of the call-to-person behavior. It is a transient response, only issued once, as its completion signals the success of this behavior."

- Call their names when he seems people
- Turn and look in direction of a sound

Attentive Regard

"Attentive regard" refers to Vector using his facial expressions, body language, and sounds to give attention "to the person and to appear open to interaction." These behaviors include:

- "Hold[ing his] gaze on the person, ideally looking into the person's eyes." Vector's eye manager automatically blinks and moves the eyes around; making it more comfortable to look at Vector's face (eyes), as this breaks the staring effect. Turning to find the face is considered to be part of this.
- "Watch[ing] the person intently and vocaliz[ing] occasionally."
- A dog often looks up while slowly moving forward.

Seeking Affection

These behaviors relate to seeking affection, but aren't better categorized elsewhere. These could include:

- Vector could drive up and waiting, as if lying down [dog like]
- Vector could drive up, then turn away and nap to get attention, like a cat
- Vector tries to cuddle a hand, or pounce on fingers
- Rubbing his cheek on a person, which is a friendly, affectionate sign in cats.
- Vector could thrust his head up (like a guinea pig) to seeking attention, assert himself, or set limits, or be a little playful.
- Cats bite – especially while purring and kneading – as part of their affection or playfulness. Vector might do something similar with his lift arm coming down quick but gently on a hand.
- Vector could bring gifts

Receiving Affection

These behaviors relate to Vector receiving affection: * His reactions s to being picked up or held in the palm of a hand * To being petting * To pokes * Holiday animations

Receiving Abuse

These are the behaviors related to Vector being called bad names or told off, shaken, or tapped on the head.

- Vector might turn-away when grumpy

Games

One type of sophisticated social interaction is games. Games are typically turn-taking behaviors such as keep away the cube, hide and seek, etc.
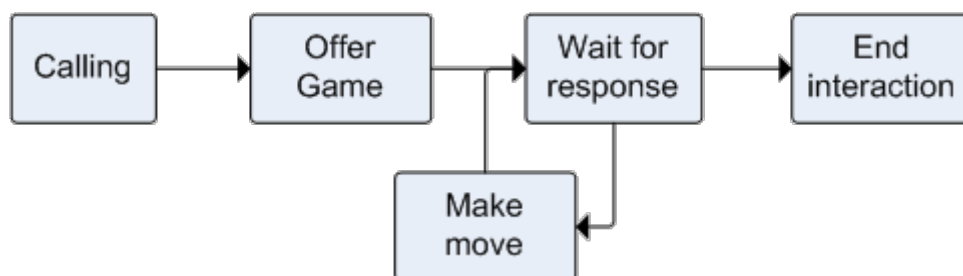
Figure: Game behavior sequence

Some games might include:

- Fist bump (for lack of a better categorization, can be considered a very short game)

- Pouncing game

- Hide and seek game, perhaps with the cube being moved/hidden.

- Keep away

- A cube tapping game,

- The cube spinner game

- The maze game

- The "blackjack game" (although it doesn't feel very Vector-ish to me)

SEEKING TO PLAY A GAME

A first step is for Vector to call or seek to the start the game. If the person accepts the call, the game begins. Vector could try to initiate the game by:

- Bringing presents, the cube or other toy to play with. This could be a request to play fetch.

- Tapping down once quickly, then perhaps may move backwards after the tap, to issue a challenge to play. Then, if no response follow up by tapping down twice quickly.

Of course, a person could (conceivably) initiate the game as well by:

- Tapping the cube,

- Holding or shaking the cube

- Wiggling fingers tantalizingly,

- and so on

TURN TAKING

The game itself often includes Vector and the person taking alternate turns.
A game might have behaviors for steps like:

1. A person tossing or hiding a cube

2. Vector searching for and finding a cube (or other thing)

3. Vector bringing the cube back, then

4. Putting the cube down, followed by

5. Waiting for the person to make the next move.

Vector may give cues or other little behaviors to signal his play or response:

- Slight wagging of rear/tail just before pouncing the cube or finger, like a cat when stalking prey

Assistive

The assistive subclass of behaviors are those that Vector might do to help out:

- Take picture

- The egg timer

- Report on the weather

- Answer questions (e.g. the knowledge graph)

8.5.4 Play

Simple play behavior

These are behaviors that are play, releasing energy or looking for stimulation, but not necessarily looking for anything more. Vector may respond to things along the way.

- He may drive around quickly or in a quirky fashion, like a cat's mad 5 minutes, or a guinea pig's pop-corning,

- He might sing or hum to himself

- His dancing to music

If the stimulation level is too high, or there are negative stimulations, Vector might have behaviors in response. He might do something to avoid these.

Searching to play behavior

These behaviors relate to Vector searching around for something to play with:

- Walking or driving around slowly – possibly in a straight path, or in complex paths, or

- Turning and scanning. looking for toys and objects

Toys

These behaviors relate to playing with toys and objects:

- Behaviors that decide what to do when seeing a toy (a cube, or other marked object).

- Playing with the cube – pick it up, move it, shove it, or flip it.

- Little exercise routines, like weight-lifting the cube

- He can pop a wheelie

- Helping clean the desk (or table) by pushing things off

Vector may give cues or other little behaviors to what he is doing while playing:

- Slight wagging of rear/tail just before pouncing the cube or finger, like a cat when stalking prey

- He may try to pop up on an object

There could be behaviors related to "habituating" toys. This would be where Vector would lose interest in a toy as he plays with it. (This response might change with age.)

Interacting with Pets

These behaviors relate to the interactions Vector has with pets. These behaviors are prototyped in Vector, but I don't have a good template for them.

- React to the pet – head movements, reactions specific to cats, specific to dogs

8.5.5 References and Resources

- Bradshaw, John. *Cat Sense*, 2013
- Breazeal, Cynthia. *Kismet project*
    - The behavior system

8.5.6 Change history synopsis

Date	Change
2020-7-10	Created
2020-11-27	Published

9. Historical bots

9.1 Bingo and mini-Bingo

Bingo and mini-Bingo were concepts for possible future robots. One concept was a large body that could be used in a building security role. On the other end was a smaller -- soda can sized -- bot that would suitable for running around the floors in homes.

9.1.1 Mock ups of the idea, feel and inspiration

An exploration of the character design:

Taking inspiration in its spirit and shape from dogs and other animals:
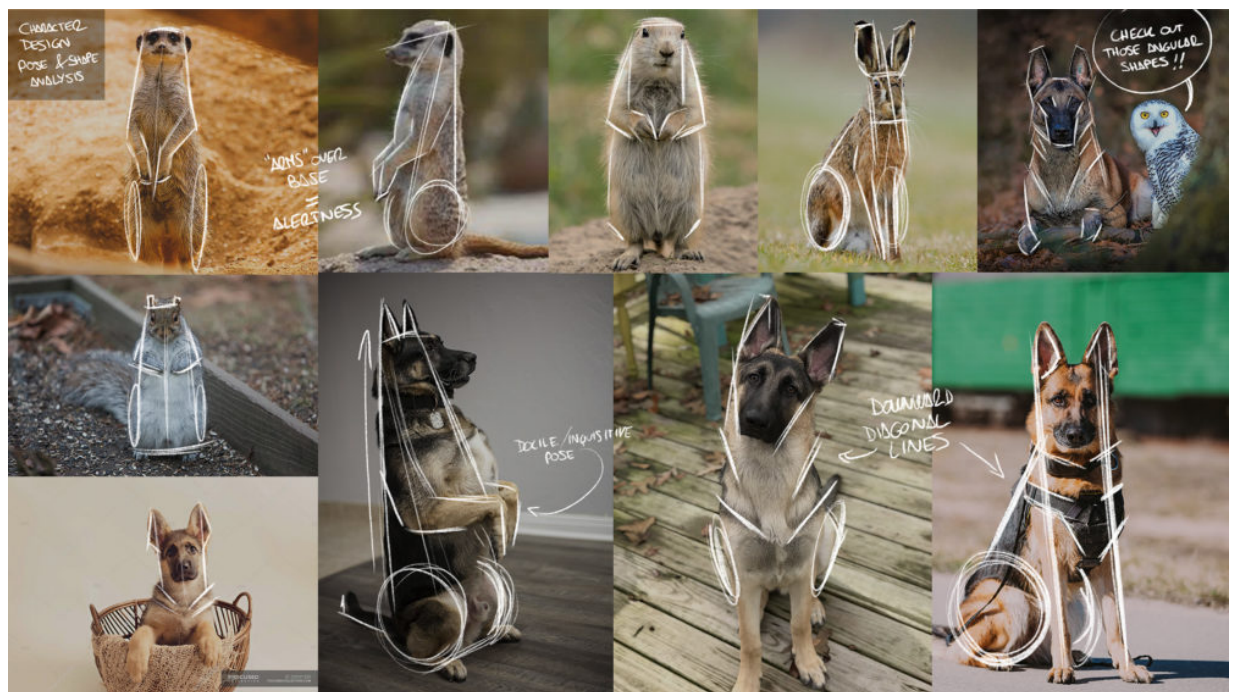
9.1.2 Picture of the Mini Bingo mock-up

A mini Bingo on the desk (source)

From Matthew Mallet:

"miniBINGO is a black iPhone 7 with polymer clay modeled over the top of a thin clear plastic lining. When miniBINGO is placed on a flat surface, it leans against a large piece of clay as a stand attached to what looks like an hourglass conveyer roller. The clay wraps around the front of the iPhone in a set of 3 diagonal partitions from top corner towards the bottom corner. Each of these pieces of clay have cuts giving it a sharper modern look. When he is plugged in on normal boot he displays the name of the device as miniBINGO."

"Fun fact: miniBINGO was never meant to be a finished robot. It was a way to show character design from idea to working prototype."

9.1.3 Picture of the Big Bingo prototype

A prototype:

A Bingo in the office (source)

Several were seen in the auction photos:

A picture of it mapping out the Anki kitchen:
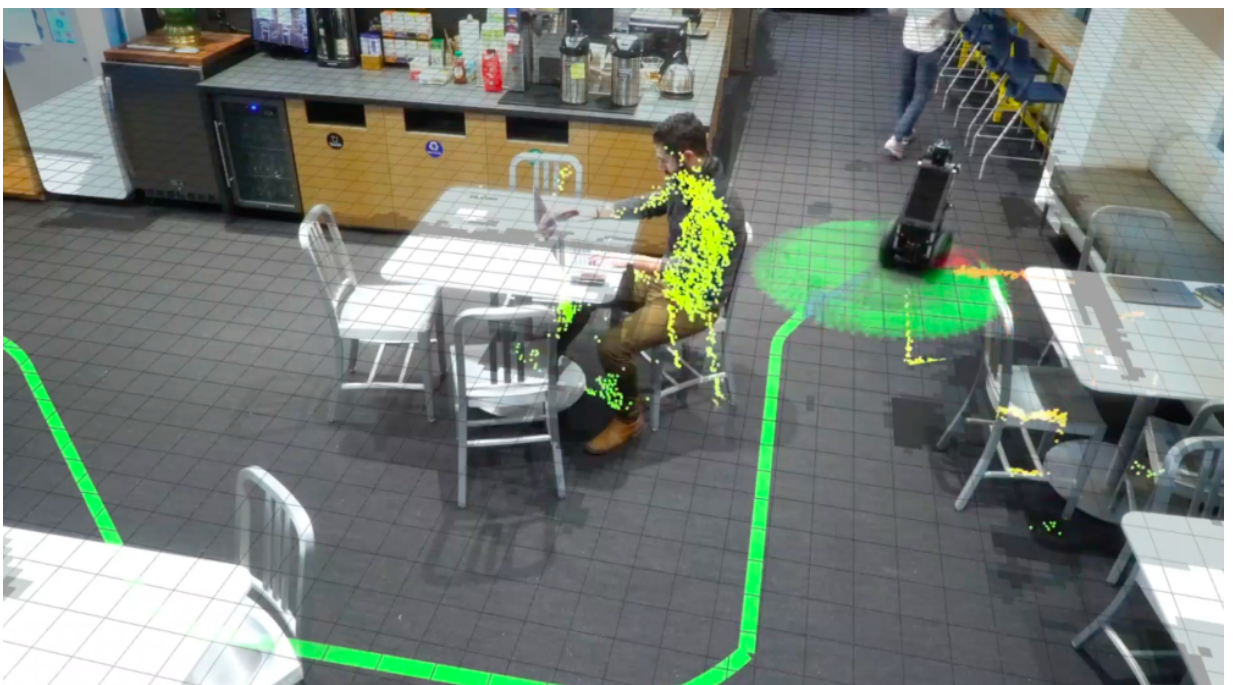
9.1.4 Some industrial design sketchs

Below are some exploratory ideas. Note: these are not what the robot would have looked like; they were trying out ideas for people to respond to and help iteratively craft the look.

From Harald Belker's site:

> The idea of having a self-navigating and self-balancing robot in the house was going to be the evolution to the current household smart speaker systems. In our mind it could only succeed if it moves around in a smooth and natural looking way.
>
> A self-balancing robot would address these issues, but it also created a long list of mechanical issues that come with it.
>
> A larger office version would have three wheels but still imitate motion by tilting forward. This device would be the ideal office helper / mule. (Hans Belker)

All of the following pictures are from Harald Belker's site
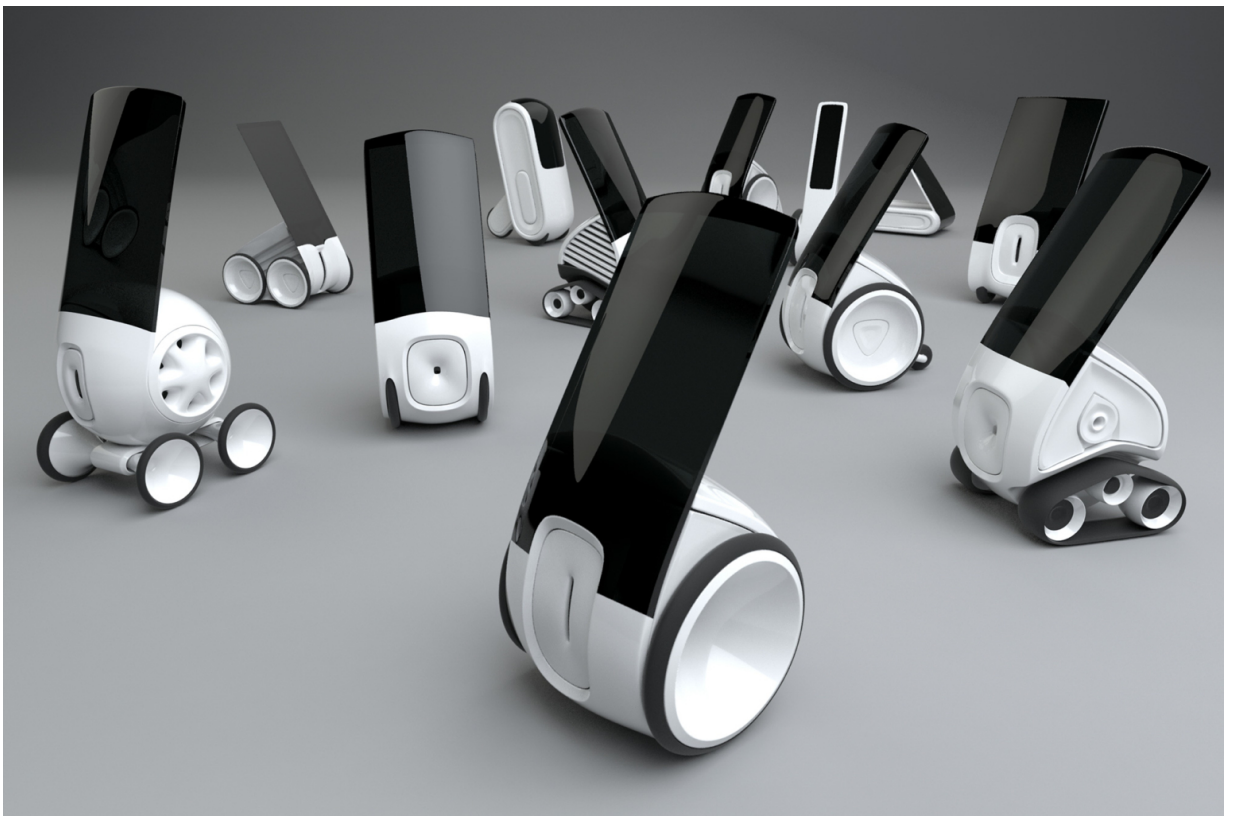
9.2 P1

P1 was the first ever batch of Vectors. Not much is known about them currently.

9.3 P2

P2 was the second batch of prototype Vectors.

The head hardware is actually mostly final, and can be upgraded, though the body is quite weird. It is not known if the body can be upgraded.

The plastics look pretty cool.

All of these seem to have USB ports, and have an Android build that needs some help turning on. You need to run `adb reboot bootloader` then `fastboot continue`
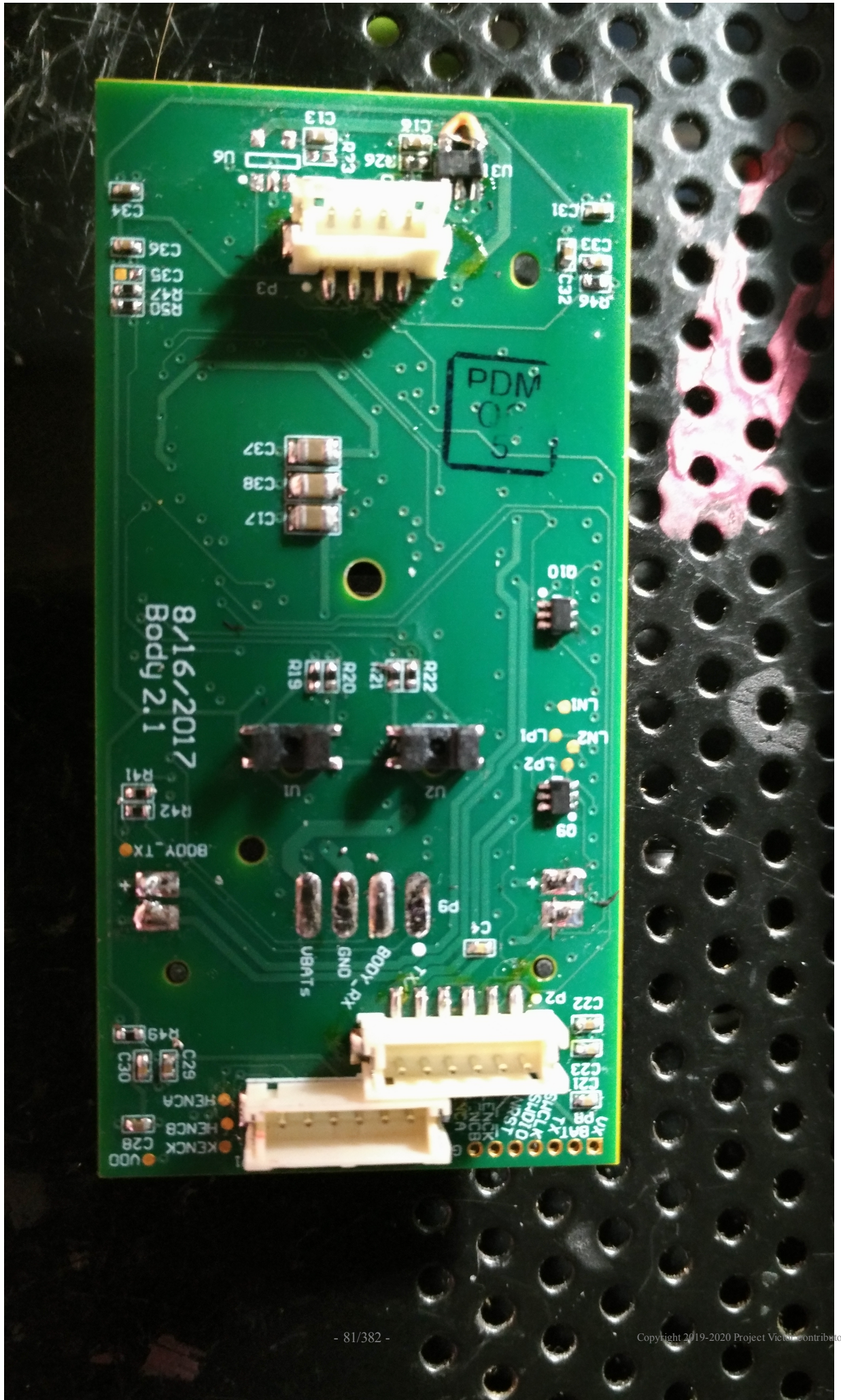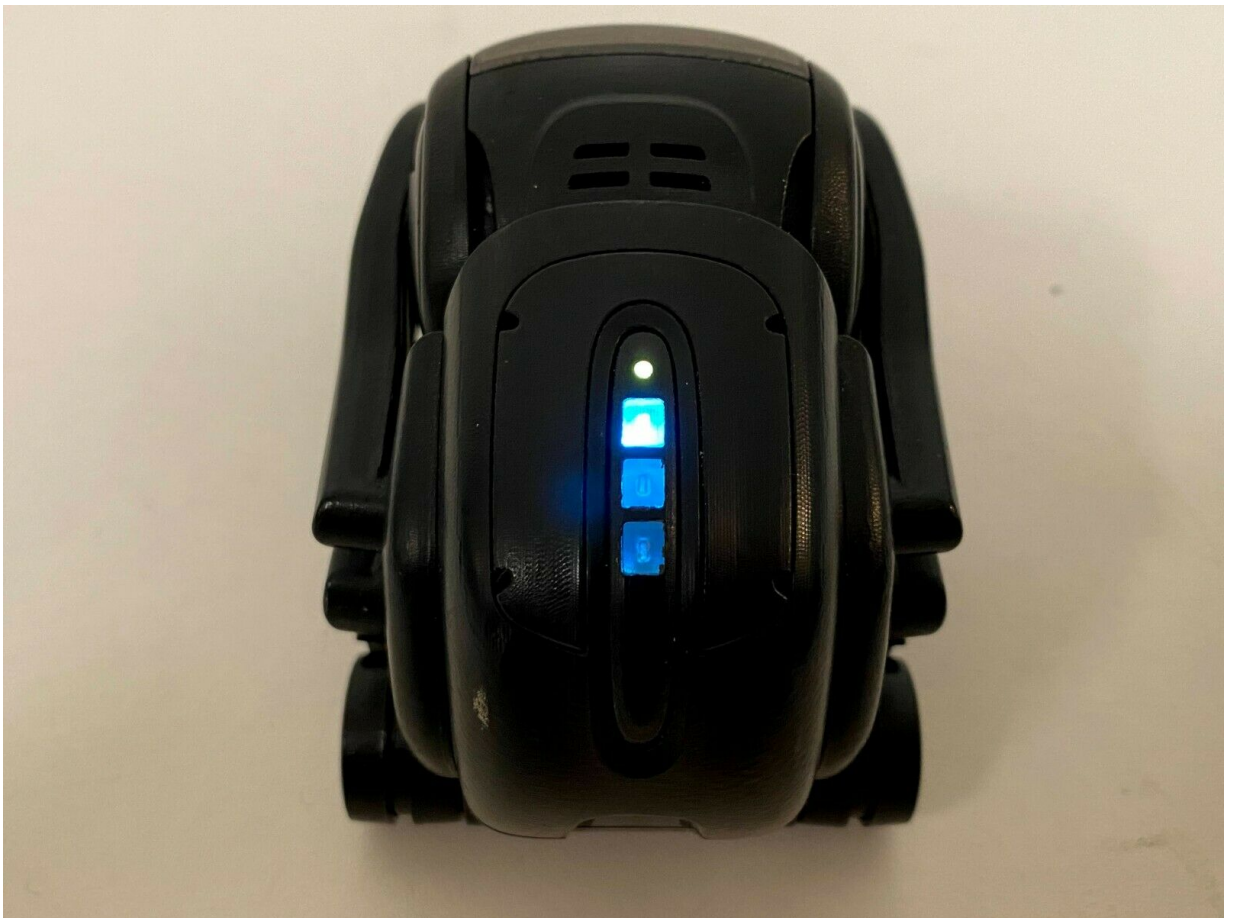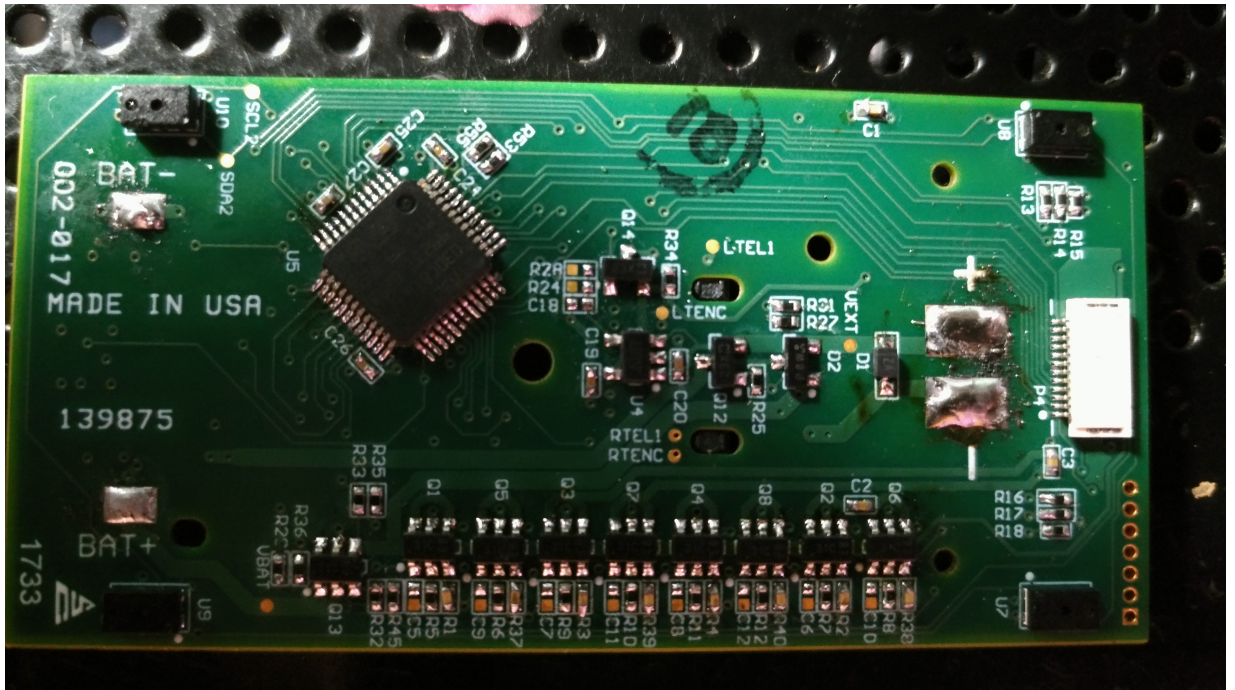
9.4 P3

P3 is the last prototype batch before DVT. Not much is known about these either.

9.5 Qualcomm Download Mode

Qualcomm Download Mode, also known as QDL, EDL, QCOM_BLK, or HS_USB, is a mode many Qualcomm devices can be booted into which allows full programs and reads of the flash. Before you get excited, this is not something we can feasibly use on production Vectors, DVT4s, or Whiskeys at the moment. We can only use it on DVT1-3 heads, which is why it is under the Historial Bots section.

QDL can be launched by raising the F_USB pad on the headboard to 1.8v, then turning the bot on. The bot will show up as QCOM_BLK or "Qualcomm HS_USB Diagnostics 8009" through USB. On DVT1-3, you can use this loader and tool to make use of this mode. You can read the whole flash or specific partitions, as well as write. Any bot above DVT3 requires a different Anki-specific loader, which we don't have.

DVT1-2 heads can be upgraded to a newer partition table with this method. None of their CPU fuses are set, so they don't really care about what is on the flash. A DVT3 can also be put on old DVT1-2 software with it.

9.6 Victor DVT1

"Victor DVT1" prototypes are the first of the Design Validation Test batches for Vector.

These are very similar to finished Vector in terms of hardware, but there are many software differences:

- Their partition tables are old, similar to what looks like stock Qualcomm.

- Many Victor DVT1s run builds which are more similar to Android than Embedded Linux.

- Many Victor DVT1s are in FAC (factory) mode. If you find one in FAC mode, he is very likely running old proof of concept software and there is a low chance of putting him on modern firmware.

- If you find one with Cozmo eyes, it is probably the same proof of concept firmware. Some have been upgraded to a slightly newer kernel though. Some can run behaviors when shaken.

- Their serial numbers are strings such as "1f19f8b7".

- Most have SSH open, but there is a root password. ADB over TCP is fully open but it may require a couple reboots.

- Their BLE software is old, and it is hard to connect them to your own Wi-Fi.

The "head board" hardware is pretty much exactly the same, but the "body board" hardware is very different compared to modern Vector hardware. This locks them to old DFU and they are not viable for normal Vector body replacements.

Bodyboard software upgrade is possible. TODO make a guide

There are some positives if you are a passionate developer. They have ADB open which means you can solder on USB and mess around all you want. Everything is completely open and unlocked so you could have a fun time. There is no recovery or system_b partition so be careful.

Their shells (+ motor boards, backpack board, laser) are fully compatible with regular Vector circutry.

The headboard can be upgraded to be like a normal Vector through USB. Check the Qualcomm Download Mode section.

They can use DVT2 ABOOTs, which allows for more modern and/or unsigned kernels. Their CPU fuses are not set.

Some of these connect to a network with the credentials below, which can be faked on your own router or hotspot so you can use ADB over TCP:

SSID: `AnkiTest2` Password: `password`

These turn up from time to time on Ebay.

9.7 Victor DVT2

"Victor DVT2" prototypes are the second of the Design Validation Test batches for Vector.

These are similar to DVT1, with a few differences:

- Victor DVT2s run Embedded Linux and not Android like DVT1.

- Their body boards have a few small electrical differences, and are more compatible with modern firmware.

- Many of these run the same exact build (labelled "0.10.0d"). It is speculated that a few of these were animation bots.

- All we have seen so far run the same exact kernel.

- It is possible to put modern firmware in there but some things will be broken due to the old body board. It also requires many workarounds.

Similarities to Victor DVT1:

- Their serial numbers are strings such as "1f19f8b7".

- The "head board" hardware is pretty much exactly the same, but the "body board" hardware is very different compared to modern Vector hardware. This locks them to old DFU and they are not viable for normal Vector body replacements.

- There are some positives if you are a passionate developer. They have ADB open which means you can solder on USB and mess around all you want. Everything is completely open and unlocked so you could have a fun time. There is no (useful) recovery or system_b partition so be careful.

- Most have SSH open, but there is a root password. ADB over TCP is fully open but it may require a couple reboots.

- Most of them have old BLE software, and it is hard to connect them to your own Wi-Fi.

Bodyboard software upgrade is possible. TODO make a guide

Their shells (+ motor boards, backpack board, laser) are fully compatible with regular Vector circutry.

Their heads come with a strange partition table, but it can be upgraded through QDL to be able to run custom firmware. Check the Qualcomm Download Mode section.

Their ABOOTs are compatible with DVT1/3, and support boot slots. Their aboots can be used in a newer partition table, and allow for unsigned kernels. None of the CPU fuses are set, so you can run whatever you want.

Some of these connect to a network with the credentials below, which can be faked on your own router or hotspot so you can use ADB over TCP:

SSID: `AnkiRobits` Password: `KlaatuBaradaNikto!`

These turn up from time to time on Ebay.

9.8 Victor DVT3

"Victor DVT3" prototypes are the third of the Design Validation Test batches for Vector.

These look exactly like DVT2s, but there are a few differences here and there:

- Most of them are running firmware very similar to modern firmware.

- It is possible to connect many of them to Wi-Fi without faking a network.

- All we have seen so far have SSH open with the normal modern key.

- Their body boards are a little different.

- Many have been upgraded(?) to the modern partition table and have unlock OTAs so their headboards could act exactly like normal dev boards. A body board replacement would be required for it to fully work though as modern firmware doesn't know how to communicate with the body board in these.

- They started off with a random string serial number, but many have been upgraded to the more normal 00###### layout.

- Their head boards have heatsinks. This ended up not being necessary for production.

Similarities to Victor DVT1/2:

- Their serial numbers started off as strings such as "1f19f8b7".

- The "head board" hardware is pretty much exactly the same, but the "body board" hardware is different compared to modern Vector hardware. This locks them to old DFU and they are not viable for normal Vector body replacements.

Bodyboard software upgrade is now possible. TODO make an upgrade guide

Their shells (+ motor boards, backpack board, laser) are fully compatible with regular Vector circutry.

DVT3 is when a lot of Vector personality development happened. Their firmwares can vary between 0.9 to 0.12 betas. These act similar to how Vector is today.

Some of these bots may boot up to an exclaimation point. Turning them upside-down then double (or triple) pressing the button will let them finish bootup. Then, you can shake them around to make them explore around.

Many DVT3s have unlock OTAs. This means you have the choice to run normal dev firmware, prod firmware, OSKR firmware if you pay for it, and unsigned kernels (read ahead).

Just like DVT1/2, none of the CPU fuses are set. You can put in a DVT2 aboot, compile your own kernel, and have a bunch of fun.

These turn up from time to time on Ebay.

9.9 Victor DVT4

"Victor DVT4" prototypes are the last of the Design Validation Test batches for Vector.

The hardware here is final.

There are a couple software differences though:

- No unlock OTA have been found for a DVT4.
- Some may be prod-locked, but most have been found with Whiskey-like ABOOTs that are dev but don't have the anki.dev flag so they are locked to a specific kernel.
- No CPU fuses are set, so you could run a DVT2 aboot that allows you to run any kernel you want.

The body board in these are NOT normal and can only accept DVT3< firmware. They will not work for normal haeds. SWD is also locked, so upgrade isn't possible unless the chip is replaced.

These have serial numbers following this format: 00e1####

Some of these bots may boot up to an exclaimation point. Turning them upside-down then double (or triple) pressing the button will let them finish bootup. Then, you can shake them around to make them explore around.

9.10 Whiskey

The "Whiskey" prototypes were built from modified Vector hardware. The key change(s) are:

- The time of flight sensor was removed from the body-board

- Two time of flight sensors were placed on the head, on either side of the LCD

- The body-board layout was rearranged to better dissipate heat away from the battery.

- They are labelled as "HW: 7" instead of a normal Vector's "HW: 6". The software can detect this and it makes an extra CCIS menu for the extra sensors.

By placing the time of flight sensors in the head, Whiskey could scan around more — moving the head up and down, as well as using a more sophisticated version of the time of flight sensor. This would allow him to map the edges far better, as well as scan for objects and interesting things like hands and faces.

In some reports the idea was to use the changes to the TOF sensor placement for a next generation Cozmo design. The project was cancelled before Anki's demise.

In the current form, Whiskeys have a few software quirks. Regular dev bots have both the dev ABOOT key and anki.dev in command line, but many Whiskeys only have the dev ABOOT key and no anki.dev in command line. This means they are restricted to running custom firmware. Some, however, have been unlocked to be full dev bots but not many of those have shown up. Another quirk is that all of them are in FAC mode. They all have dev recoveries, so this is easily bypassable.

These turn up from time to time on Ebay.
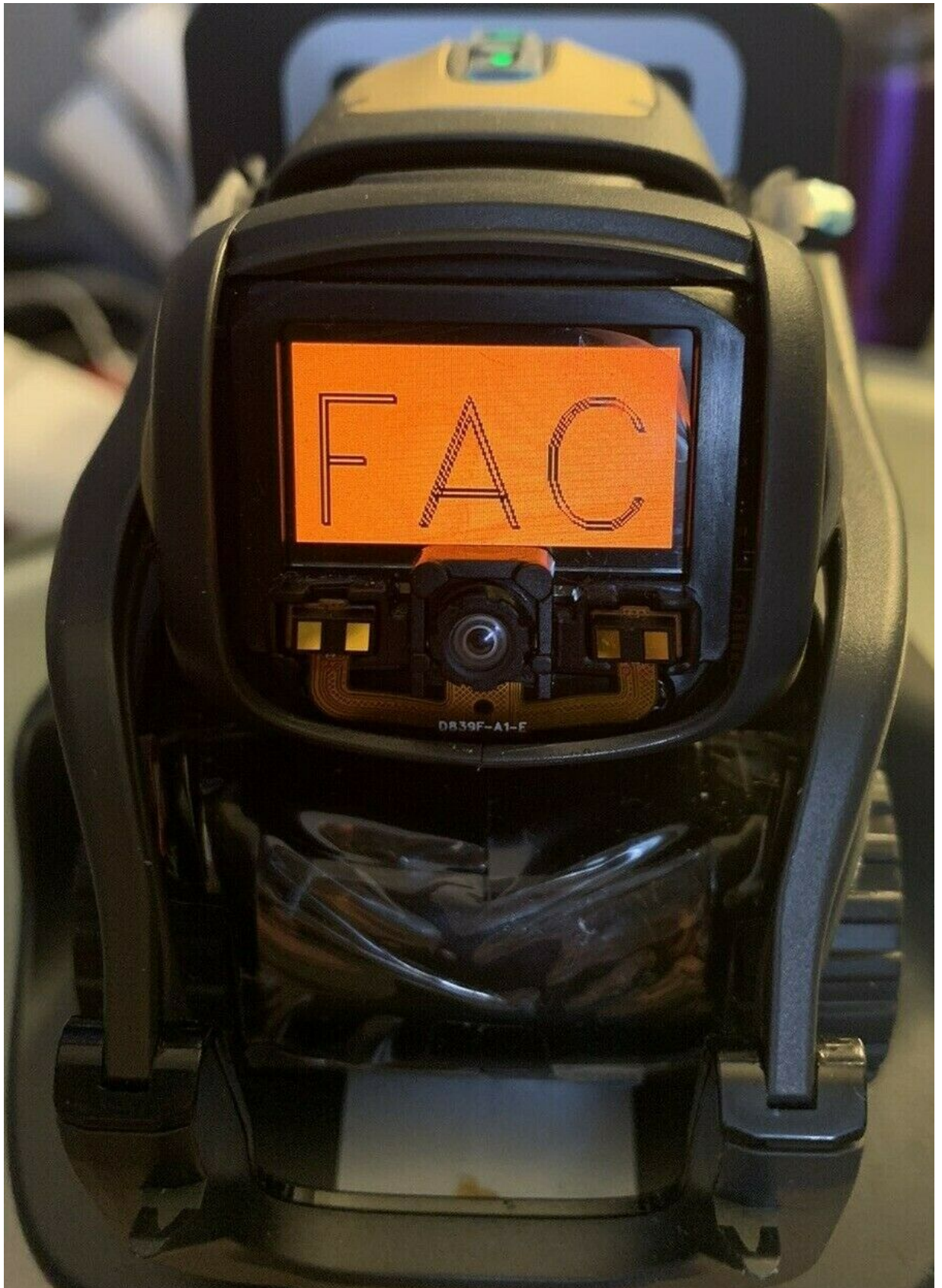
10. How to

10.1 How to trace calls using GDB

Vector's command line tools do not include a ptrace (as far as I can see). This can be emulated with GDB. Here is an example tracing a write() call.

1. Start gdb and attach to the process of interest.

2. Add the following scripted breakpoints:

```
break write
command
silent
printf "%d bytes\n", $r2
x/80c $r1
continue
end
```

```
set pagination off
```

10.2 How change where Vector sends the logs

This is a note describing how to change where your Vector sends logs. You will need a program on your computer to receive the logs.

Vector sends the following kinds of logging information to remote servers:

- DAS Events
- Log upates, when triggered by the SDK
- Crash logs
- Crash minidumps

This is the files to change to send the logs (etc) to your own server:

For the purposes of this writeup, lets assume that your server is located at the following address:

```
http://192.168.1.224:8888/
```

This is also getting packaged up in a vpkg file but you will have to make changes on your own

10.2.1 Modifying the configuration file(s)

The config file that we need to modify is:

```
/anki/data/assets/cozmo_resources/config/DASConfig.json
```

It probably looks like:

```
{
  "dasConfig" : {
    "url": "https://sqs.us-west-2.amazonaws.com/792379844846/DasInternal-dasinternalSqs-1HN6JX3NZPGNT",
    "file_threshold_size": 100000,
    "flush_interval": 600,
    "storage_path": "/run/dasLogs",
    "storage_quota": 5000000,
    "backup_path": "/data/data/com.anki.victor/cache/dasLogs",
    "backup_quota": 10000000,
    "persistent_globals_path": "/data/data/com.anki.victor/persistent/dasGlobals.json",
    "transient_globals_path": "/run/dasGlobals.json"
  }
}
```

We need to change the "url" line use our local URL. From:

```
"url": "https://sqs.us-west-2.amazonaws.com/792379844846/DasInternal-dasinternalSqs-1HN6JX3NZPGNT",
```

to

```
"url": "http://192.168.1.224:8888/das",
```

So the configuration file will look like:

```
{
  "dasConfig" : {
    "url": "http://192.168.1.224:8888/das",
    "file_threshold_size": 100000,
    "flush_interval": 600,
    "storage_path": "/run/dasLogs",
```

```
    "storage_quota": 5000000,
    "backup_path": "/data/data/com.anki.victor/cache/dasLogs",
    "backup_quota": 10000000,
    "persistent_globals_path": "/data/data/com.anki.victor/persistent/dasGlobals.json",
    "transient_globals_path": "/run/dasGlobals.json"
  }
}
```

There is a second configuration that is tempting to modify. We won't need to. But lets look at it any way. The file path is:

```
/anki/data/assets/cozmo_resources/config/server_config.json
```

It has the following contents:

```
{
      "jdocs": "jdocs.api.anki.com:443",
      "tms": "token.api.anki.com:443",
      "chipper": "chipper.api.anki.com:443",
      "check": "conncheck.global.anki-services.com/ok",
      "logfiles": "s3://anki-device-logs-prod/victor",
      "appkey": "oDoa0quieSeir6goowai7f",
      "devappkey": "xiepae8Ach2eequiphee4U",
      "offboard_vision": "192.168.1.224:8888"
}
```

It looks tempting to change "logfiles" entry. We're going to bypass it completely.

10.2.2 Now add more scripts

Download the rcm-log-upload script and place it in the '/anki/bin/' directory on your Vector.

(We can improve the name later)

Edit this file. (We could use your help to use a proper .env file for configuration here) Look for the line:

```
: ${VIC_LOG_URL:="http://192.168.1.224:8888"}
```

Change the IP address and port number to the one your server uses.

Editing vic-log-uploader

Next step is to edit `/anki/bin/vic-log-uploader` so that it will use the modified uploader.

Change the line

```
UPLOADER="/anki/bin/vic-log-upload"
```

to

```
UPLOADER="/anki/bin/rcm-log-upload"
```

Editing vic-crashuploader.env

Now edit `/anki/etc/vic-crashuploader.env` so that the crash dump script will send the minidumps to your server.

Look for the line that starts with `VIC_CRASH_UPLOAD_URL`

```
VIC_CRASH_UPLOAD_URL='https://anki.sp.backtrace.io:6098/post?
format=minidump&token=6fd2bd053e8dd542ee97c05903b1ea068f090d37c7f6bbfa873c5f3b9c40b1d9'
```

And change that to your local server. For instance:

```
VIC_CRASH_UPLOAD_URL='http://192.168.1.224:8888/'
```

10.2.3 What about the server on my computer ?

We need help creating a python or node.js program to receive the variety of log and crash files. Contact Randy (randym@randym.name) if you can help, or for a reference C# program that works on Windows.... (it does require granting a lot of permissions tho')

10.2.4 Finally reboot, if you want

Vector won't use the new server addresses (in most cases) until you do a reboot

10.3 How to convert animation bin files to JSON

The animation binary files are based on Google's flatbuffers using a binary format. Forturnately it is easy to read, since Anki left the description file in the Vector software, and it is an evolution of what was used in Cozmo.

The files can be turned into JSON, and then back. Google's tools will do this for you, see "Using flatc as a JSON Conversion Tool"

You can also turn the JSON file back into a binary file using the same tool.

10.3.1 Developer Animation JSON files

The developer releases of Vector software includes animation JSON files. These are the equivalent to animation binaries, but in JSON format. The developer software -- and perhaps the production software as well -- can read the animation in the JSON form.

10.4 How to create a soundbank

Vector's audio engine does not directly use `mp3` or other common audio files. Vector uses AudioKinetic's WWise sound engine. This engine uses a proprietary set of file `.bnk` and `.wem` files. The result is a sophisticated mechanism to create audible responses, little physiological effects, and more.

Unfortunately it is not intuitive to add sounds. WWise is free for non-commercial use; but this is not an easy tool to use. That is why I've written up this how-to. It should make the process a lot more straight-forward.

I won't describe how to hook the new sounds into the animations or behavior tree here. That has its own multi-step process.

The sample WWise project that we create below can be downloaded here

10.4.1 Preparation

Tools You will need

You will need AudioKinetic WWise, version 2017.2. This version is the one that creates file with the same version id that Vector is expecting. (Other version may work, but the internet reports version mismatch is a very common cause of WWise errors.).

- Download and install AudioKinetic WWise by using teh WWise Launcher From there you will have to select this version.

- WWise 2017.2 documentation

The free version of WWise has a cap of 200 audio files; it is unlikely you will ever create that many in a single sound bank. If you do, just break the sound bank up to two or more.

Your sound files

You will need to convert your sound files to `WAV` files ahead of time. That is not described here.

10.4.2 WWise to create the soundbank

This section we'll describe how to build a "media only" soundbank that can be added to Vector.

Starting a Project

Launch WWise, from either your application menu or the WWise launcher. When WWise starts it will give you a window create a new project
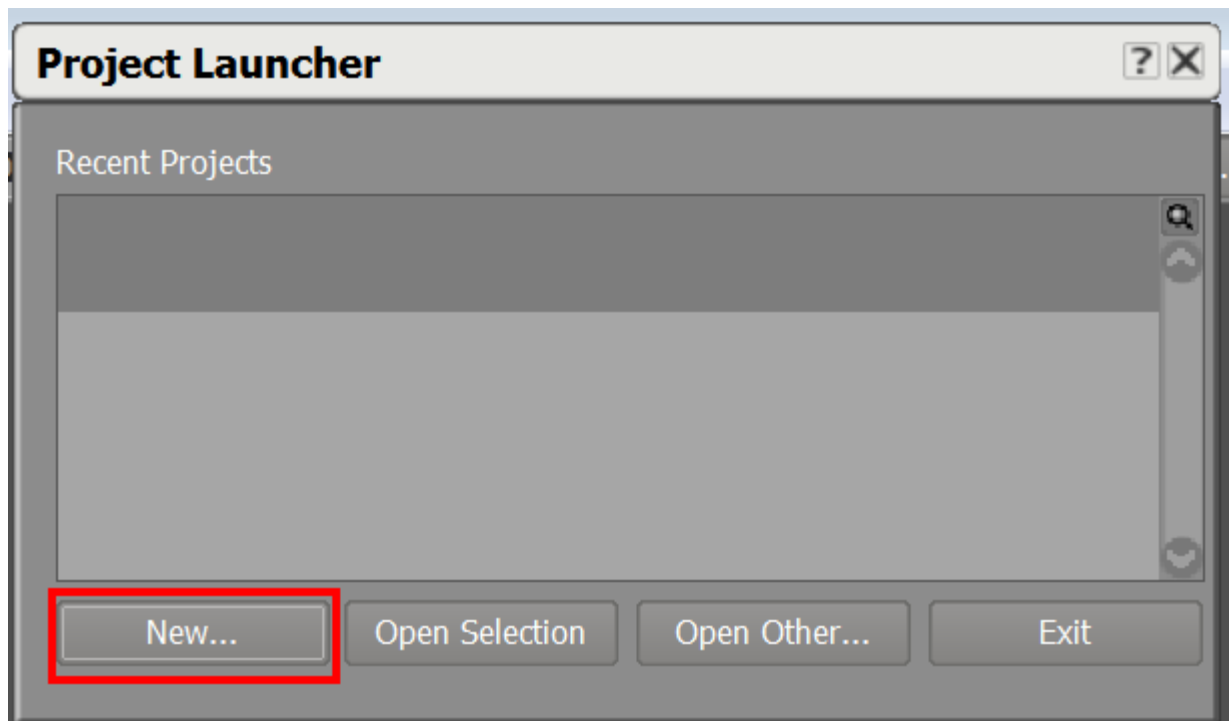
Figure: Starting WWise

Click "New"

Note: Along the way WWise may present pop-ups to let you know that you don't a have a license -- that this is only for non-commercial and evaluation use. Click ok whenever that happens.

After clicking new, it give a pop-up to do a little initial paperwork to create the project:
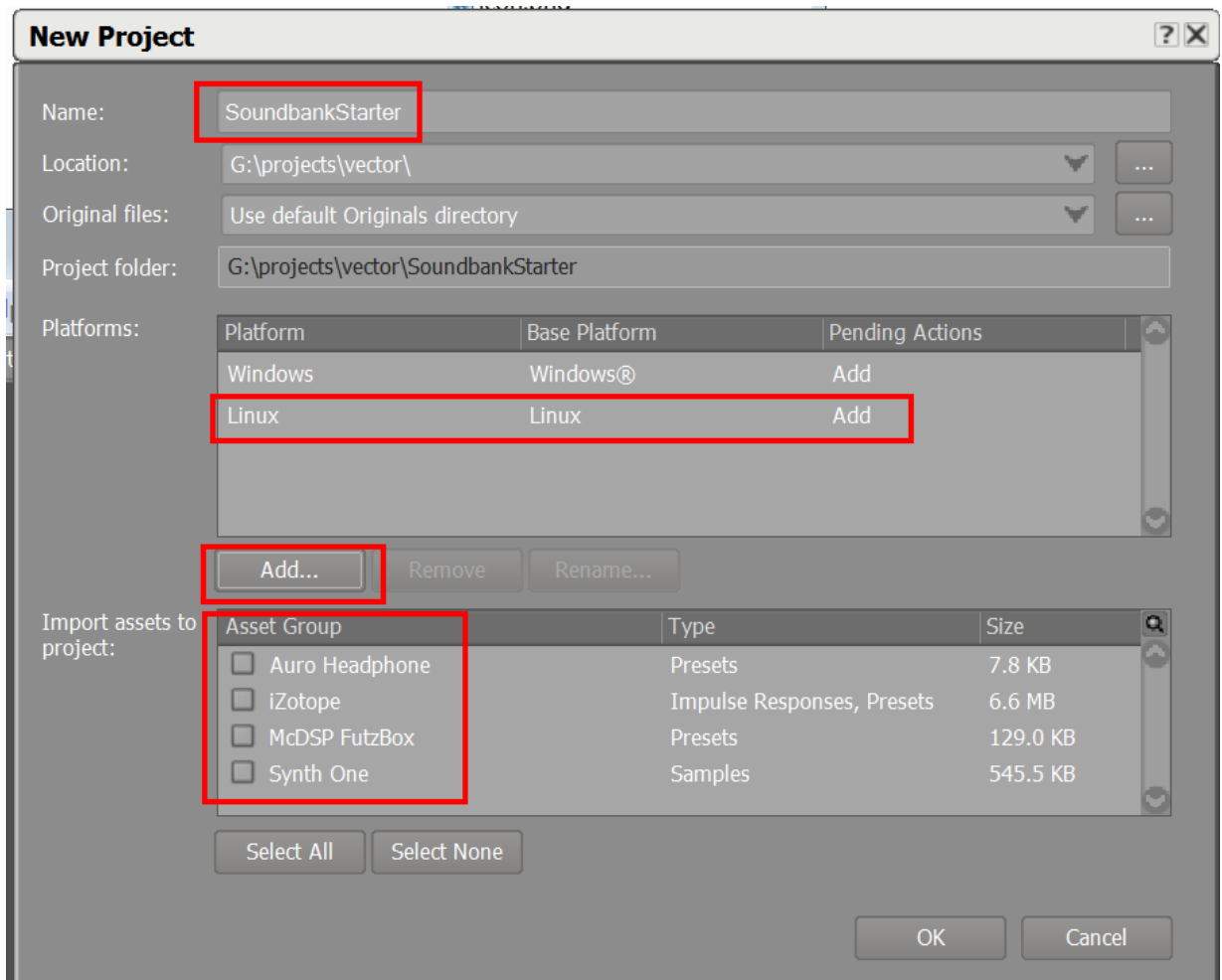
Figure: Create a new project

1. Fill in the name of the project with whatever you want. Be unique

2. Add the linux platform: click the "Add" and then select Linux. (I don't know that this strictly neccessary)

3. Uncheck all of the other assets groups.

4. Click "ok"

Creating a soundbank

The next step is to create our SoundBank. Look for the project pane:
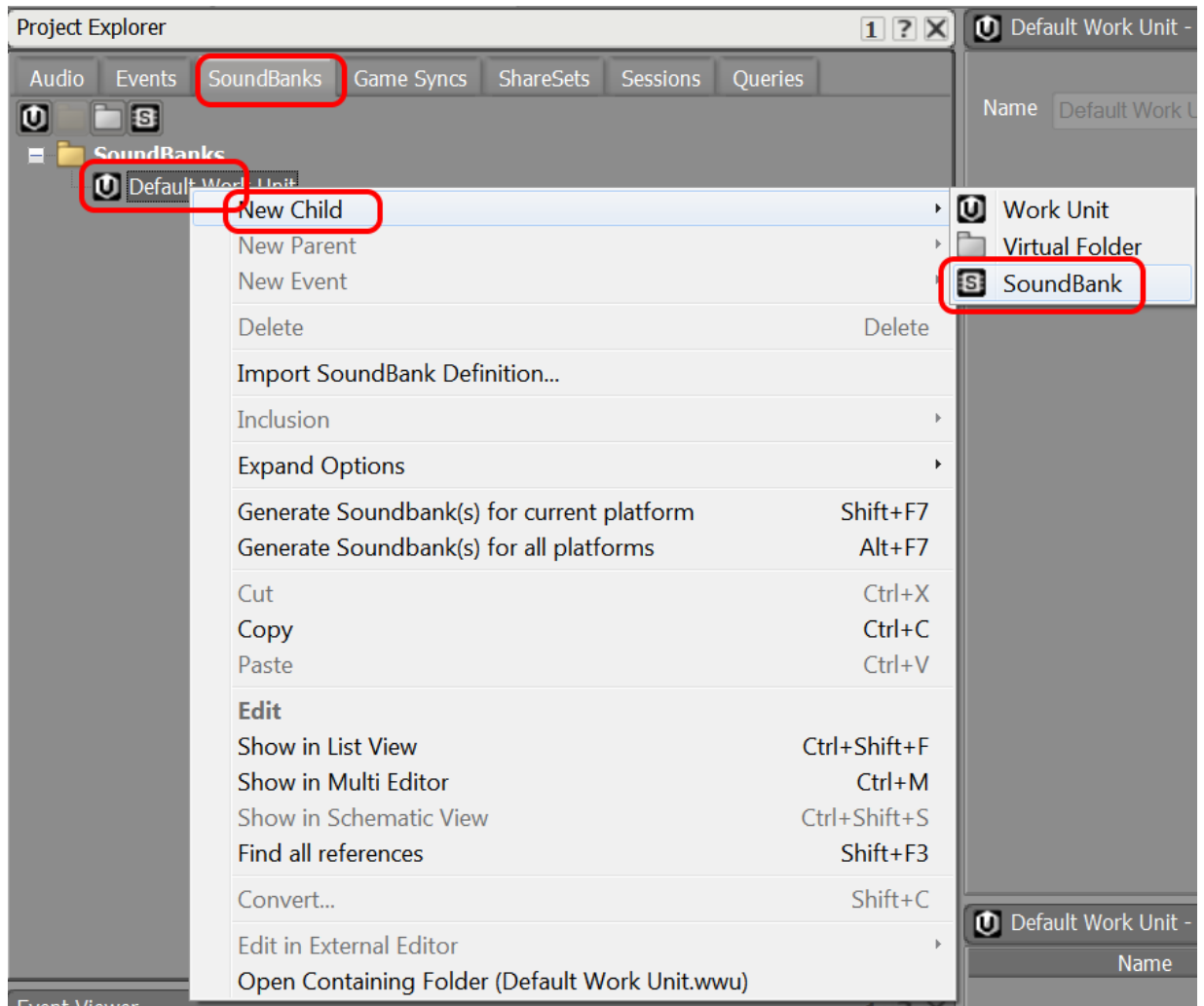
Figure: Create a new soundbank

1. Click on the "SoundBanks" tab

2. Right click on the "Default Work Unit" item under the "SoundBanks" tree

3. Go into the "New Child" submenu

4. Select "SoundBank"

This will create a panel to edit the sound bank. If not, double click on your new sound bank in the tree. You should get a SoundBank Editor:
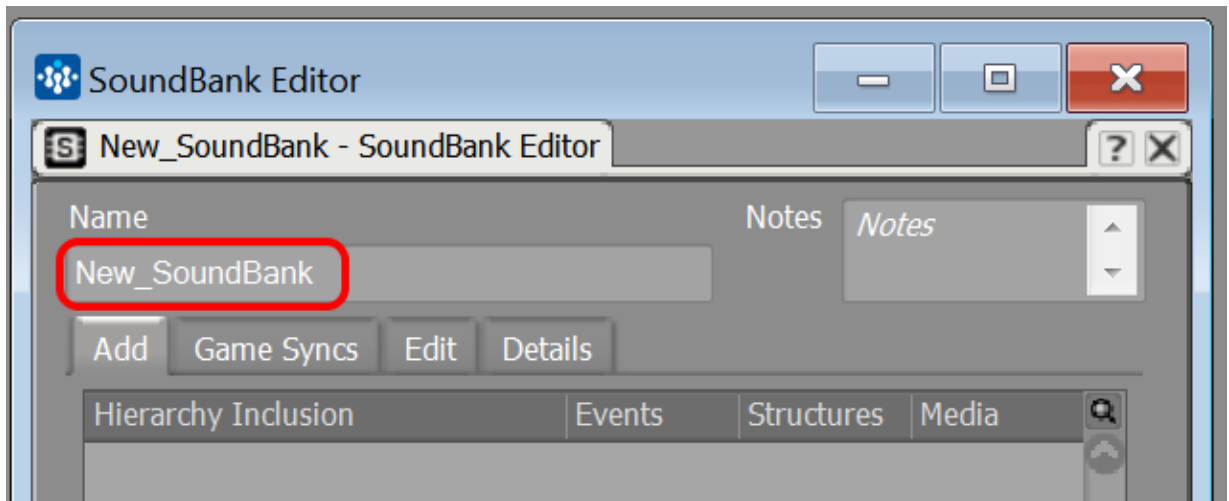
Figure: Naming a new sound bank

Give your sound bank a unique name. This will be the name on the robot, so make it descriptive, but unique.

Importing Audio files

Next is importing the audio files. You will need to convert your audio files into the `WAV` format, if you have not done so already.
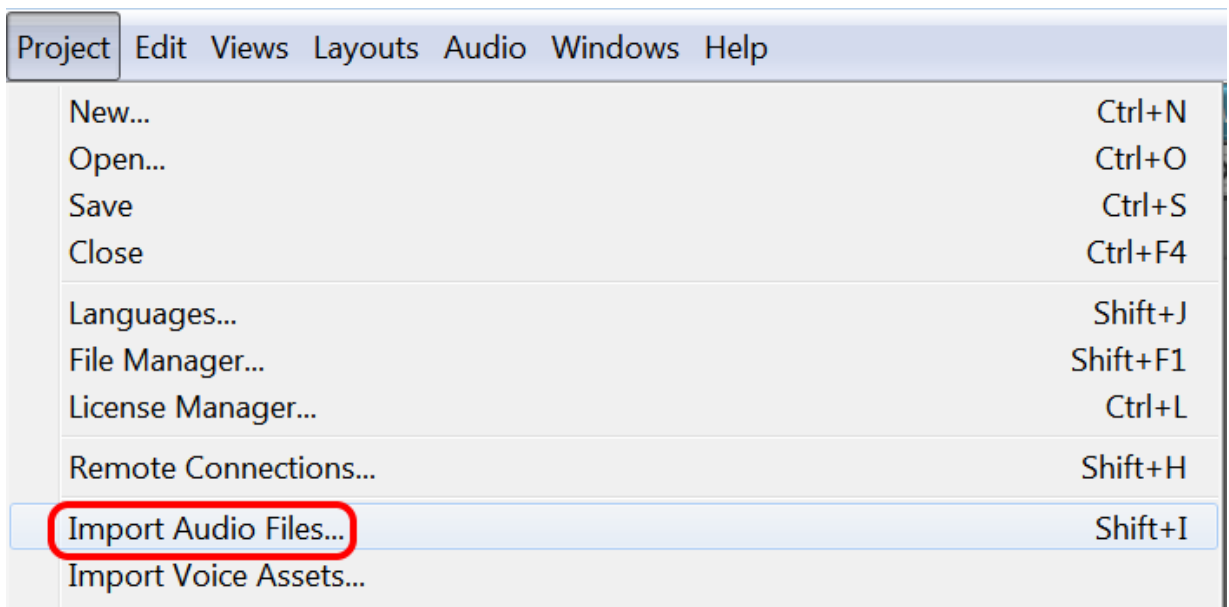
Figure: Import audio files

1. Open the Projects menu

2. Select "Import Audio files"

This will bring up a dialog:
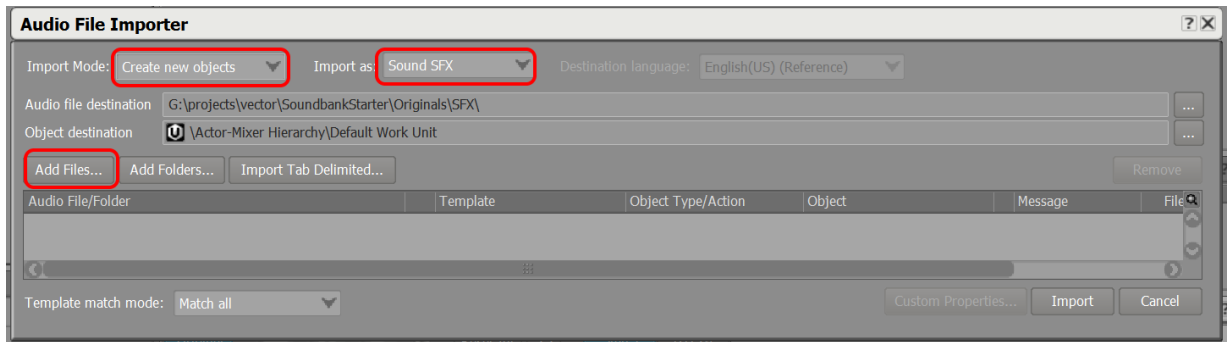
Figure: Add audio files

1. Change the "Import Mode" to "Create new objects"

2. Change the "Import as" to "Sound SFX"

3. Click "Add Files" button

4. Select the files you wish through the usual dialog

Click "Import"

Setting the file properties, ie, name, and conversion

Next, lets set the audio conversion. Find the audio file in the project tree:
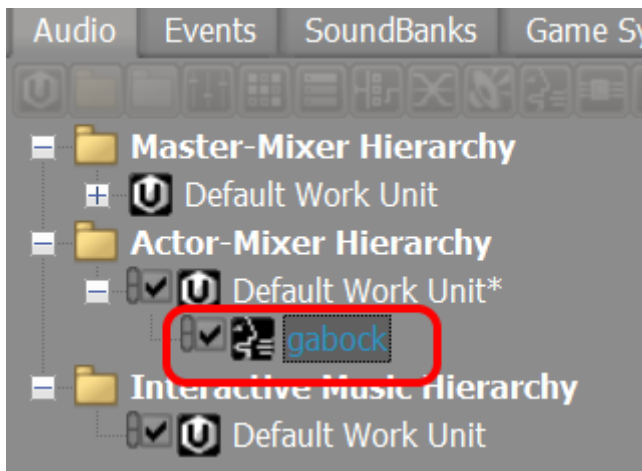
Figure: The audio files in the tree

Click on the file. It should open a panel on the right to configure the properties of this file:
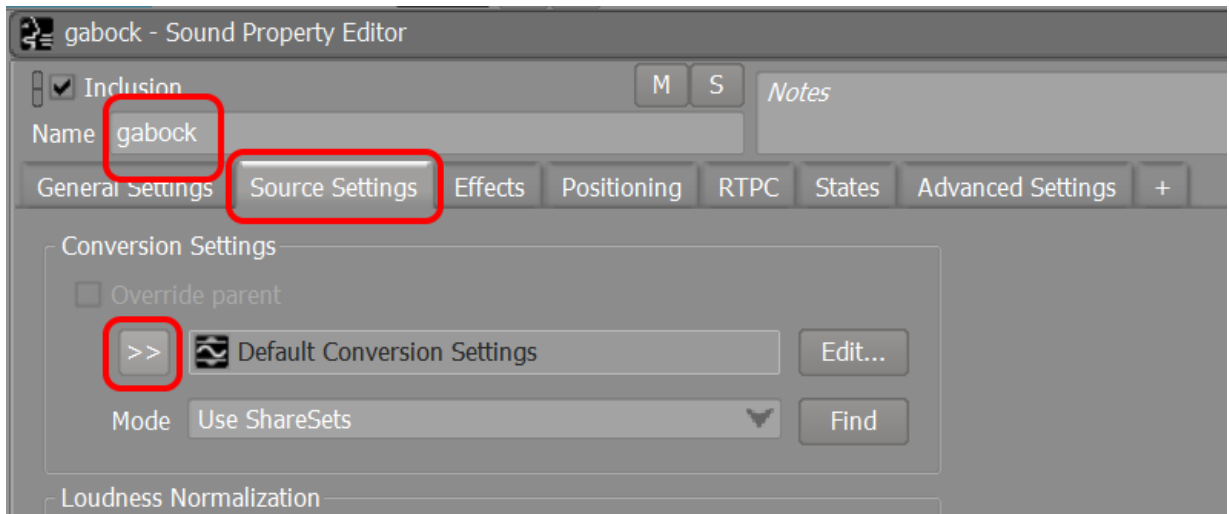
Figure: The audio file properties

1. Set the name of the file. Optional, but this is a good time to change the name of the file to have a nice consistent, clean name.

2. Click the "Source Settings" tab

3. Under than click the chevrons next to the word "Default Conversion Settings"
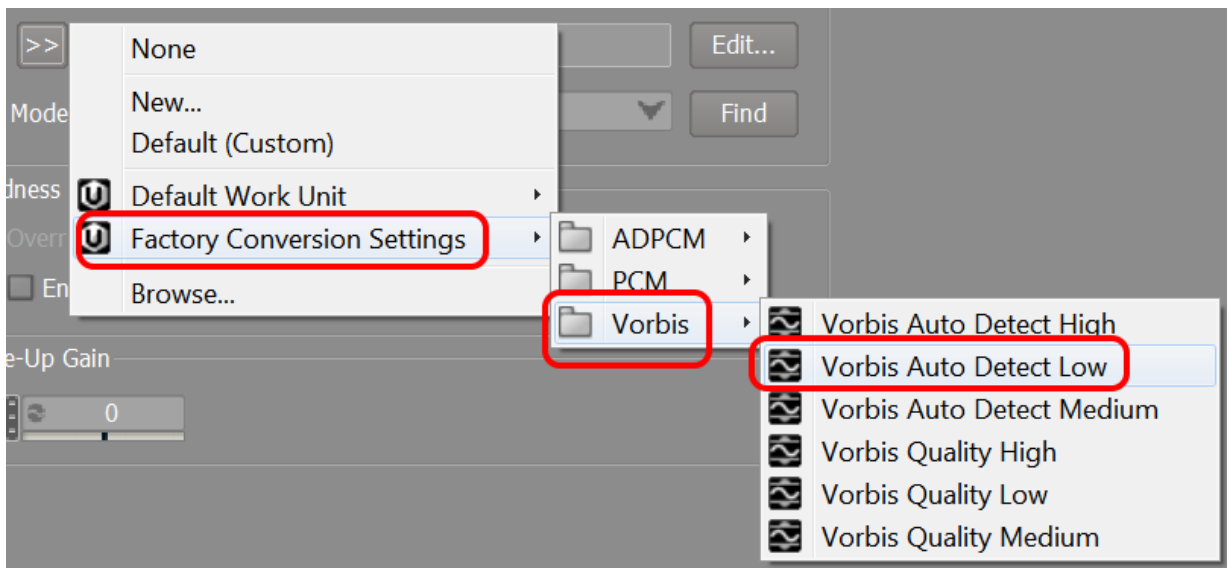
Figure: The conversion settings

Clicking on the chverons will bring up a menu

1. Click on "Factory Conversion Settngs"

2. Select "Vorbis"

3. Select one of the Vorbis formats. Probably any will do. (Probably any ADPCM will do to, but I haven't confirmed it. I don't have guidance for which to choose.)

Finally, we need tell it to create a separate WEM file.
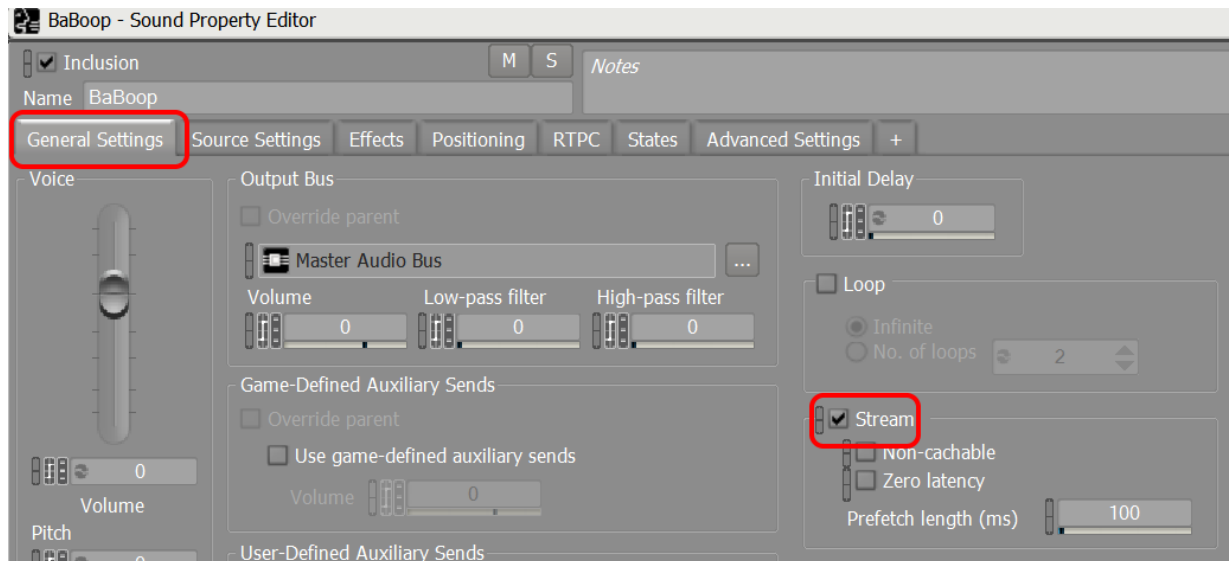
Figure: The stream setting

1. Click on the "General Settings" tab.

2. Check the "Stream" check box.

Create an event and action to play it

Next we need tell the audio engine that this is playable. To do that we need to create a Event with a play action. Click on the "Event" tab:
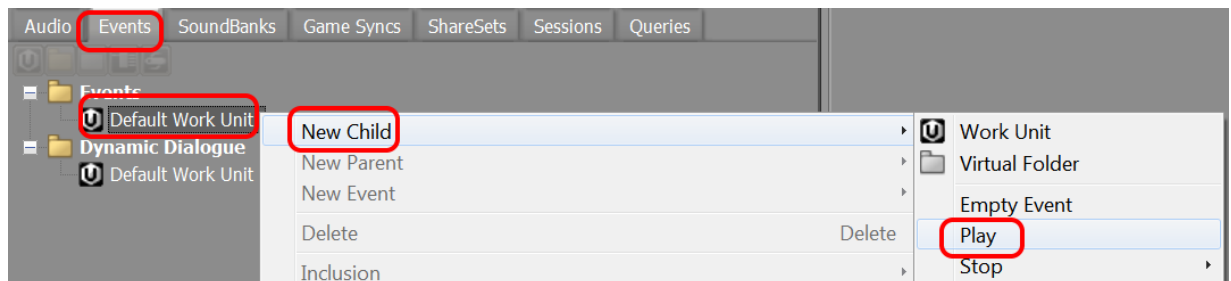
Figure: Creating a play event

1. Right click on the "Default Work Unit" This will bring up a menu

2. Select "New Child"

3. Then click on "Play"

This will add an item to the tree on the left under "Default Work Unit", in edit mode:
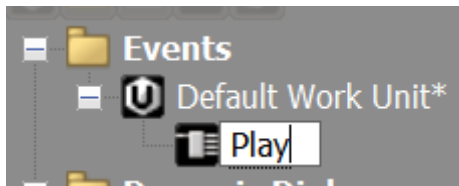
Figure: New a play event

Give the name something like "Play_*sound name*"

On the right side a panel should appear with the properties for this event. It doesn't know yet to play this particular sound. We'll connect that now.
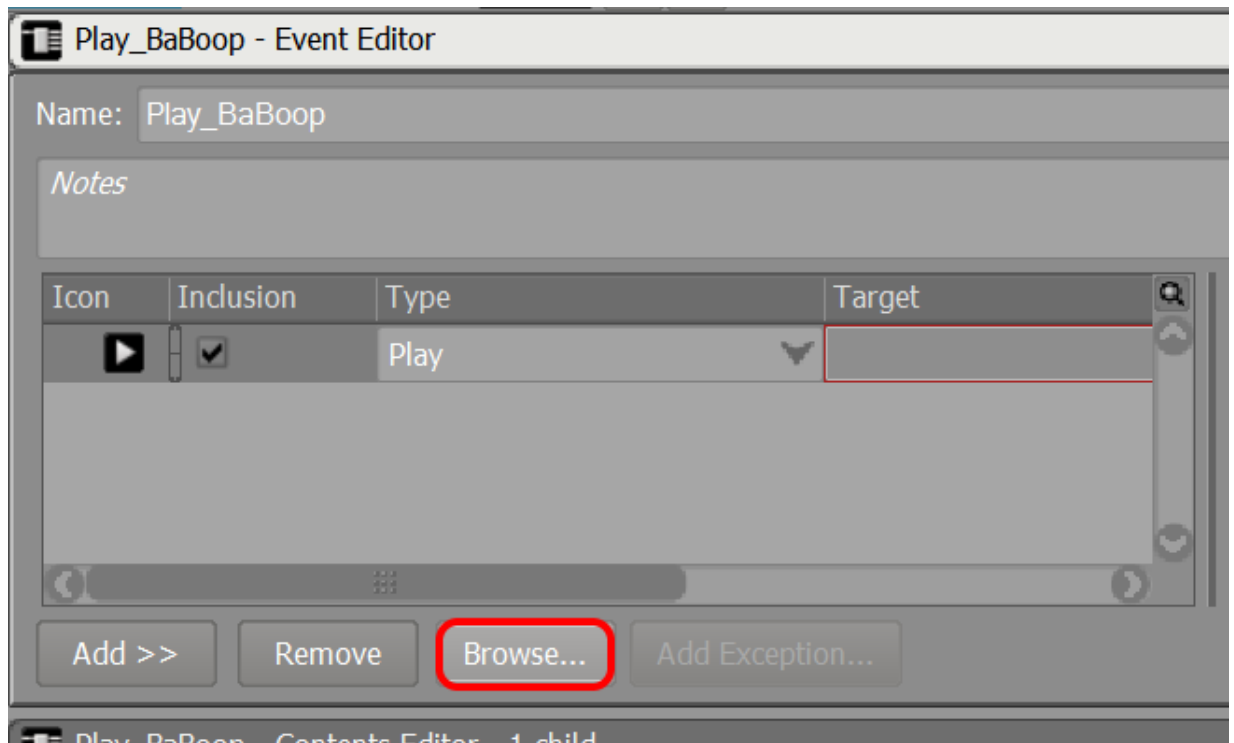
Figure: Attach a sound to the play event

Click on "Browse". This will pop up a window to select which sound.
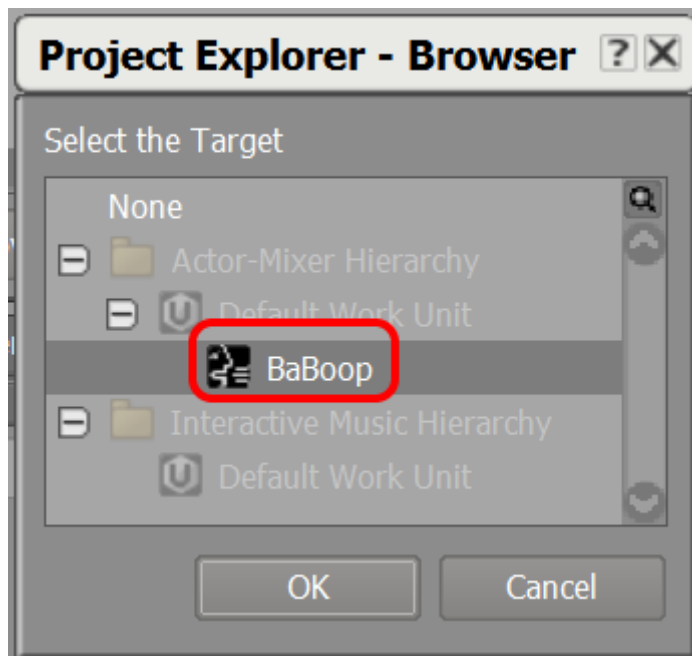
Figure: Selecting a sound to attach to the play event

We need to add it to the sound bank. If the sound bank panel isn't open, click on the SoundBanks tab, and double click on the sound bank.

Click on the Events tab and drag the new event into the soundbank:
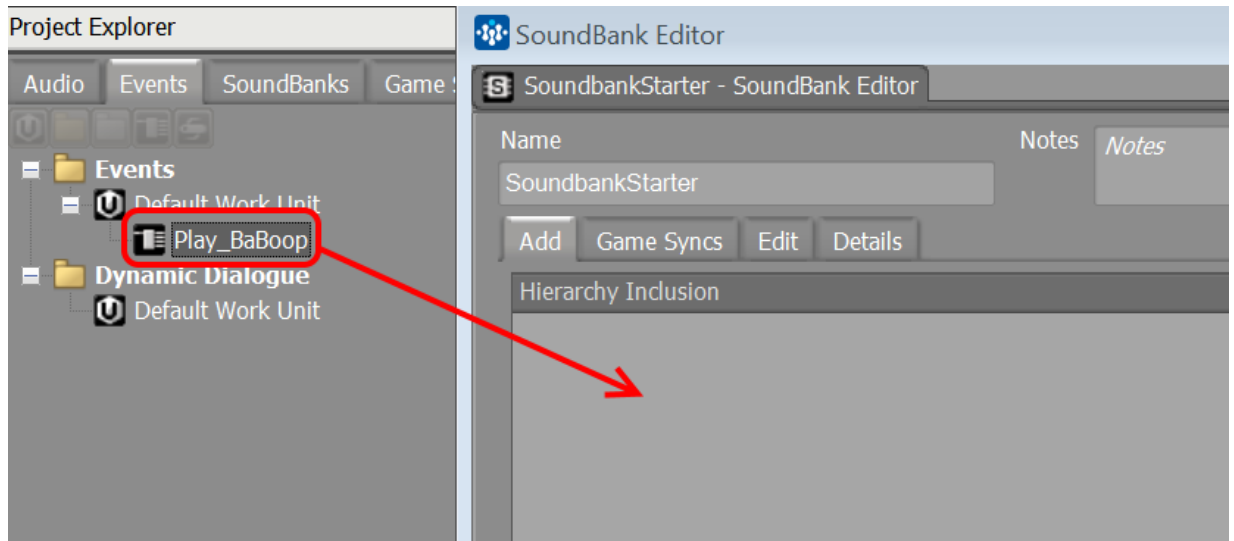
Figure: Adding the play event to the sound bank

You should then see:
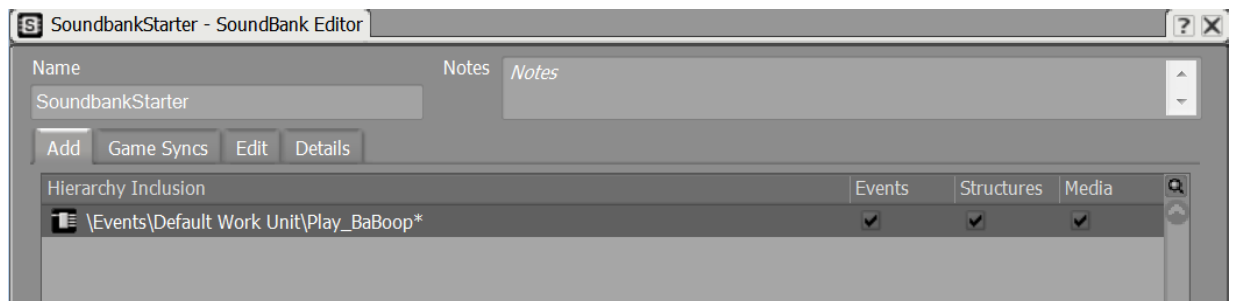
Figure: The play event in the sound bank

Click on the Audio tab, and drag the audio file into the soundbank as well:
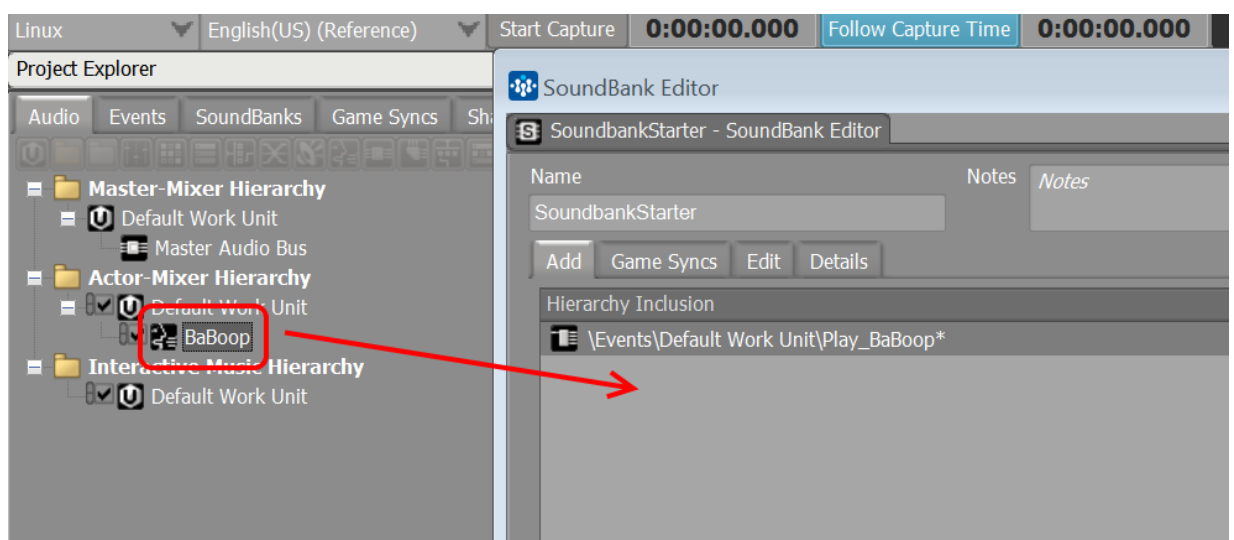
Figure: Adding the audio file to the sound bank

You should then see:
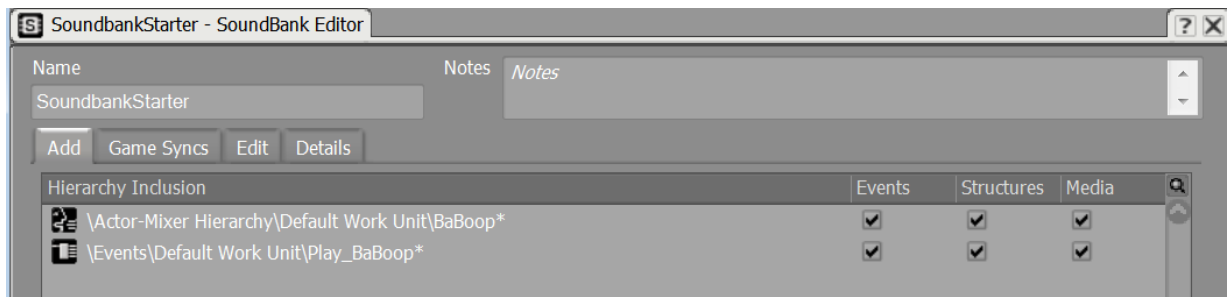
Figure: The play event and audio file in the sound bank

Generating the Soundbank

Now it is time to convert the files and generate the sound bank.
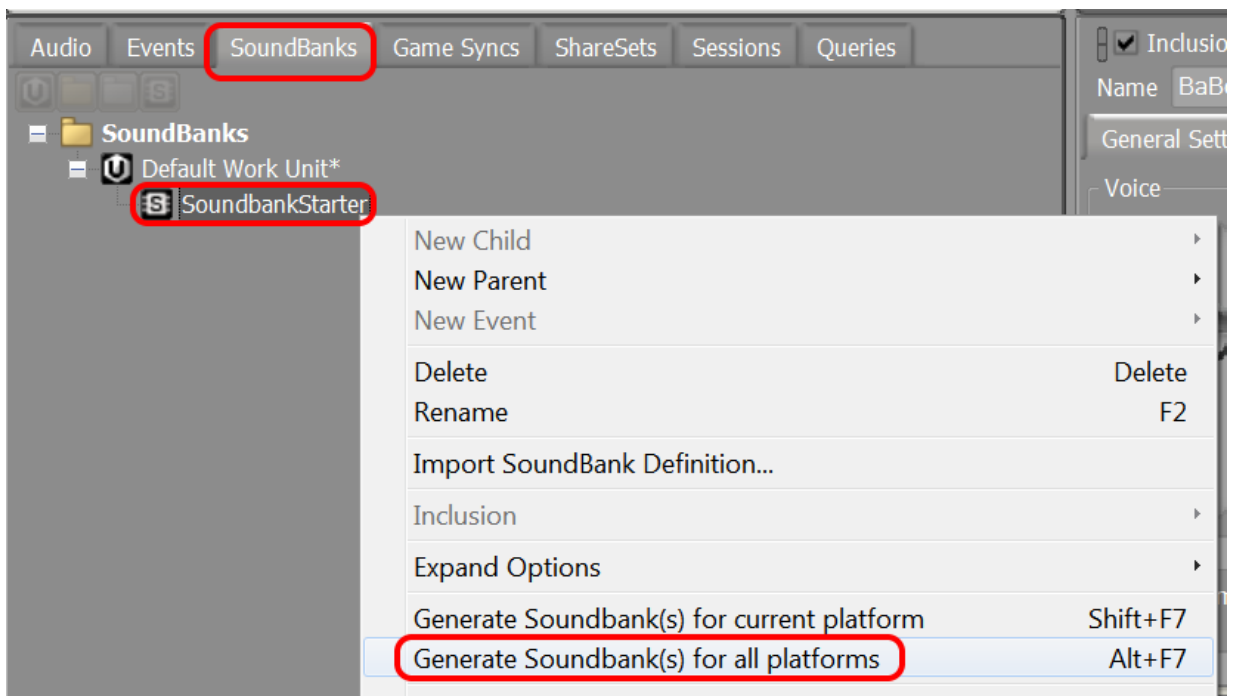
Figure: Generate the sound bank

1. Click on the SoundBanks tab.

2. Right click on our SoundBank. This will pop up a menu

3. Choose "Generate SoundBank(s) for all platforms"

WWise will convert all of the audio files and create a sound bank.

10.4.3 Packaging and Installations

The folder with the files

Lets look at the generated files now. Open the folder you created for your WWise project. It should look like:
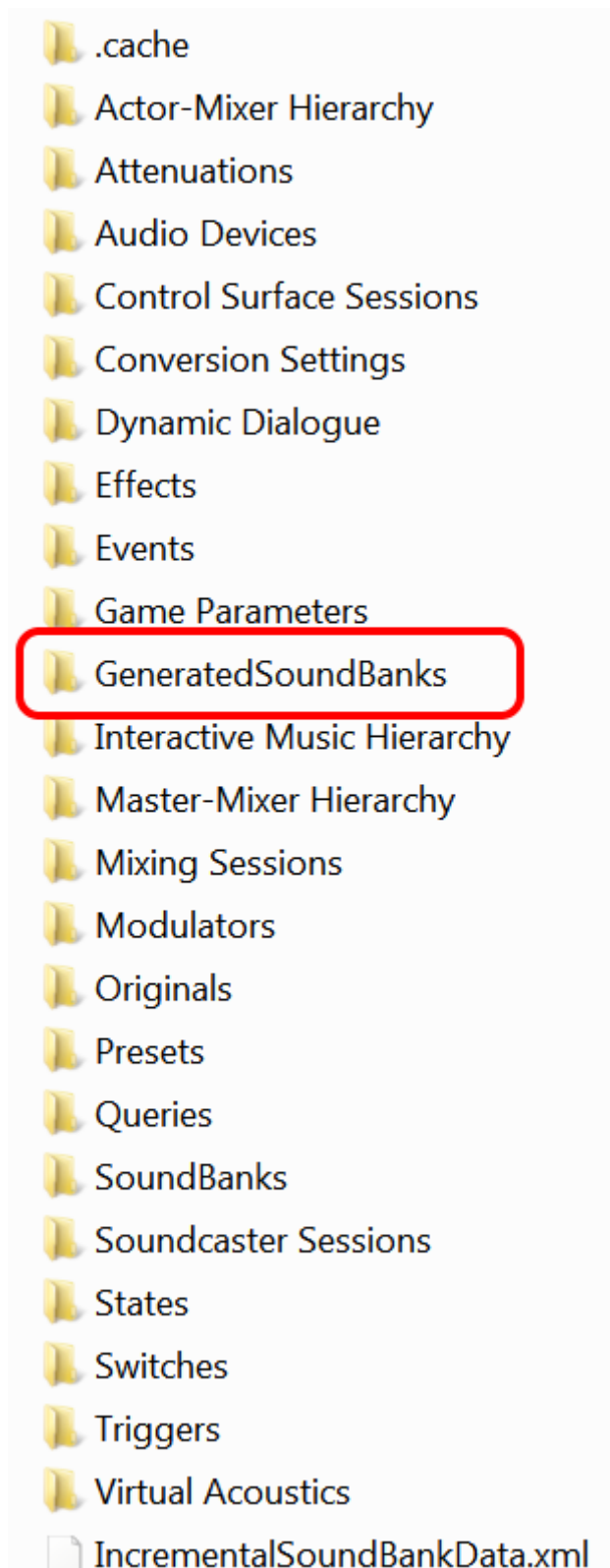
Figure: The WWise project folder

Open the "GeneratedSoundBanks" folder; inside of that open the "Linux" folder. You should see a folder like:
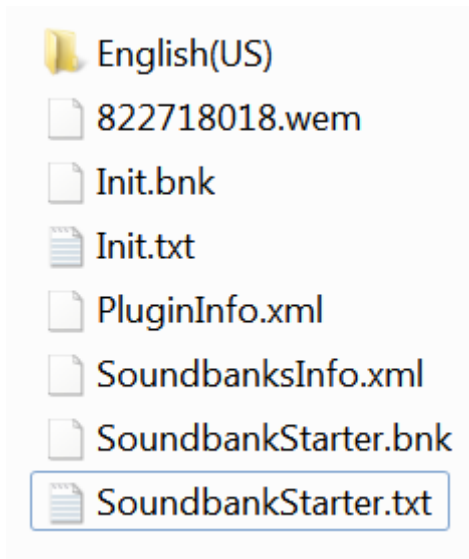
Figure: The soundbanks folder

(The numbers and names will be different.)

Make a note of the ".txt" file with the soundbank name. We will need that later.

Making a package

We need to get the .wem files and out .bnk file (ignore the Init.bnk) onto Vector and placed in the following folder:

```
/anki/data/assets/cozmo_resources/sound/
```

One way to do this is to make a `vpkg` . The `.ini` for the package file might look something like:

```
[META]
name=SoundbankStarter
rel_num=1
restart_type=maintenance-restart

[files]
/anki/data/assets/cozmo_resources/sound/SoundbankStarter.bnk=SoundbankStarter.bnk
/anki/data/assets/cozmo_resources/sound/SoundbankStarter.txt=SoundbankStarter.txt
/anki/data/assets/cozmo_resources/sound/822718018.wem=822718018.wem

[permissions]
/anki/data/assets/cozmo_resources/assets/cozmo_resources/sound=anki:anki 0444
```

Then create a vpkg, in this example called "SoundbankStarter":

```
vector-pkg.py install --pkg=SoundbankStarter.vpkg
```

You will then copy the file on the bot and then install it with a command line like

```
vector-pkg.py install --pkg=SoundbankStarter.vpkg
```

You will also need to edit a file so that Vector's audio engine knows to load it. This file is

```
/anki/data/assets/cozmo_resources/sound/SoundbankBundleInfo.json
```

By default in 1.7 it looks like:

```
[{"bundle_name": "Victor_Global_Data_English(US)", "language": "English(US)", "path": "English(US)/
Victor_Global_Data.bnk", "soundbank_name": "Victor_Global_Data"}, {"bundle_name": "Init", "language": "SFX", "path":
"Init.bnk", "soundbank_name": "Init"}, {"bundle_name": "Victor_UI", "language": "SFX", "path": "Victor_UI.bnk",
"soundbank_name": "Victor_UI"}, {"bundle_name": "Victor_VO", "language": "SFX", "path": "Victor_VO.bnk",
"soundbank_name": "Victor_VO"}, {"bundle_name": "Victor_Alexa", "language": "SFX", "path": "Victor_Alexa.bnk",
"soundbank_name": "Victor_Alexa"}, {"bundle_name": "Victor_SFX", "language": "SFX", "path": "Victor_SFX.bnk",
"soundbank_name": "Victor_SFX"}, {"bundle_name": "Victor_Dev_English(US)", "language": "English(US)", "path":
"English(US)/Victor_Dev.bnk", "soundbank_name": "Victor_Dev"}]
```

(Yes, this one long run-on line.)

We want to add a like that it, like so:

```
[{"bundle_name": "Victor_Global_Data_English(US)", "language": "English(US)", "path": "English(US)/
Victor_Global_Data.bnk", "soundbank_name": "Victor_Global_Data"}, {"bundle_name": "Init", "language": "SFX", "path":
"Init.bnk", "soundbank_name": "Init"}, {"bundle_name": "Victor_UI", "language": "SFX", "path": "Victor_UI.bnk",
"soundbank_name": "Victor_UI"}, {"bundle_name": "Victor_VO", "language": "SFX", "path": "Victor_VO.bnk",
"soundbank_name": "Victor_VO"}, {"bundle_name": "Victor_Alexa", "language": "SFX", "path": "Victor_Alexa.bnk",
"soundbank_name": "Victor_Alexa"}, {"bundle_name": "Victor_SFX", "language": "SFX", "path": "Victor_SFX.bnk",
"soundbank_name": "Victor_SFX"}, {"bundle_name": "Victor_Dev_English(US)", "language": "English(US)", "path":
"English(US)/Victor_Dev.bnk", "soundbank_name": "Victor_Dev"}
,{"bundle_name": "SoundbankStarter(US)", "language": "English(US)", "path": "SoundbankStarter.bnk",
"soundbank_name": "SoundbankStarter"}
]
```

Next, restart the Vector application by:

```
systemctl stop anki-robot.target
sleep 5
systemctl start anki-robot.target
```

Then we need to check the logs that the file loaded:

```
grep SoundBank /var/log/messages
```

This shouldn't show a problem. If you see something like the following, there was a problem.

```
12-09 04:40:43.725 warning vic-anim 2103 2103 vic-anim: AudioEngineController.LoadSoundbank: Failed to load
soundbank 'SoundbankStarter'
```

(The name of the soundbank being your soundbank.)

(Check that the time stamp is about "now" -- just so that we aren't confused with old errors)

10.4.4 Testing

When you wish to play the animation, lets open the text file we saw earlier. In the example case it was called "SoundbankStarter.txt". This file has the info we need to play the sound.

Figure: The soundbank event ids

The number is the event id to be used inside of animations. The name can be used in some JSON files; but it is als helpful when working with multiple sounds in the file, to know which one is the right event.

The sound file can be tested using the console vars. Please see Development Web Servers for how to set up access to these.

Go to http://localhost:8889/

Figure: Console vars button

Click on the "consolevars". This will bring up a pannel of tabs. Click on the "Audio" tab:

Figure: Audio Console vars

In the "PostAudioEvent" you can paste the Event Name (from your txt above). Then click "Call" This should cause your new sound to play. The Event ID will not work here.

10.4.5 Future improvements

It would be nice to be able to bundle the soundbank and files into a folder, so that it was clear which files belonged to which soundbank.

It's probably possible to script up the generation of the project files, and then just open in WWise command line to do the final steps...

10.5 How to add (or change) voice commands

This is an article describing how to add (or modify) a Voice command on the EscapePod. It focuses on how to craft a small grammar and convert it into rule(s) for EscapePod. (I hope to create a later article explaining in more detail how to create behaviors that exist on extensions to the Escape Pod.)

10.5.1 A glossary of terms

Let's define some terms first to be clear and consistent within the article:

Term	Definition
behavior	A structure on Vector to represent and manage a potentially complex task that might involve animations, changing his emotions, path planing, driving, and so on. It is also used on the EscapePod as a catchall for utterances and how they are made into an intent.
intent	An intent is a structure with an internal code that is used to represent the how to respond to the phrases spoken by a person. It may represent the action requested, an answer to a query, or an action that emotionally responds to what was said.
property name	The name of a key in the structure; also called a field.
structure	A table of property names (aka field, or key) and the value associated with it.
utterance	What a person said, and in the context of this article, the transcription to text of what the person said.

10.5.2 Planning

I'm going to focus on the idea of a new voice command, to give an idea of the overall process. After all, modifying an existing voice command is easier, just a matter of winging it. This may give some idea how to extend existing commands.

We'll use 5 steps to create a voice command on the EscapePod:

1. Decide what you want Vector to do when he hears the voice command

2. Make a list of what you want to say

3. *Optional (or in rare cases)* what extra information that Vector needs to know

4. Create a helper table, in preparation for entering it into the EscapePod

5. Plonk this into the EscapePod; the EscapePod UI is still a bit new and very techy-focused in this area, so I'll try to explain what some of the fields do.

The first step is to decide what you want Vector to do, from the list of supported cloud intents. (For now let's ignore EscapePod extensions.) It has to be one that Vector recognizes. (See here for a table of the intents Vector recognizes.)

For demonstration purposes I'm going to use **intent_imperative_eyecolor_extend** (and pretend that it does not already have a voice command.)

Making Table of What you can say

Making a list of what you want to say to Vector is the second easiest part. Write the items down in a bullet-pointed list. For the set eye color intent, the stock list of phrases that Vector recognizes include:

- change eyes

- change eye color

- make your eyes

Tip: stick with one to two words, occasionally three; try getting rid of common articles (a, the, his), determiner (your) and other common words. Go for "google whacks" — where the word or word pair is not used in any other voice command.

Following this tip, we might change the last item to:

- make your eyes

See if Vector needs extra information in the intent and creating a helper table

Next, look up in the cloud intents page to find if the intent needs extra information.

There are three possiblities here

1. The cloud intent doesn't need any extra information (most cases) If this is the case, skip to the next section.

2. The cloud intent has a single property that it needs a value for; this property have a name, and fixed set of values that it accepts. (This happens for a handful of intents.)

3. The cloud intent takes a single property, but the value can be anything. This is used with the intent to teach Vector your name.

Following the eye color example, it takes one property, **eye_color**, that says which color to shade the eyes. This property only accepts very specific values. Lets create a table of the property values to keep it organized.

Property Value	Spoken phrases
COLOR_BLUE	
COLOR_GREEN	
COLOR_ORANGE	
COLOR_PURPLE	
COLOR_TEAL	
COLOR_YELLOW	

What we need to do now is add the phrases that go with each possible value:

Property Value	Spoken phrases
COLOR_BLUE	blue, azure, sapphire
COLOR_GREEN	green, lime

etc.

Tip: like earlier stick with one to two words, occasionally three; try getting rid of common articles (a, the, his), determiner (your) and other common words. And go for"google whacks" — where the word or word pair is only used here for a color.

10.5.3 Entering this into the EscapePod UI

Now we're ready. Let's begin entering this into the EscapePod now.

1. First, Click on the menu in the upper right hand corner, and select "Behaviors":

Figure: Behaviors item on menu

2. Next, Click on the pull down menu and slect "Add A Behavior"
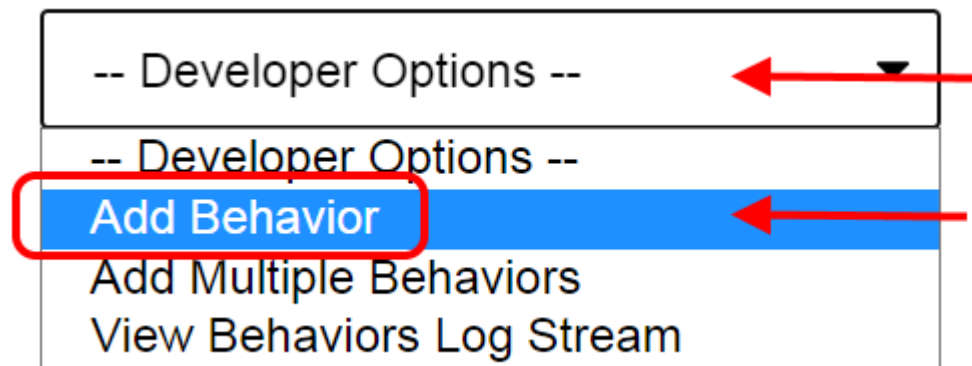
Figure: Add Behavior item on menu

3. A popup will appear giving dire warings. Just click ok.

escapepod.local:8443 says

Action: Add Behavior

WARNING: You should only edit these fields if you are a developer that
is editing/extending an existing behavior or one that you have created
with OSKR. It will not be possible to easily restore this data if lost. Click
OK if you are SURE you want to do this. Click Cancel to go back.

OK   Cancel

Figure: Add Behavior item on menu

4. Next you'll a place to start entering the information

Create Behavior

☐ Check to enable developer options

Name

Description
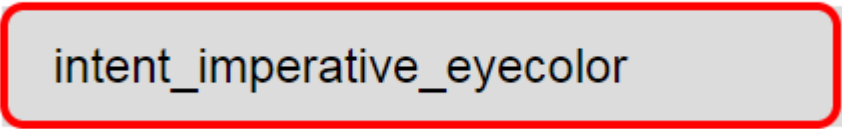
Figure: Starting to create a voice command

Leave the checkbox clear for now. Give it a nice name (it doesn't matter, and a helpful description (again it won't affect
anything).

Entering the Intent name and trigger phrases

Next fill in the intent name -- **intent_imperative_eyecolor_extend** -- in the field below the word "Behavior"

Behavior
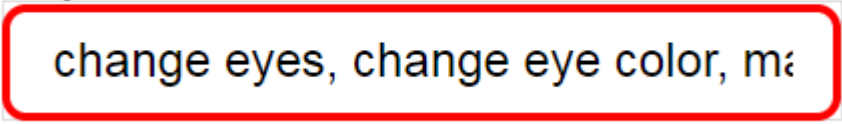
intent_imperative_eyecolor

Figure: Enter the intent name

Then take the list of words and phrases were made earlier -- e.g. "change eyes" -- and combine them using commas to separate the phrases. If the intent has extra properties (like this one does), then add all of the words and phrases for the property values from the table made earlier; separate with commas too.

Take this big, long list of words, and put it into the field below "Key Words". (The field is small so I recommend combining the words into a list in your favorite text editor, then copy-pasting it to the field.)

Key Words

change eyes, change eye color, ma

Figure: Enter the key words

If you don't have any extra properties, then just click "Save" and you're done. If you do have properties, then we need to go to the next step:

Entering the property names, and their phrases

To enter in the property names and their key words, we have to leap thru a few extra hoops:

1. Check the box for "Check to enable developer options"

Create Behavior

☐ Check to enable developer options

Figure: Click it to enable devleper options

2. Another popup will appear giving more dire warings. Just click ok.

escapepod.local:8443 says

WARNING: You should only edit these fields if you are a developer that is editing/extending an existing behavior or one that you have created with OSKR. It will not be possible to easily restore this data if lost. Click OK if you are SURE you want to do this. Click Cancel to go back.

OK   Cancel

Figure: Enabling developer options is... scary?

3. Click "Add Extended Key"

Extended Options

Block List

➕ ADD EXTENDED KEY

Figure: Time to add the property names and values

4. Enter the property name in the "Extended Key" entry.

Figure: Enter the property name

Now let's go back to the table of property values and their phrases that we made earlier. For each row in the table:

1. In the "Key Phrase" field, enter the property name (left column of the row in the table we made):

Figure: Enter the property value

For example:

Figure: An example the property value

2. Add in the phrase that indicate this value . Unlike the main list of phrases, we have to enter each of the phrases in here separately. Enter the first key word or phrase into the "Parameter" field.
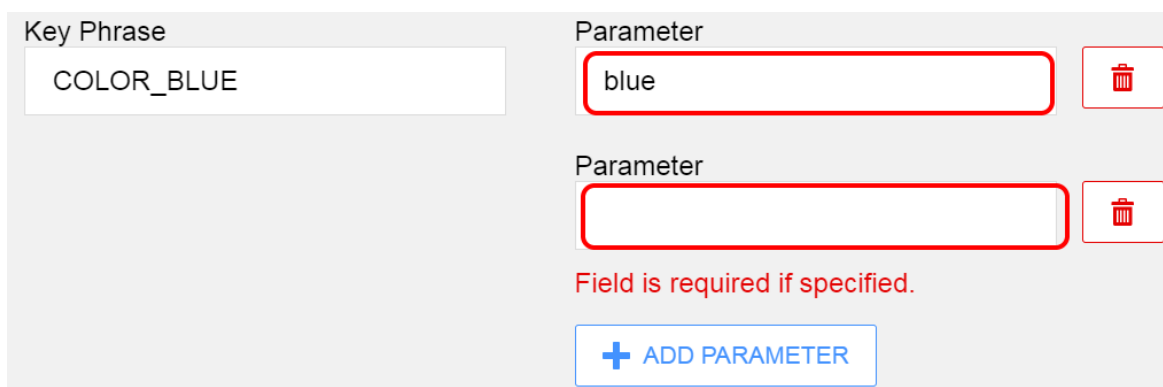
Figure: One property value key phrase at time

Then click "Add Parameter" and repeat for the rest of the phrases for this property.

3. If there is are more rows in the table, click "Add Key Phrase"

Once you're done entering in the table, click "Save."

10.5.4 Advanced Properties: Wild cards

Now that we've gotten used to the property name and values for intents, we can go on to an advanced case. Phrases can also capture whatever the person actually said and send that text for the property value.

This is used in only one intent at present: Teaching Vector your name. Let's look at it. The list of phrases that trigger the intent are like any other intent:

- my name is

- call me

- you may call me

- please call me

In this example **username** is the property name and will be filled in with whatever the persons says after the phrases above.

Entering this into the EscapePod UI

Here;s how to do it. Go thru all of steps before. And stop before clicking on "Add Extended Key".

1. Enter the property name in the field called "Extended Key" under "Parser":

Parser

Extended Key

Figure: Where the Wild Card property names grow.. erm, go

For example:

Extended Key

username

_Figure: Example of the **usename** wild card property name _

2. Then add in `[INTENT_INVERSE]` to the "Parser Target" field:

Parser Target

[INTENT_INVERSE]

Figure: Magic field values!

Then click "Save" and you're done.

10.5.5 Linking with the EscapePod

A voice command can be tagged to be sent to an EscapePod extension for further processing. The steps are the same as the above, except modify these to work with EscapePod extensions:

1. Make up a fake intent name for the intent for your EscapePod extension to key off of.

2. Under the "Parser" section, check the "Enable External Parsing" option.

Parser

Extended Key

Parser Target

☑ Enable External Parsing

Figure: Check this to forward it to an EscapePod extension

3. Click "Add Response Parameter" :

Response Parameters

➕ ADD RESPONSE PARAMETER

Figure: Check this to forward it to an EscapePod extension

4. Enter "final_intent" into the "Parameter Key" field, and the name of the intent to use as a back up if the EscapePod can't contact the extension or doesn't get a result. **intent_play_cantdo** is recommended, but you can use whatever you want.

Parameter Key	Parameter Value
final_intent	intent_play_cantdo

Figure: The intent name to use if the EscapePod extension doesn't work out

Then click "Save" and you're done.

10.5.6 Follow up

Note: someone can copy the table of intents to the forums wiki

Note: You can, in principal, have more than one property in your phrase pattern. Including combining an enumerated property and a wild card, but I have not characterized how well tested that works or a use case where it'd feel natura.

I recommend changing the following in the EscapePod UI:

- "Extended Key" to "Parameter Name" (or at least something less confusing)

- ("Wildcard Key" is also a parameter, so renaming to "Wildcard Parameter Name" at the same time ould be more consistent)

- "Key Phrase" isn't a phrase, it's a very programmer specific thing. I recommend that it be changed to "Property Value"

- ("Parameter" isn't a parameter at all, its key words or other utterance. Recommend changing it to "Key words"

- "Behavior" to a more correct term. Leaving it as is will cause lots of confusion, frustration and hard to help people who being to work with the behavior tree and we have to spend half a dozen messages establishing which ambiguous thing they are working on.

10.6 How to make a companion cube

10.6.1 Make one with paper or cardstock

heypapaya on discord shared a printable template that you print, cut and fold a cube with. An archive copy of the template is here as well.

10.6.2 3D Print your own cube

Anki Vector Dummy Cube Box by Dauler. This also includes a PDF with the symbols for the cube sides

You can buy STL files from etsy.

10.6.3 Emulate the cubes electronics

[Efforts to create a "clone" of the cubes electronics.] (https://forums.anki.com/t/communicating-with-vectors-cube/43042) SparkFun Pro nRF52840 Mini bluetooth development

10.7 How to re authenticate SDK apps

10.7.1 Introduction & Overvioew

With the EscapePod beta, for purposes of scope control, some robot function had to be tabled until later. One of those was the HTTPS SDK. If you're robot was already authenticated with your SDK (that is, you had the certificates and token needed) *and* you did **not** clear the user data to start using the EscpadePod, then you're fine. You don't need this.

If you haven't authenticated with the HTTPS SDK since you installed OSKR, or you cleared your data as part of setting up the EscapePod,.. or are just curious, this app note is for you.

This is how to get the certificate and SDK API guid so that your python SDK apps, .NET SDK Apps, including Vector Explorer, can get work again with the EscapePod.

Just a bit of fair warning: this process does still use the old Anki servers for one step. As mentioned at the top, the SDK sign-in / authentication is will change in the future as the EscapePod matures.

You will need to know the robot name, serial number, and IP address.

The steps:

1. Get the certificate from the robot

2. Get the GUID

3. Update the sdk_config.ini with these

10.7.2 Getting the certificate from the robot

The API .cert (certificate) file is actually on the robot in the `/data/vic-gateway/` folder. If you read the python SDK source, the SDK gets the certificate from the server. During startup, the robot checks for the existence of a certificate. After a "clear user data", there won't one. So the robot creates one. And, when it next talks with the cloud (such as part of the onboarding steps to get an account linked to the robot), the certificate is uploaded to the server.

You can copy from your robot `/data/vic-gateway/gateway.cert` file to your `~/.anki_vector` folder. You will need to rename it with the robot name and serial number, with a pattern like:

```
Vector-A1B2-007067cd.cert
```

I did it in one swoop with scp:

```
scp root@192.168.1.123:/data/vic-gateway/gateway.cert ~/.anki_vector/Vector-A1B2-007067cd.cert
```

(The robot performs SDK calls without the certificate, although I'm not sure if the SDK will have an error without out the certificate. If you're curious what the cert is used for: it is used so that you can be sure that you're talking to the real robot, not some sneaky imposter.)

10.7.3 Getting a new GUID token

Next we need to get a "guid" for communication. This is a succinct token that Vector uses to know that the API commands are coming from someone authorized to use the robot.

To get this guid token, run the attached python script. You run it with a command line like:

```
python robot.py 192.168.1.123 guid
```

Changing the ip address, to the one for your robot.

You will be asked to give your email address and password pair that your anki account is in. It will print out the guid after that.

For example:

```
>python robot.py 192.168.1.123 guid
Enter your email and password. Make sure to use the same account that was used to set up your Vector.
Enter Email: someone@someplace.org
Password:
ZLIO/y48QzeXynjiORrxgQ==
```

That funny text on the line below the password is the guid that we need to complete the .ini file.

The script passes username, and password to the old Anki server. The server knows how to validate your account, and what the shared secret is with the robot. And then sends back information that is used to create the guid.

Vector doesn't know you're account or password. The Vector serial name isn't passed to the old Anki servers when the guid is made.

10.7.4 Putting this all together in the sdk_config.ini file

Now let's, edit the '~/.anki_vector/sdk_config.ini'

Open it up. First look to see if there is a section already with your robots serial number, you will need to remove it.

Next, lets create a new section for this robot. The section name is the serial number of the robot. The section will look like this when we're done:

```
[007067cd]
cert = /Users/JoeUser/.anki_vector/Vector-A1B2-007067cd.cert
ip = 192.168.1.123
name = Vector-A1B2
guid = ZLIO/y48QzeXynjiORrxgQ==
```

Change `007067cd` to the serial number of your robot Give it your robot's name.

```
name = Vector-A1B2
```

And put in the IP address for your robot:

```
ip = 192.168.1.124
```

Next, in the new section update the path to the cert file that you downloaded in step 1:

```
cert = /Users/JoeUser/.anki_vector/Vector-A1B2-007067cd.cert
```

Finally, add an entry for the guid that we received in the second section:

```
guid = ZLIO/y48QzeXynjiORrxgQ==
```

10.7.5 Final steps

After that, you should be able to use Vector Explorer, or any SDK program with your robot. If you run into trouble, double check:

- the path and name of teh certificate file,

- the guid

- the robot IP address, and (of course)

- the robot name

10.7.6 Resources

Attached is the script created by Mike Corlett that we use to get the guid. (It'd be cool if someone updated the script to do all the work so a human didn't have to any .ini files.)

https://gist.github.com/randym32/16bde0ce2dda841336e3f9a250cca009 1

10.8 How to bring back the Snowglobe effect

It came up in the forums that Vector no longer played the SnowGlobe effect when shaken. Here is how to re-enable it.

10.8.1 Preparation

You'll have to know how to SSH in, make the file system modifiable and edit a file. To make the file system modifiable:

```
mount -o rw,remount /
```

You will need to edit the following file:

```
/anki/data/assets/cozmo_resources/config/engine/behaviorComponent/behaviors/victorBehaviorTree/
globalInterruptions.json
```

first, make it write able (you can skip this if you know how to override it in vi)

```
chmod +w /anki/data/assets/cozmo_resources/config/engine/behaviorComponent/behaviors/victorBehaviorTree/
globalInterruptions.json
```

10.8.2 Edit the top list of behaviors

Next edit the file:

```
vi /anki/data/assets/cozmo_resources/config/engine/behaviorComponent/behaviors/victorBehaviorTree/
globalInterruptions.json
```

Look for the lines

```
    "WeatherResponses",
    "TakeAPhotoCoordinator",
    "ReactToRobotShaken",
    "ReactToTouchPetting",
```

Change the line

```
  "ReactToRobotShaken",
```

to

```
  "ReactToRobotShakenSnowGlobe",
```

You can also leave both. The first item has higher priority.

10.8.3 Adjusting the shake threshold

you can tweak the threshold for the shaking:

```
/anki/data/assets/cozmo_resources/config/engine/behaviorComponent/behaviors/victorBehaviorTree/reactions/
reactToRobotShakenSnowGlobe.json
```

Look for the block

```
"wantsToBeActivatedCondition":
{
  "conditionType" : "RobotShaken",
  "minAccelMagnitudeThreshold" : 16000
}
```

Change the "16000" a lower or higher number for the threshold.

The robot shaken file has a similar config:

```
/anki/data/assets/cozmo_resources/config/engine/behaviorComponent/behaviors/victorBehaviorTree/reactions/
reactToRobotShaken.json
```

Look for the same block as above, and change the threshold.

If you leave both "ReactToRobotShaken" and "ReactToRobotShakenSnowGlobe", have the first item with a higher number. If it is lower, it will always win.

10.8.4 Reboot

Finally you have to restart the vic applications for the updates to load and take effect. This can be done with:

```
systemctl stop anki-robot.target
systemctl start anki-robot.target
```

or a reboot.

10.9 How to set up a new Yocto-linux build environment for Vector's base OS

This is how to create a new build environment for Vector's base OS -- Yocto Linux and his drivers. *Note: this does not include the Vector application software!*

Steps.

1. Install Ubuntu (or reuse a machine with Ubuntu)
2. Get Yocto installed, e.g. using Docker
3. Install the base OS source code
4. Test build

10.9.1 Install Ubuntu

Your options are:

1. You already use Ubuntu, so you don't need this (skip to the next section)
2. You want to install it on a VirtualBox on your computer:
   - Follow the instructions here to set up the basics (this sets up Ubuntu 16.04 but you can use others) https://medium.com/@tushar0618/install-ubuntu-16-04-lts-on-virtual-box-desktop-version-30dc6f1958d0
   - Double the size of the harddrive though!
   - As part of this you will need the "ISO" file for the Ubuntu OS. Select your particular version of Ubuntu and download the ISO from here: Ubuntu 16.04 download (This is 16.04, switch to version that matches your preference)
3. You want to install it on your Raspberry Pi or on something else. (You'll have to let us know what those instructions are!)

10.10 Install Docker image

We'll use a docker image (vaddio/yocto-16.04) to preinstall Yocto dependencies.

1. First start a command shell. This done by clicking on the Ubuntu logo at the top left, typing "command line" and selecting the terminal application.
2. Next, install docker.

```
sudo apt install docker.io
```

1. Install vaddio/yocto-16.0.04

```
sudo docker run -it vaddio/yocto-16.04:16.04-latest /bin/bash
```

10.10.1 Install the base OS source code

Now that Docker and Yocto are installed, we need to install the source code specific for Vector:

```
sudo chmod 0777 .  && sudo su builduser
curl https://anki-vic-pubfiles.anki.com/license/prod/1.0.0/licences/OStarball.v160.tgz | tar -xz
```

10.10.2 Perform a test build

Finally, it's time to perform a test build. This will run a **long** time:

```
cd opensource/poky && source build/conf/set_bb_env.sh && build-victor-robot-image
```

To remove the intermediate files then:

```
buildclean
```

The && are used to avoid multiple run commands. Each run "command creates a new container with the deltas."

10.10.3 Credits:

Information from nammo on discord

10.11 How to unzip the OTA files

See the Project Victor Firmware folder for a description how to download the .ota files and how to verify them. It also includes a tool that can aid with the extraction.

There are three parts

1. First, the OTA's have to be decrypted

2. Next, the system files are extracted from the sysfs archive

3. Finally boot initramfs files can be extracted. (Their archive is a bit different)

10.11.1 Decrypting the OTA archives

The OTA files are tar.gz files, so they can be opened with tar (or similar tool). Among the files inside are two files:

`apq8009-robot-boot.img.gz` (encrypted)

`apq8009-robot-sysfs.img.gz` (encrypted)

Decrypting these files is done by:

```
openssl enc -d -aes-256-ctr -pass file:ota.pas -in apq8009-robot-boot.img.gz -out apq8009-robot-boot.img.dec.gz
```

```
openssl enc -d -aes-256-ctr -pass file:ota.pas -in apq8009-robot-sysfs.img.gz -out apq8009-robot-sysfs.img.dec.gz
```

With OpenSSL 1.1.0 or later, add "-md md5" to the command:

```
openssl enc -d -aes-256-ctr -pass file:ota.pas -md md5 -in apq8009-robot-boot.img.gz -out apq8009-robot-boot.img.dec.gz
```

```
openssl enc -d -aes-256-ctr -pass file:ota.pas -md md5 -in apq8009-robot-sysfs.img.gz -out apq8009-robot-sysfs.img.dec.gz
```

The keys can be found in the detail/keys folder in the Project Victor repository.

10.11.2 Unziping the system filesystem (sysfs) archive

On windows, the decoded `.img` files can extracted with 7zip

On linux, you can mount the file

1. gunzip the decrypted `apq8009-robot-sys.img.dec.gz`

2. `sudo mkdir /media/iso`

3. `sudo mount -o loop apq8009-robot-sys.img.dec /media/iso`

10.11.3 Unziping the boot initramfs filesystem (boot) archive

There are a couple of alterantives for tool sets:

- With Linux and Windows 10, there is a convenient tool

- For other systems, imgtool/imjtool from the New Android Book works

Linux and Windows 10 WSL methd

1. Go to How to unpack and repack boot and ramdisk files easily and follow the directions for the tool down load and installation

2. Added them to my path,

3. Opened wsl

4. gunzip the decrypted `apq8009-robot-boot.img.dec.gz`

5. Finally "unpack apq8009-robot-boot.img"

imgtool / imjtool

For other systems there is a help tool already exists

1. Download, build and install imjtool

2. gunzip the decrypted `apq8009-robot-boot.img.dec.gz`

3. Extracted the files using the image tool

```
imjtool boot.dec.img extract
```

That creates an extract folder with the ramdisk. The ramdisk is in "cpio" format.

Finally Extracted the files with

cd extract gzcat ramdisk| cpio -idmv

10.11.4 References and Resources

The decryption was originally posted to the Anki Vector Rooting google group

How to unpack and repack boot and ramdisk files easily on Linux and windows 10

- imgtool — now called imjtool

10.12 How to use Cozmo animation files

Cozmo's animation .bin files can be used on Vector, mostly. You do need know how to trigger them.

10.12.1 Why does this even work?

Cozmo's animation schema is very similar to Vector's.

When Vector reads and interprets the animation file it uses the flatbuffers library. This library uses default values for fields that are missing in a file — fields that Vector uses but that the Cozmo animation files doesn't provide. And the library ignores fields in the file that it doesn't know about — fields that Cozmo uses but Vector doesn't. So that gives it a lot of compatibility for faces, lights, motions.

Where Vector completely ignores Cozmo features is the sound. The sound features in the animation files is completely different between the two. (If cozmos sounds tracks work without fuss, Id be surprised ... or maybe they have a Cozmo compatibility layer?)

10.12.2 How to get a Cozmo animation file

10.12.3 How to put it on Vector

• Include how to link it into the behavior or what not

10.12.4 What about fixing up the audio stuff?

A bit of background the animation files send audio events, or audio trigger names (plus some audio parameter adjustments) that are used to tell the audio engine to play a particular sound.

You will have to convert the animation to JSON Then edit them to the new schema and change the audio trigger name to one that Vector supports. Then repack it into an animation bin file.

10.13 OSKR Tutorial for Windows 10

Digital Dream Labs has released a product called "OSKR". This allows you to turn your Vector into, essentially, a "dev" bot.

This means you can install software onto him which allows you to edit files on him.

Soon, the source will be released and this wiki will contain build instructions.

Follow these intructions carefully, and read through them before starting. OSKR isn't easy stuff.

10.13.1 Prerequesites

Windows 10 computer with Bluetooth support

Get your Vector's serial number to give to DDL

1. In Google Chrome (this has to be Google Chrome), go to Project Victor Web Setup.
2. It should show instructions and a "PAIR WITH VECTOR" button. If it says you need Chrome, go to `chrome://flags` in the URL bar and enable `Enable experimental web platform features`. Relaunch Chrome twice to make sure it got applied.
3. Turn on Vector and make sure he is at eyes.
4. Follow the instructions on the Vector web setup site. This may take many refreshes and reboots.
5. Once you are connected, type `logs` to download his logs.
6. Install this: 7-Zip (Windows)
7. Once they are downloaded: press the arrow on the logs which have downloaded, press "Show in folder", right click on the file, go into the `7-zip` part of the right click menu, press "Open archive".
8. Your serial number is in `factory/log1`. To open this file, double press it and select Notepad.
9. The QSN and ESN are at the bottom. For instance, mine is `QSN=323339903 # ESN=0030a012`. Copy this and fill out the form Digital Dream Labs gave you in an email.
10. Wait for the OTA to be sent to you, then do the rest of the steps.

Install Python

1. Python can be found at https://python.org. Here is a direct link to Python 3.9.1.
2. Python 3.9.1 Installer
3. When installing, make sure you check "Install Python 3.9.1 to PATH". Just press "Next" on all the other menus.

Download your OSKR OTA, find IP address, run Python server

1. Download your OSKR OTA with the link you have recieved from Digital Dream Labs. Open this link in the browser of your choice, and make sure it is saved in your Downloads folder.
2. Open Powershell. To do this: open the start menu, type "Powershell", then click the first thing that shows up.
3. Type `ipconfig`. This shows the network interface information. Your IP address is usually in the top section, next to "IPv4 Address ...". It usually starts with "192.168" or "10.".

4. In the same Powershell window, type `cd Downloads`, then type `py -m http.server`. `cd` changes your directory to the directory provided and `py` is Python. In this case, we have just told Python to open an HTTP server.

5. To test that you have the correct IP address and the server is running correctly, open a browser window and put the IP address you got into the browser URL bar followed by `:8000` (for instance, my local IP is 192.168.1.3. I would type `192.168.1.3:8000`). When you hit enter, there should be a directory listing.

6. If there is no directory listing, try a different IP address in `ipconfig` and make sure the server shows that it is running at `0.0.0.0:8000`.

Test your server

1. Download this: latest.ota

2. Make sure your Vector is turned on and at eyes/phone onboarding screen.

3. Put Vector into recovery by holding his button for 15 seconds on the charger. Keep holding it until the light turns green or purple again. He should be on `anki.com/v` after a while.

4. In Google Chrome (this has to be Google Chrome), go to Project Victor Web Setup.

5. It should show instructions and a "PAIR WITH VECTOR" button. If it says you need Chrome, go to `chrome://flags` in the URL bar and enable `Enable experimental web platform features`. Relaunch Chrome twice to make sure it got applied.

6. Follow the instructions on the web setup. If it is giving you trouble, try reloading the page and rebooting Vector (make sure you use the 15 second button hold method so he stays in recovery). It may take many attempts.

7. It should put you on a terminal. To connect him to Wi-Fi, type `wifi-connect ssid password`. Replace `ssid` with your network name and `password` with your network password. If you have a space in either of those, put quotations (") around it. For instance, one would be `wifi-connect "The Man Cave" pA55w4d`

8. Once connected to Wi-Fi, type `ota-start http://ipaddress:8000/latest.ota`. Replace `ipaddress` with your computer's actual IP address (for instance, mine would be `ota-start http://192.168.1.3:8000/latest.ota`). What you are doing here is installing the latest production OTA, and this isn't OSKR yet. This is like simulating what the phone app does when you first setup Vector.

9. If all has gone well, he should be at eyes. If he errors out, Vector may not be on the same network as your computer or you have already applied the OSKR unlock to your Vector.

Installing OSKR unlock

1. Make sure your Vector is at eyes. Do NOT put him into recovery this time.

2. Go to this site in Google Chrome. Project Victor Web Setup

3. Pair with Vector by following the instructions on the site. It will dump you to a terminal.

4. If he isnt connected to Wi-Fi, type `wifi-connect ssid password`. Replace `ssid` with your network name and `password` with your network password. If you have a space in either of those, put quotations (") around it. For instance, one would be `wifi-connect "The Man Cave" pA55w4d`

5. Time to install the OSKR unlock OTA. Type `ota-start http://ipaddress:8000/serial.ota`. Replace `ipaddress` with your computer's actual IP address and `serial` with the bot's serial number/name of the OTA (for instance, mine would be `ota-start http://192.168.1.3:8000/0060059b.ota`).

6. I recommend telling him to go to sleep while this is installing.

Installing OSKR firmware

After the unlock application, he should boot into recovery with the "OSKR" splash screen.

Congratulatons! Your bot is now unlocked!

Now we need to put on firmware which will allow you to do all the cool dev stuff.

1. Go to this site in Google Chrome. Project Victor Web Setup

2. Pair with Vector by following the instructions on the site. It will dump you to a terminal.

3. Download this: lkg.ota

4. a. Once connected to Wi-Fi, type `ota-start http://ipaddress:8000/lkg.ota`. Replace `ipaddress` with your computer's actual IP address (for instance, mine would be `ota-start http://192.168.1.3:8000/lkg.ota`).

5. Once you are done, user data will be cleared. Set him up with the Vector Robot app. If you are unable to do so, try using this .bat file (use PROD env): VectorSetup.bat

Getting in

You are now running OSKR firmware. This means you can go in via SSH and do a whole bunch of cool stuff.

1. In Google Chrome (this has to be Google Chrome), go to Project Victor Web Setup.

2. It should show instructions and a "PAIR WITH VECTOR" button. If it says you need Chrome, go to `chrome://flags` in the URL bar and enable `Enable experimental web platform features`. Relaunch Chrome twice to make sure it got applied.

3. Turn on Vector and make sure he is at eyes.

4. Follow the instructions on the Vector web setup site. This may take many refreshes and reboots.

5. Once you are connected, type `logs` to download his logs.

6. Make sure you have this installed: 7-Zip (Windows)

7. Once they are downloaded: press the arrow on the logs which have downloaded, press "Show in folder", right click on the file, go into the `7-zip` part of the right click menu, press "Open archive".

8. Your SSH key is in `data/ssh/id_rsa-Vector-####`. Drag this to your desktop, and make sure you leave the .pub one alone.

9. Open Powershell (Start menu, type "Powershell", press first thing that shows up)

10. In Powershell, type `cd Desktop`, then `ssh -i id_rsa-Vector-#### root@vectorip`. Replace `####` with the actual Vector ID and `vectorip` with Vector's actual IP address. Vector's IP address can be found in CCIS. You can go to this by placing him on the charger, double pressing his button, then lifting his lift up then down. His IP address will be the number in green (or yellow idk. im colorblind). For instance, mine would be `ssh -i id_rsa-Vector-H9P8 root@192.168.1.4`. When you are typing it, and you are in the middle like `ssh -i id_rsa-V`, you can press tab for it to auto complete.

11. If you get any error: make sure you are on the same network as Vector. If it can't find the command `ssh`, install Git Bash. You can open Git Bash and run the same `cd Desktop` and `ssh -i id_rsa-Vector-#### root@vectorip`

10.14 Making paper dolls for Vector to play with
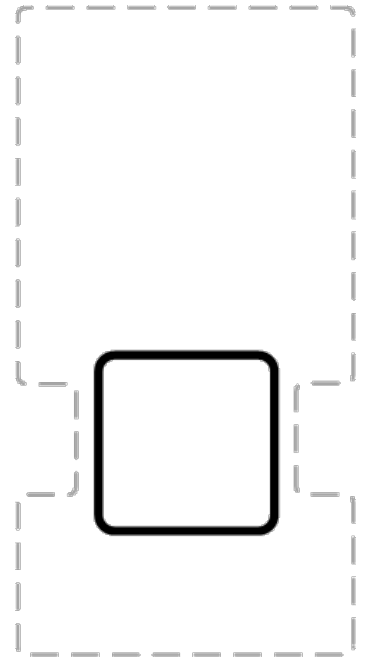
10.14.1 Laser Printer Version

1. Download one of the two version of the pattern template:

   • Without the symbols: PDF or Visio

   • One includes little symbols on them

   • The other doesn't (You can add the pictures using stickers)

2. Make any changes, like adding color patters (Optional)

3. Print. If you're like me, the printer doesn't like card stock

4. Cut out

5. Use a glue, like a spray glue, to attach to cardstock

6. Cut that.

7. Color it in (Optional)

8. Add little tails so it will stay upright

10.14.2 Cricut Version

1. Download one of the two version of the pattern template:

   • Without the symbols: PDF or Visio

   • One includes little symbols on them

   • The other doesn't (You can add the pictures using stickers)

   • One has the cut pattern

   • Two have the print patterns. Pick one.

2. Make any changes, like adding color patters (Optional)

3. Use Cricut, print-then-cut mode

   • Print

   • Cut out

   • Cut out card stock

4. Print on cardstock? If you're like me, the printer doesn't like card stock

5. Use a glue, like a spray glue, to attach to cardstock

6. Color it in (Optional)

7. Add little tails so it will stay upright

11. Protocols

11.1 The Chipper Services

This describes the interactions with Anki's automatic speech response server. The audio after a "Hey Vector" is sent to servers for processing. The servers send a response back, in the form of an intent. This is a code and a structure that represents an action to carry out in response to the spoken request, query, or statement; it may represent the action requested, an answer to a query, or an action that emotionally responds to what was said. The intent structures are described in another page.

11.1.1 Common Elements

The enumerations and structures in this section are common to many commands.

Enumerations

AUDIOENCODING

INTENTSERVICE

LANGUAGECODE

ROBOTMODE

Structures

The following structures are present in the Go code, but their use is not known.

WEATHER LOCATION

The *WeatherLocation* structure has the following fields:

Table: JSON Parameters for the weather location structure

Field	Type	Description
city	string	
country	string	
state	string	

11.1.2 Commands and Responses

Unknown

We see these in the logs, but it doesn't match what the Go code has for generated grpc protobuf stuff…?

REQUEST

The request sent to the server has the following fields

Table: Parameters for ASR request

Field	Type	Description
session	string	Weirdo hex line thing
type	string	e.g. "streamOpen"

Not sure where the stream open goes. Does it upload the file, or live stream it?

RESPONSE

The server response message has the following fields

Table: Parameters for ASR response

Field	Type	Description
intent	string	The type of intent
metadata	string	This can be an empty string, but it can also be a string with colon delimited parameters. It often has the pattern "text: unquoted-string confidence: float handler: LEX" The "text:" can be followed by transcription of the spoken text, the "confidence:" followed by a floating point number representing how confident the speech-to-text engine is in the transcription.
parameters	JSON string	This is a string containing the JSON serialization of the intent parameters.
type	string	e.g. "result"

Streaming Connection Check

REQUEST

The *StreamingConnectionCheckRequest* request message has the following fields:

Table: JSON Parameters for the streaming connection check request

Field	Type	Description
app_key		
audio_per_request		
device_id		Probably the robot's ESN.
firmware_version		
input_audio		
session		
total_audio_ms	int	

RESPONSE

The *ConnectionCheckResponse* response message has the following fields:

Table: JSON Parameters for the connection check response

Field	Type	Description
frames_received		A count?
status	*Status*	

Streaming Intent

This is used to TBD on the server.

The *StreamingIntentRequest* request message has the following fields:

Table: JSON Parameters for the streaming intent request

Field	Type	Description
app_key		
audio_encoding	*AudioEncoding*	Probably opus or ogg
boot_id		
device_id		Probably the robot's ESN.
firmware_version		
input_audio		
input_service		
language_code	*LanguageCode*	
mode	*RobotMode*	
save_audio	bool	
session		
single_utterance		
skip_das	bool	
speech_only	bool	

The *IntentResponse* response message has the following fields:

Table: JSON Parameters for the intent response

Field	Type	Description
audio_id		
device_id		Probably the robot's ESN.
intent_result	*IntentResult*	
is_final	bool	
mode	*RobotMode*	
session		
speech_result	*SpeechResult*	

The *IntentResult* structure has the following fields:

Table: JSON Parameters for the intent result structure

Field	Type	Description
action		
all_parameters_present	bool	
has_context	bool	
intent_confidence	float	
kgresponse		
parameters		
query_text		
service		
speech_confidence	float	

The *SpeechResult* structure has the following fields:

Table: JSON Parameters for the speech result structure

Field	Type	Description
is_final	bool	
transcript	string	

Streaming Knowledge Graph

This is used to query the knowledge graph on the server. Note: I'm not convinced that Vector uses this. It may be some of how the server internally works that got left in Vector's *vic-cloud*.

REQUEST

The *StreamingKnowledgeGraphRequest* request message has the following fields:

Table: JSON Parameters for the streaming knowledge graph request

Field	Type	Description
app_key		
audio_encoding	AudioEncoding	Probably opus or ogg
boot_id		
device_id		Probably the robot's ESN.
firmware_version		
input_audio		
language_code	LanguageCode	
save_audio		
skip_das	bool	
timezone		

RESPONSE

The *KnowledgeGraphResponse* response message has the following fields:

Table: JSON Parameters for the streaming knowledge graph response

Field	Type	Description
audio_id		
command_type		
device_id		Probably the robot's ESN.
domains_used		
query_text		
session		
spoken_text		
text_input		

Text

Note: I'm not convinced that Vector uses this. It may be some of how the server internally works that got left in Vector's vic-cloud.

REQUEST

The *TextRequest* request message has the following fields:

Table: JSON Parameters for the text request

Field	Type	Description
device_id		Probably the robot's ESN.
firmware_version		
intent_service	*IntentService*	
language_code	*LanguageCode*	
mode	*RobotMode*	
session		
skip_das	bool	

11.2 The JDocs Services

The *Vic-Cloud* services stores information on a "JDocs" server. This unusual name appears to be short for "JSON Documents." This server allows Vector to store settings and usage statistics. This allows the settings and usage to be viewed on a mobile device on a remote network.

The interactions are basic: store, read, and delete a JSON blob by an identifier. The description below gives the JSON keys, value format. It is implemented as gRPC/protobuf interaction over HTTP.

The commands include:

- An 'echo' command to check connectivity with the server.

- Reading and writing a document

- Deleting a document

- Viewing account documents

11.2.1 Common Elements

The enumerations and structures in this section are common to many commands.

Enumerations

STATUS

Structures

JDOC

The JDoc structure has the following fields:

Table: JSON structure

Field	Type	Description
client_meta	string	Probably an empty string
doc_version	uint64	A number used to uniquely identify changes to the setting structure, and be able to tell which ones is the more recent settings. Most often this is the number of times that the settings have been changed.
fmt_version	uint64	The version number of the jdoc structure schema; this is always 1.
json_doc	string	The jdoc structure serialized as a string.

11.2.2 Commands and Responses

Delete Document

This is used to remove the document from the server.

REQUEST

The *DeleteDocReq* request message has the following fields:

Table: JSON Parameters for delete document request

Field	Type	Description
account	string	The account to delete the document from.
doc_name	string	The name of the document to delete.
thing	string	The thing id is a 'vic:' followed by the serial number

RESPONSE

The *DeleteDocResp* response message has the following fields:

Table: JSON Parameters for the delete document response

Field	Type	Description
latest_version	uint64	The current version of the document in the repository.
status	string	

Echo Test

REQUEST

The *EchoReq* request message has the following fields:

Table: JSON Parameters for the echo request

Field	Type	Description
data		

RESPONSE

The *EchoResp* response message has the following fields:

Table: JSON Parameters for the echo response

Field	Type	Description
data		comment: I'm not sure this field is sent back

Read Documents

REQUEST

The *ReadDocsReq* request message has the following fields:

Table: JSON Parameters for the read documents request

Field	Type	Description
account	string	The account to read from.
items	*ReadDocsReq_Item* []	Array of the items requested.
thing	string	The thing id is a 'vic:' followed by the serial number.

The *ReadDocsReq_Item* structure has the following fields:

Table: JSON Parameters for the read documents item

Field	Type	Description
doc_name	string	The name of the document to retrieve.
my_doc_version	UInt64	The version to retrieve(?)

RESPONSE

The *ReadDocsResp* response message has the following fields:

Table: JSON Parameters for the read documents response

Field	Type	Description
items	_ReadDocsResp_item[]	An array of the documents.

The *ReadDocsResp_Item* structure has the following fields:

Table: JSON Parameters for the read document item response

Field	Type	Description
doc	*JDoc*	The document structure.
status	*Status*	

View Account Document

This command is used to retrieve a JSON blob on the server. The request allows personally identifying information to be included or omitted.

REQUEST

The *ViewDocReq* request message has the following fields:

Table: JSON Parameters for view account document request

Field	Type	Description
account	string	The account to read from.
json_doc	*JDoc*	The document structure. {TODO: why is this here? this makes it seem like it doesn't } *Optional*
doc_name	string	The name of the document to view. *Optional*
thing	string	The thing id is a 'vic:' followed by the serial number. *Optional*

RESPONSE

The *ViewDocsResp* response message has the following fields:

Table: JSON Parameters for view account document response

Field	Type	Description
docs	TBD[]	The documents (?)

Write Document

This command is used to store a JSON blob on the server.

The *WriteDocReq* request message has the following fields:

Table: JSON Parameters for write document request

Field	Type	Description
account	string	The account to write to.
doc	*JDoc*	The document structure.
doc_name	string	The name of the document to write.
thing	string	The thing id is a 'vic:' followed by the serial number.

The *WriteDocResp* response message has the following fields:

Table: JSON Parameters for write document response

Field	Type	Description
latest_doc_version	UInt64	The current version of the document in the repository.
status	*Status*	

11.3 Intent Structures

This describes the structures associated with intents. The audio after a "Hey Vector" is sent to servers for processing. The servers send a response back, in the form of an intent. This is a code and a structure that represents an action to carry out in response to the spoken request, query, or statement; it may represent the action requested, an answer to a query, or an action that emotionally responds to what was said.

11.3.1 Cloud Intents

Cloud Intent	Description
intent_amazon_signin	
intent_amazon_signout	
intent_blackjack_hit	
intent_blackjack_playagain	
intent_blackjack_stand	
intent_character_age	
intent_clock_checktimer	
intent_explore_start	
intent_global_delete_extend	
intent_global_stop_extend	
intent_greeting_goodbye	
intent_greeting_hello	
intent_greeting_goodmorning	
intent_greeting_goodnight	
intent_imperative_abuse	
intent_imperative_affirmative	
intent_imperative_apologize	
intent_imperative_come	
intent_imperative_dance	
intent_imperative_eyecolor	
intent_imperative_eyecolor_specific_extend	
intent_imperative_fetchcube	
intent_imperative_findcube	
intent_imperative_lookatme	
intent_imperative_lookoverthere	
intent_imperative_love	
intent_imperative_negative	
intent_imperative_praise	
intent_imperative_scold	
intent_imperative_quiet	
intent_imperative_shutup	
intent_imperative_volumedown	
intent_imperative_volumelevel_extend	
intent_imperative_volumeup	
intent_knowledge_promptquestion	

Cloud Intent	Description
intent_knowledge_response_extend	
intent_knowledge_no_response	
intent_names_username_extend	
intent_message_playmessage_extend	
intent_message_recordmessage_extend	
intent_imperative_backup	
intent_imperative_forward	
intent_imperative_turnaround	
intent_imperative_turnleft	
intent_imperative_turnright	
intent_names_ask	
intent_play_anygame	
intent_play_anytrick	
intent_play_blackjack	
intent_play_fistbump	
intent_play_pickupcube	
intent_play_popawheelie	
intent_play_rollcube	
intent_play_specific_extend	
intent_seasonal_happyholidays	
intent_seasonal_happynewyear	
intent_clock_settimer_extend	
intent_clock_time	
intent_system_noaudio	
intent_status_feeling	
intent_system_charger	
intent_system_sleep	
intent_photo_take_extend	
intent_weather_extend	

11.3.2 Parameters for the Intents

The following are the parameters for each of the intents. These structures are serialized as a JSON string and passed in the parameters property of the ASR response. The intents not listed below do not have any added parameters properties.

Clock set timer

The *intent_clock_settimer_extend* intent parameters structure has the following properties:

*Table: **intent_clock_settimer_extend** properties*

Property	Type	Units	Description
timer_duration	int	seconds	number of seconds to set the timer to.

Global Delete

The *intent_global_delete_extend* intent parameters structure has the following properties:

*Table: **intent_global_stop_deletable** properties*

Property	Type	Units	Description
entity_behavior_deletable	string	See the table below for an enumeration of the allowed values.	The item to delete.

The set of acceptable items that can be deleted include:

Property Value	Description
message	
photo	
timer	

Global stop

The *intent_global_stop_extend* intent parameters structure has the following properties:

*Table: **intent_global_stop_extend** properties*

Property	Type	Units	Description
entity_behavior_stoppable	string	See the table above for an enumeration of the allowed values.	The item to delete.

Imperative Eye Color

The *intent_imperative_eyecolor_extend* intent parameters structure has the following properties:

*Table: **intent_imperative_eyecolor_extend** properties*

Property	Type	Units	Description
eye_color	string	See the table below for an enumeration of the allowed values.	The name of the color to set the eye color to.

The enumeration of eye color values:

Property Value	Description
COLOR_BLUE	
COLOR_GREEN	
COLOR_ORANGE	
COLOR_PURPLE	
COLOR_TEAL	
COLOR_YELLOW	

Imperative Volume Lvel

The *intent_imperative_volumelevel_extend* intent parameters structure has the following properties:

*Table: **intent_imperative_volumelevel_extend** properties*

Property	Type	Units	Description
volume_level	string	See the table below for an enumeration of the allowed values.	The name of the volume level to change to.

The enumeration of volume levels:

Property Value	Description
VOLUME_1	
VOLUME_2	
VOLUME_3	
VOLUME_4	
VOLUME_5	

Knowledge Response

This *intent_knowledge_response_extend* intent parameters structure has the following properties:

*Table: **intent_knowledge_response_extend** properties*

Property	Type	Units	Description
answer	string	The text to be spoken	
answer_type	string	"InformationCommand" "NoResultCommand"	
query_text	string	The text of the question asked.	

Play Message

This *intent_message_playmessage_extend* intent parameters structure has the following properties:

*Table: **intent_message_playmessage_extend** properties*

Property	Type	Units	Description
given_name	string		The name of the person to send the message to.

User name

This *intent_names_username_extend* intent parameters structure has the following properties:

*Table: **intent_names_username_extend** properties*

Property	Type	Units	Description
username	string		The name of the user

Take Photo

The *intent_photo_take_extend* intent parameters structure has the following properties:

*Table: **intent_photo_take_extend** properties*

Property	Type	Units	Description
entity_photo_selfie	string		Empty string if taking a photo, "photo_selfie" if taking a selfie.

Weather

The *intent_weather_extend* intent parameters structure has the following properties:

*Table: **intent_weather_extend** properties*

Property	Type	Units	Description
condition	string		The current weather conditions. One of "Clear", "Cloudy", "Cold", "Rain", "Snow", "Stars", "Sunny", "Thunderstorms", or "Windy"
is_forecast	string	"false" or "true"	"false" if it is the current weather conditions; "true" if forecasted weather conditions.
local_datetime	string		The local time (where the weather conditions apply) in UTC ISO 8601 format.
speakable_location_string	string		The location name that Vector could employ in his verbal description of the temperature.
temperature	string	degrees	The current or forecasted temperature, in the given units.
temperature_unit	string		F or C, for the units

11.4 Log Server for Vector

This is an overview of how a server to receive logs from Vector can work.

11.4.1 The file system layout

I created a folder to store information from this Vector's logs. The received file contents in the HTTP upload will be saved in a folder nested underneath that. This is the naming scheme that I settled on:

```
[Server base] / [serial #] / year-month /  [ time stamp]
```

This creates separate directory trees for each robot, even if the name robot name changes.

Next is a folder for each month. The format I went with has year as a 4 digits, and months as two digits in my example. (yyyy-MM).

Finally each upload gets its own timestamped folder — timestamped with the time it was received. I used the format. The contents differ with each kind of upload.

Why is there a separate folder for each month? Vector produces at least 50-100 DAS files per day; This can create a lot of files (and folders) very quickly. If we don't spread them across a few sub-folders, the number of files for a given folder is too much for a person to manage. Then, at a bit larger number, the OS will have a collapse in efficiency, taking exponential time to list or access the files. The number depends on the file system and OS... it could be 32768, or 65536 or less. In other words, a single robot could create 36500 uploads in a single year, that is too much people and computers in a single folder, so I had 'em spread out.

11.4.2 The types of data sent in logs

The types of data and file formats:

- The crash logs

- minidump

- DAS json events

- Linux system logs. Note: as I recall these might be included in the crash logs. Thee SDK can also trigger sending them

Things not included:

- Wifi info: understanding the issues the network connectivity: data rate, latency, dropped connections. This might be part of the logs sent above (I don't recall) but aren't separately gathered and sent.

11.4.3 How to decide what kind of file has been received.

1. Look at the file name (excluding the path) of the posted file. If the file name is "DAS" (regardless of case), and this is a HTTP POST (not a form), it is DAS log upload; otherwise

2. Look at the name of the file without the extension and drop the file extensions. If the file name starts with "victor-" (regardless of case), this is a compress Vector log file archive.

3. Check for the HTTP header,"Usr-RobotESN" If there is one, this is a fault report containing crash dumps… Otherwise,

4. Does it have a body? Then it is a crash dump

DAS files

The DAS upload includes the data in the HTTP stream content. The serial number for the robot is buried in the DAS contents, so has to be extracted /after/ the stream has been received.

- Looked for the attached "MessageBody.json.gz" that is the DAS events to save to the folder. Decompress a copy, read the first record and pop the robot id for the folder to save it in.
- Save the parameters associated with the stream to a file called "params.txt"
- Save the header fields to a file called "info.txt" in the folder. The most important are: UserAgent RemoteEndPoint

The log files

This stores the logs uploaded from the Vector The robot's electronic serial number is the part after the "victor-" in the file name. When logs files are uploaded, I found it was important to store some meta related to the upload. I saved the header fields to a file called "info.txt" in the folder. The most important are:

- *UserAgent*
- *RemoteEndPoint*

This includes a file attached to the upload. This uploaded file is saved to the folder.

Crash Dumps

The crash dumps are multiple parts attached to a form upload stream. The robot serial number is in the form parameter "robot.esn". If there isn't one associated, fall back to "unknown" I saved the header fields to a file called "info.txt" in the folder. The most important are:

- *UserAgent*
- *RemoteEndPoint*

The form parameters fields are captured into a file called "params.txt" The files attached to form are also saved to the folder (using the name of the file, not any other parts of the path.)

11.5 The Token Manager

This describes the interactions with token manager. This server allows Vector to protect any-old application from connecting to it. Instead it requires proof that the application "knows" the users account name and password. That proof is in the form of a token given to it and the application after the application has authenticated with the token manager.

The commands include:

- Primary user management: associating a user or client, refreshing the association, and remove the association

- Revoking tokens; listing the revoked tokens

- Revoking a factory certificate

11.5.1 Common Elments

The enumerations and structures in this section are common to many commands.

Structures

STSTOKEN

The *StsToken* structure has the following fields:

Table: Parameters for the STS token structure

Field	Type	Description
access_key_id		
expiration		
secret_access_key		
session_token		The token from the security token service for the session.

TOKENBUNDLE

The *TokenBundle* structure has the following fields:

Table: Parameters for the token bundle structure

Field	Type	Description
client_token		
sts_token		The token from the security token service
token		

TOKENPAGE

The *TokenPage* structure has the following fields:

Table: Parameters for the token page structure

Field	Type	Description
done		
last_key		
tokens	???[]	

11.5.2 Commands and Responses

Associate Primary User

This command is used to TBD? See also the disassociate primary user and reassociate primary user commands

The *AssociatePrimaryUserRequest* request message has the following fields:

Table: Parameters for the associate primary user request

Field	Type	Description
app_id		
client_name		
expiration_minutes		
generate_sts_token		
revoke_client_tokens		
session_certificate		
skip_client_token		

The *AssociatePrimaryUserResponse* response message has the following fields:

Table: Parameters for the associate primary user response

Field	Type	Description
data		

Associate Secondary Client

This command is used to TBD?

The *AssociateSecondaryClientRequest* request message has the following fields:

Table: Parameters for the associate secondary client request

Field	Type	Description
app_id		
client_name		
user_session		

RESPONSE

The *AssociateSecondaryClientResponse* response message has the following fields:

Table: Parameters for the associate secondary client response

Field	Type	Description
data		

Disassociate Primary User

This command is used to TBD? See also the associate primary user and reassociate primary user commands.

REQUEST

The *DisassociatePrimaryUserRequest* request message has no fields.

RESPONSE

The *DisassociatePrimaryUserResponse* response message has no fields.

List Revoked Tokens

This command is used to TBD?

REQUEST

The *ListRevokedTokensRequest* request message has the following fields:

Table: Parameters for the list revoked tokens request

Field	Type	Description
previous_key		

RESPONSE

The *ListRevokedTokensResponse* response message has the following fields:

Table: Parameters for the list revoked tokens response

Field	Type	Description
data		

Reassociate Primary User

This command is used to TBD? See also the associate primary user and disassociate primary user commands.

The *ReassociatePrimaryUserRequest* request message has the following fields:

Table: Parameters for the reassociate primary user request

Field	Type	Description
app_id		
client_name		
expiration_minutes		
generate_sts_token		
skip_client_token		

The *ReassociatePrimaryUserResponse* response message has the following fields:

Table: Parameters for the reassociate primary user response

Field	Type	Description
data		

Refreshing a Token

This command is used to TBD?

The RefreshTokenRequest request message has the following fields:

Table: Parameters for the refresh token request

Field	Type	Description
expiration_minutes		
refresh_jwt_tokens		
refresh_sts_tokens		

The *RefreshTokenResponse* response message has the following fields:

Field	Type	Description
data		

Revoking a Factory Certificate

This command is used to TBD?

The *RevokeFactoryCertificateRequest* request message has the following fields:

Table: Parameters for the revoke factory certificate request

Field	Type	Description
certificate_id		

The *RevokeFactoryCertificateResponse* response message no fields/

Revoking a Token

This command is used to TBD?

The *RevokeTokensRequest* request message has the following fields:

Table: Parameters for the revoke tokens request

Field	Type	Description
key		
search_by_index		

The *RevokeTokensResponse* response message has the following fields:

Table: Parameters for the revoke tokens response

Field	Type	Description
tokens_revoked	[]	A list of the tokens that have been revoked.

12. Service

12.1 Arms

In a fall, Vector's lift arms may pop apart. This was an intentional design to prevent them from breaking.

12.1.1 Broken lift gear

From Discord:

Sometimes, when Vector or Cozmo take a particularly unlucky fall, the force of impact is transferred from the arm into the gearbox. There's a repair that works about half the time: You need to remove the arms, rotate the lift gear 180 degrees (you can use the arm as a tool to do this), and reinstall the arms. When this works, it's because you're moving the broken gear tooth out of the way, and using the "other half" of the gear teeth.

Cozmo and Vector have the same basic arm design and arm disassembly/reassembly is one of the safer operations you can perform.

To disassemble:

1. "Pull at the upper set of arms at the shoulder (the joint where they connect to the body, in back). Pull hard enough to pop both off their joints.
2. "Tilt the lift up - higher than it normally can, over the head.
3. "When the lift is high enough, you'll find a point where you can pull gently on the lower arms and they'll pop free.

Reassembly is the reverse of the above. The important thing to notice during disassembly is that the lower arms have a certain angle where they easily come out. You have to use that angle to pop them in and out. If they are not coming out/ going in easily, the angle is not high enough or too high.

12.1.2 Spare parts

Anki Vector Lifting Forks by Dauler July 05, 2019

3D Model .stl Vector Robot Lift Forks FDM and Resin Models DesignsByDauler

12.2 Replacing the Battery

Please fill this in!

See this iFixit instruction

Replacement battery options:

- https://www.ebay.com/itm/3-7V-320-mAh-Polymer-Li-battery-Lipo-For-GPS-Mp4-DVD-PDA-Camera-Tablet-PC-402535/122584822407

- A bigger battery: https://www.aliexpress.com/item/32956226523.html

Note: neither of these is a the same "toy safe" kind as the original, and so have a few different characteristics

Desolder the battery's positive lead first. Then wrap the end of the lead in electrical tape to insulate it -- to help prevent it from touching sensitive electronics.

See also https://www.reddit.com/r/AnkiVector/comments/i48qg8/vector_story_with_happy_ending/
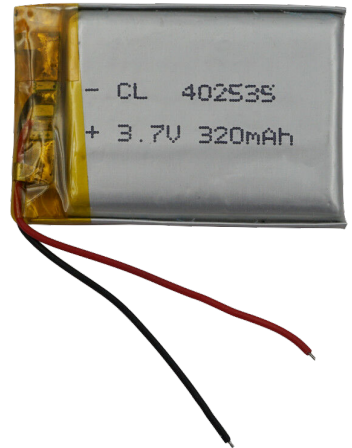
12.3 Disassembly notes

Summary:

- Avoid shorting anything

- Wear gloves

- Don't disassemble / move the time of flight sensor & window

- Don't disassemble / move the camera and its lens/window

See also iFixit's services guides for Vector

12.3.1 Avoid Shorts, disconnect the battery

Tip from Discord:

> Take care to avoid shorting anything while the battery is connected. I had a habit of desoldering the battery's positive lead as soon as I could reach it, before pulling the guts entirely out of the robot - just to avoid damage.

Wrap the end of the positive lead in electrical tape to seal it off, and be sure that it gets no where near any of the electronics.

12.3.2 Wear gloves

Wear gloves to keep fingerprints off of the inside of the camera lens and time of flight sensor lens.

Tip from Discord:

> Sometimes a fingerprint or smudge on the [time of flight sensor] window can mess it up.

That can be hard to clean, especially if it is on the inside. Nitrile gloves can help prevent this.

12.3.3 Don't muck with the time of flight sensor

Tip from Discord:

> Disassembling a Vector can cause a change in the relationship between the [time of flight] sensor window and [time of flight] sensor behind it. Any change like that requires re-calibration.

And we can't recalibrate.

12.4 Exploded View

12.4.1 Exploded Views of Vector's assembly

NUMBER	PART NAME
1	BACKPACK
2	LIGHT PIPE
3	MIC-BACKPACK PCBA
4	BACKPACK MIC-PCBA FRAME
5	LIFT SHAFT
6	CASE CENTER BACK
7	HEAD LEFT
8	HEAD RIGHT
9	HEAD BIAS SPRING
10	HEAD EAR
11	HEAD TRANSMISSION
12	LIFT TRANSMISSION
13	CASE LEFT
14	CASE RIGHT
15	LIFT ARM ALIGN LEFT
16	LIFT ARM ALIGN RIGHT
17	LIFT ARM RIGHT
18	LIFT ARM LEFT
19	RIM BACK
20	RIM FRONT
21	RIM FRONT RETAIN
22	RIM BACK RETAIN
23	SCREW, WASHER HEAD M2X4.5X8
24	TREAD
25	WEIGHT SIDE
26	TRANS BODY PCBA
27	CASE STOP
28	SCREW M2X6
29	HEAD CAMERA FRAME PCBA
30	HEAD MASK
31	LIFT FRAME
32	LIFT HOOK
33	LIFT SCREW
34	NECK PIN
35	SPEAKER ASSEMBLY
36	CASE CENTER
37	WEIGHT HARD-POINT FRONT
38	BACKPACK HARNESS

Zoom in on the exploded view of head

An exploded view of backpack assembly

BACKPACK
FRAME

MICROPHONE
BACK MOUNTED
(4X)

BACKPACK/MICROPHONE PCBA

PORON
GASKET/SEAL
LOWER
(4X)

BACKPACK MESH/SEAL
/HOLDER/COMPRESSOR
TUBES (LEFT/RIGHT)

PORON
GASKET/SEAL
UPPER
(4X)

WATER/DUST
RESISTANT
MEMBRANE
(4X)

STAINLESS STEEL
WIRE MESH
(4X)

BACKPACK PLASTIC
WITH MICROPHONE
HOLES

SEE DETAIL A

SECTION MICROPHONES-MICROPHONES
SCALE 2:000

MICROPHONES

MICROPHONES

12.5 LCD Replacement

A batch of Vectors have LCD screens that form lines on them. The lines often start at the bottom, both marring the eyes and making the Bluetooth LE pairing pin codes illegible.

The community initially thought that the LCD connections to the head-board became delaminate with falls, lots of head motion, shaking and bad luck. This was wrong.

It was a bad batch of LCDs with a fauly gasket on the glass/plastic pieces that let humidity in and corrodes the electrical bits.

Project Victor has done some work to locate a replacement LCD LCD replacement

   • ST0103A3W from http://www.santechnology.com/products/

terminator3d3700 has been working on a home set up to replace LCD displays, with some success:

   • See https://www.reddit.com/r/AnkiVector/comments/jwu77d/vector_displays/

   • https://www.reddit.com/r/AnkiVector/comments/ju7i4i/vectors_new_screen/

   • Contact him if interested

12.6 Parts kits

- Replacement Parts: Wheel hubs, wheels, treads, lift arms, body boards, back pack boards, ears, gears, etc.
- A listing 3D printable parts

12.6.1 Cube service

The cube uses a 1.5V "N" battery, aka "E90" or "LR1". **DO NOT USE AN A23 -- IT WILL DESTROY THE ELECTRONICS**

- Kinvert how to change the battery
- Official how to change the battery

3D Printable cube battery door

12.6.2 Replacement boards

We do not have these, but it would be nice:

- Body boards
- Time of flight boards
- LCD module
- Motor encoder
- Backpack boards

Modifying board firmware:

- Mechanism to sign new body board FW?

13. Software design

13.1 Animation Triggers

Trigger Name	Description
AlexaError2Idle	
AlexaErrorLoop	
AlexaErrorLoop	
AlexaIdle2Listen	
AlexaIdle2Speak	
AlexaListen2Error	
AlexaListen2Idle	
AlexaListen2Speak	
AlexaListen2Think	
AlexaListenLoop	
AlexaNotification	
AlexaSignOut	
AlexaSpeak2Error	
AlexaSpeak2Idle	
AlexaSpeak2Listen	
AlexaSpeakLoop	
AlexaThink2Error	
AlexaThink2Idle	
AlexaThink2Speak	
AlexaThinkLoop	
AlreadyAtFace	
AudioOnlyHuh	
BlackJack_Deal	
BlackJack_GetIn	
BlackJack_GoodLuck	
BlackJack_Idle	
BlackJack_Quit	
BlackJack_Response	
BlackJack_RtpIdle	
BlackJack_RtpPlayerNo	
BlackJack_RtpPlayerYes	
BlackJack_RtpRequest	
BlackJack_RtpTimeOut	
BlackJack_SpeechGetIn	
BlackJack_SpeechShortStatement	

Trigger Name	Description
BlackJack_Spread	
BlackJack_Swipe	
BlackJack_VictorBlackJackLose	
BlackJack_VictorBlackJackWin	
BlackJack_VictorBust	
BlackJack_VictorLose	
BlackJack_VictorPush	
BlackJack_VictorWin	
BumpObjectFastGetIn	
BumpObjectFastGetOut	
BumpObjectFastLoop	
BumpObjectSlowGetIn	
BumpObjectSlowGetOut	
BumpObjectSlowLoop	
Carrying	
ChargerDockingAlreadyHere	
ChargerDockingDrivingEnd	
ChargerDockingDrivingLoop	
ChargerDockingDrivingStart	
ChargerDockingFailure	
ChargerDockingLeftTurn	
ChargerDockingRaiseLift	
ChargerDockingRequest	
ChargerDockingRequestGetout	
ChargerDockingRequestPickup	This animation is played by *EmergencyModeInAir* behavior to ask a human companion to put Vector in the charger.
ChargerDockingRequestWaitLoop	
ChargerDockingRightTurn	
ChargerDockingSearchAfterCompletedSearch	
ChargerDockingSearchSingleTurn	
ChargerDockingSearchSingleTurnEnd	
ChargerDockingSearchWaitForImages	
ChargerDockingSettle	
ChargerDockingSevereRequest	
ChargerDockingSevereRequestGetout	

Trigger Name	Description
ChargerDockingSorryButLowBattery	This animation is played by *EmergencyModeAnimDispatcher* behavior to show that Vector's battery is low
ChargerReaction	
ClockGetIn	
ClockGetOut	
ComeHereStart	
ComeHereSuccess	
ConnectToCubeFailure	
ConnectToCubeGetIn	
ConnectToCubeLoop	
ConnectToCubeLostConnection	
ConnectToCubeSuccess	
ConnectWakeUp	
ConnectWakeUpLights	
Connected	
CountingFastLoop	
CountingGetInEven	
CountingGetInOdd	
CountingGetOut	
CountingSlowLoop	
CubePounceBackup	
CubePounceDriveGetIn	
CubePounceDriveGetOut	
CubePounceDriveLoop	
CubePounceFake	
CubePounceGetIn	
CubePounceGetOutBored	
CubePounceGetReady	
CubePounceGetUnready	
CubePounceIdleLiftDown	
CubePounceIdleLiftUp	
CubePounceLoseHand	
CubePounceLoseSession	
CubePouncePlayerLose	
CubePouncePlayerWin	
CubePouncePounceClose	

Trigger Name	Description
CubePouncePounceNormal	
CubePounceReactToCube	
CubePounceWinHand	
CubePounceWinSession	
DEPRECATED_AcknowledgeFaceNamed	
DEPRECATED_AcknowledgeFaceUnnamed	
DEPRECATED_AcknowledgeObject	
DEPRECATED_ComeHere_SearchForFace	
DEPRECATED_CubeMovedSense	
DEPRECATED_CubeMovedUpset	
DEPRECATED_DizzyReactionHard	
DEPRECATED_DizzyReactionMedium	
DEPRECATED_DizzyReactionSoft	
DEPRECATED_DizzyShakeLoop	
DEPRECATED_DizzyShakeStop	
DEPRECATED_DizzyStillPickedUp	
DEPRECATED_LaserAcknowledge	
DEPRECATED_LaserDriveEnd	
DEPRECATED_LaserDriveLoop	
DEPRECATED_LaserDriveStart	
DEPRECATED_LaserGetOut	
DEPRECATED_LaserPounce	
DEPRECATED_LookDownForLaser	
DEPRECATED_NamedFaceInitialGreeting	
DEPRECATED_SearchForFace_FoundFace	
DEPRECATED_SearchForFace_Search	
DEPRECATED_StackBlocksSuccess	
DanceBeatCantDoThat	
DanceBeatEyeHold	
DanceBeatGetIn	
DanceBeatGetOut	
DanceBeatGetReady	
DanceBeatListening	
DanceBeatNoBeatDetected	
DanceToTheBeat	

Trigger Name	Description
DealerCardLayout	
DockEndDefault	
DockLoopDefault	
DockStartDefault	
DriveEndAngry	
DriveEndDefault	
DriveEndHappy	
DriveEndLaunch	
DriveLoopAngry	
DriveLoopDefault	
DriveLoopHappy	
DriveLoopLaunch	
DriveOffChargerFarLeft	
DriveOffChargerFarRight	
DriveOffChargerLeft	
DriveOffChargerRight	
DriveOffChargerStraight	
DriveStartAngry	
DriveStartDefault	
DriveStartHappy	
DriveStartLaunch	
DrivingTo	
ExploringHuhClose	
ExploringHuhFar	
ExploringLookAround	
ExploringLookAtHuman	
ExploringQuickScan	
ExploringReactToHandDrive	
ExploringReactToHandGetIn	
ExploringReactToHandGetOut	
ExploringReactToHandLift	
ExploringReactToHandReaction	
ExploringScanCenterFromLeft	
ExploringScanCenterFromRight	
ExploringScanToLeft	

Trigger Name	Description
ExploringScanToRight	
EyeColorGetIn	
EyeColorGetOut	
EyeColorIdle	
EyeColorSwitch	
EyeContactLookLoop	
FacePlantRoll	
FacePlantRollArmUp	
FailedToRightFromFace	
Feedback_Apology	
Feedback_BadRobot	
Feedback_BeQuiet	This animation is used when Vector enters quiet mode.
Feedback_GoodRobot	
Feedback_ILoveYou	
Feedback_MeanWords	
Feedback_ShutUp	This animation is used when Vector enters quiet mode, after being told to "shut up."
FetchCubeFailure	
FetchCubeSetDown	
FetchCubeSuccess	
FindCubeReactToCube	
FindCubeTurns	
FindCubeWaitLoop	
FistBumpIdle	
FistBumpLeftHanging	
FistBumpRequestOnce	
FistBumpRequestRetry	
FistBumpSuccess	
Flash	
FlipDownFromBack	
FoundFace	
FrustratedByFailureMajor	
GatherCubesAllCubesInBeacon	
GatherCubesCubeInBeacon	
GazingLookAtFacesGetInLeft	
GazingLookAtFacesGetInRight	

Trigger Name	Description
GazingLookAtFacesTurnLeft	
GazingLookAtFacesTurnRight	
GazingLookAtSurfaceReaction	
GazingLookAtSurfaceTurnLeft	
GazingLookAtSurfacesGetInLeft	
GazingLookAtSurfacesGetInRight	
GazingLookAtSurfacesTurnRight	
GazingLookAtVectorReaction	
GoToSleepGetIn	This animation is used when Vector enters sleep mode.
GoToSleepOff	
GoToSleepSleeping	This animation is used while Vector is sleeping.
GreetAfterLongTime	
HeldOnPalmEdgeNervous	
HeldOnPalmEdgeRelaxed	
HeldOnPalmGetInNervous	
HeldOnPalmGetInRelaxed	
HeldOnPalmLookingNervous	
HeldOnPalmNestling	
HeldOnPalmPickupNervous	
HeldOnPalmPickupRelaxed	
HeldOnPalmPutDownNervous	
HeldOnPalmPutDownRelaxed	
HeldOnPalmReactToJolt	
HeldOnPalmRollOff	
HeldOnPalmTransitionToRelaxed	
HighTemperatureWarningFace	This animation is played by *EmergencyModeAnimDispatcher* behavior to show that Vector's battery is too hot.
ICantDoThat	
Idle_09	
InitialWakeUp	This is used by the *InitNormalOperationBehavior*
InteractWithFaceTrackingIdle	
InteractWithFacesInitialNamed	
InteractWithFacesInitialUnnamed	
Interacting	
InteractingBehaviorLock	
InvalidAnimTrigger	

Trigger Name	Description
InvestigateHeldCubeGetIn	
InvestigateHeldCubeGetOutBored	
InvestigateHeldCubeGetOutCubeLost	
InvestigateHeldCubeOnSetDown	
InvestigateHeldCubeTrackingLoop	
KnowledgeGraphAnswer	
KnowledgeGraphGetIn	
KnowledgeGraphGetOut	
KnowledgeGraphListening	
KnowledgeGraphSearching	
KnowledgeGraphSearchingFail	
KnowledgeGraphSearchingFailGetOut	
KnowledgeGraphSearchingGetIn	
KnowledgeGraphSearchingGetOutSuccess	
KnowledgeGraphSuccessReaction	
LookAround	
LookAtDevice	
LookAtDeviceGetIn	
LookAtDeviceGetOut	
LookAtUserEndearingly	
LookInPlaceForFacesBodyPause	
LookInPlaceForFacesBodyPause_Active	
LookInPlaceForFacesHeadMovePause	
LowBattery	
MeetVictor	
MeetVictorConfusion	
MeetVictorDuplicateName	
MeetVictorGetIn	
MeetVictorLookFace	
MeetVictorLookFaceInterrupt	
MeetVictorSawWrongFace	
MeetVictorSayName	
MeetVictorSayNameAgain	
MessagingMessageDeletedShort	
MessagingMessageGetIn	

Trigger Name	Description
MessagingMessageGetOut	
MessagingMessageLoop	
MessagingMessageRecordReaction	
MessagingMessageRewind	
MovementDriveBackward	
MovementDriveForward	
MovementTurnAround	
MovementTurnLeft	
MovementTurnRight	
Muted	
NeutralFace	
NoCloudGetIn	This animation is played when the *NoCloud* behavior starts. Note that this is same as the *NoWifiGetIn* animation group.
NoCloudIcon	This animation is used with the *NoCloud* behavior. This will play the *face_nowifi_trouble_icon* sprite sequence.
NoWifiGetIn	This animation is played when the *NoWifi* behavior starts
NoWifiIcon	This animation is used with the *NoWifi* behavior. This will play the *face_nowifi_icon* sprite sequence.
NoWifiSearching	This animation is used while Vector is looking for a Wifi SSID; This will play the *face_nowifi_signal* sprite sequence.
NothingToDoBoredIdle	
ObservingIdleEyesOnly	This animation is used when Vector is sitting and only looking around.
ObservingIdleWithHeadLookingStraight	
ObservingIdleWithHeadLookingUp	
ObservingLookStraight	
ObservingLookUp	
ObservingOnCharger	
ObservingOnChargerGetIn	
ObservingOnChargerGetOut	
Off	
Offline	
Offline_Off	
Onboarding	
OnboardingComeHere	
OnboardingComeHereGetOut	
OnboardingCubeDriveGetIn	
OnboardingCubeDriveGetOut	

Trigger Name	Description
OnboardingCubeDriveLoop	
OnboardingCubeHuh	
OnboardingDriveOffCharger	
OnboardingDriveOffCharger_1p0	
OnboardingListenGetIn	
OnboardingListenGetOut	
OnboardingLookAround	
OnboardingLookAtPhoneDown	
OnboardingLookAtPhoneLoop	
OnboardingLookAtPhoneUp	
OnboardingLookAtUser	
OnboardingLookAtUserGetOut_1p0	
OnboardingLookDown	
OnboardingLookForCube	
OnboardingReactToFaceHappy	
OnboardingWakeUp	
OnboardingWakeWordGetIn	
OnboardingWakeWordSuccess	
PRDemoGreeting	
PettingBlissGetout	
PettingBlissLoop	
PettingLevel1	
PettingLevel1Getout	
PettingLevel2	
PettingLevel2Getout	
PettingLevel3	
PettingLevel3Getout	
PettingLevel4	
PettingLevel4Getout	
PickupCubePreperation	
PickupCubeRetry	
PickupCubeSuccess	
PlaceCubeByChargerFail	
PlaceCubeByChargerReactToCharger	
PlaceCubeByChargerSuccess	

Trigger Name	Description
PlanningGetIn	
PlanningGetOut	
PlanningLoop	
PlayerCardLayout	
PokeObjectDriveLoop	
PokeObjectGetIn	
PokeObjectGetOut	
PopAWheelieInitial	
PopAWheeliePreActionNamedFace	
PopAWheeliePreActionUnnamedFace	
PopAWheelieRealign	
PopAWheelieRetry	
PounceFail	
PounceSuccess	
PounceWProxForward	
PutDownBlockKeepAlive	
PutDownBlockPutDown	
RTS_OffCharger_Awake_120Left	
RTS_OffCharger_Awake_120Right	
RTS_OffCharger_Awake_150Left	
RTS_OffCharger_Awake_150Right	
RTS_OffCharger_Awake_30Left	
RTS_OffCharger_Awake_30Right	
RTS_OffCharger_Awake_60Left	
RTS_OffCharger_Awake_60Right	
RTS_OffCharger_Awake_Ambient	
RTS_OffCharger_Awake_Back	
RTS_OffCharger_Awake_Front	
RTS_OffCharger_Awake_Left	
RTS_OffCharger_Awake_Right	
RTS_OffCharger_Sleep_120Left	
RTS_OffCharger_Sleep_120Right	
RTS_OffCharger_Sleep_150Left	
RTS_OffCharger_Sleep_150Right	
RTS_OffCharger_Sleep_30Left	

Trigger Name	Description
RTS_OffCharger_Sleep_30Right	
RTS_OffCharger_Sleep_60Left	
RTS_OffCharger_Sleep_60Right	
RTS_OffCharger_Sleep_Ambient	
RTS_OffCharger_Sleep_Back	
RTS_OffCharger_Sleep_Front	
RTS_OffCharger_Sleep_Left	
RTS_OffCharger_Sleep_Right	
RTS_OnCharger_Awake_120Left	
RTS_OnCharger_Awake_120Right	
RTS_OnCharger_Awake_150Left	
RTS_OnCharger_Awake_150Right	
RTS_OnCharger_Awake_30Left	
RTS_OnCharger_Awake_30Right	
RTS_OnCharger_Awake_60Left	
RTS_OnCharger_Awake_60Right	
RTS_OnCharger_Awake_Ambient	
RTS_OnCharger_Awake_Back	
RTS_OnCharger_Awake_Front	
RTS_OnCharger_Awake_Left	
RTS_OnCharger_Awake_Right	
RTS_OnCharger_Sleep_120Left	
RTS_OnCharger_Sleep_120Right	
RTS_OnCharger_Sleep_150Left	
RTS_OnCharger_Sleep_150Right	
RTS_OnCharger_Sleep_30Left	
RTS_OnCharger_Sleep_30Right	
RTS_OnCharger_Sleep_60Left	
RTS_OnCharger_Sleep_60Right	
RTS_OnCharger_Sleep_Ambient	
RTS_OnCharger_Sleep_Back	
RTS_OnCharger_Sleep_Front	
RTS_OnCharger_Sleep_Left	
RTS_OnCharger_Sleep_Right	
ReactToCliff	

Trigger Name	Description
ReactToCliffBack	
ReactToCliffBackLeft	
ReactToCliffBackRight	
ReactToCliffFront	
ReactToCliffFrontLeft	
ReactToCliffFrontRight	
ReactToCliffTurnLeft120	
ReactToCliffTurnLeft180	
ReactToCliffTurnLeft60	
ReactToCliffTurnRight120	
ReactToCliffTurnRight180	
ReactToCliffTurnRight60	
ReactToCubeSearchForCubeLvl1	
ReactToCubeSearchForCubeLvl2	
ReactToCubeSearchForCubeLvl3	
ReactToCubeTapCubeFound	
ReactToCubeTapCubeNotFound	
ReactToCubeTapCubeTappedLvl1	
ReactToCubeTapCubeTappedLvl2	
ReactToCubeTapCubeTappedLvl3	
ReactToCubeTapInteractionGetOut	
ReactToCubeTapInteractionLoop	
ReactToDarkness	
ReactToGoodBye	
ReactToGoodMorning	
ReactToGoodNight	
ReactToGreeting	
ReactToHabitat	
ReactToMotionLeft	
ReactToMotionLeftGetout	
ReactToMotionRight	
ReactToMotionRightGetout	
ReactToMotionTurnLeft	
ReactToMotionTurnRight	
ReactToMotionTurnUp	

Trigger Name	Description
ReactToMotionUp	
ReactToMotionUpGetout	
ReactToObstacle	
ReactToOnLeftSideGetIn	
ReactToOnLeftSideLoop	
ReactToOnRightSideGetIn	
ReactToOnRightSideLoop	
ReactToOnSideEffort	
ReactToOnSideGetOut	
ReactToPerchedOnBlock	
ReactToPickupInitial	
ReactToPickupLoop	
ReactToPutDown	
ReactToShakeSnowGlobe_GetIn	
ReactToShakeSnowGlobe_Lvl1InHand	
ReactToShakeSnowGlobe_Lvl1Loop	
ReactToShakeSnowGlobe_Lvl1OnGround	
ReactToShakeSnowGlobe_Lvl1Waiting	
ReactToShake_GetIn	
ReactToShake_Lvl1InHand	
ReactToShake_Lvl1Loop	
ReactToShake_Lvl1OnGround	
ReactToShake_Lvl1Waiting	
ReactToShake_Lvl2InHand	
ReactToShake_Lvl2Loop	
ReactToShake_Lvl2OnGround	
ReactToShake_Lvl2Waiting	
ReactToShake_Lvl3InHand	
ReactToShake_Lvl3Loop	
ReactToShake_Lvl3OnGround	
ReactToShake_Lvl3Waiting	
ReactToTouchInitial	
ReactToTriggerWordOffChargerBehind	
ReactToTriggerWordOffChargerBehindLeft	
ReactToTriggerWordOffChargerBehindRight	

Trigger Name	Description
ReactToTriggerWordOffChargerFrontLeft	
ReactToTriggerWordOffChargerFrontRight	
ReactToTriggerWordOffChargerLeft	
ReactToTriggerWordOffChargerRight	
ReactToUnclaimedIntent	
ReactToUnclaimedIntentInAir	
ReactToUnexpectedMovement	
RollBlockRealign	
RollBlockRetry	
RollBlockSuccess	
SeasonalHappyHolidays	
SeasonalHappyNewYear	
ShutDown	
Sleep	
SleepNoFade	
SoundOnlyLiftEffortPickup	
SoundOnlyLiftEffortPlaceHigh	
SoundOnlyLiftEffortPlaceLow	
SoundOnlyLiftEffortPlaceRoll	
SpeedTapLose	
SpeedTapWin	
SpinnerBlueCelebration	
SpinnerBlueCycle	
SpinnerBlueHoldTarget	
SpinnerBlueLockIn	
SpinnerBlueLocked	
SpinnerBlueLockedPulse	
SpinnerBlueSelectTarget	
SpinnerGreenCelebration	
SpinnerGreenCycle	
SpinnerGreenHoldTarget	
SpinnerGreenLockIn	
SpinnerGreenLocked	
SpinnerGreenLockedPulse	
SpinnerGreenSelectTarget	

Trigger Name	Description
SpinnerPlayerError	
SpinnerPurpleCelebration	
SpinnerPurpleCycle	
SpinnerPurpleHoldTarget	
SpinnerPurpleLockIn	
SpinnerPurpleLocked	
SpinnerPurpleLockedPulse	
SpinnerPurpleSelectTarget	
SpinnerRedCelebration	
SpinnerRedCycle	
SpinnerRedHoldTarget	
SpinnerRedLockIn	
SpinnerRedLocked	
SpinnerRedLockedPulse	
SpinnerRedSelectTarget	
SpinnerStartGame	
SpinnerYellowCelebration	
SpinnerYellowCycle	
SpinnerYellowHoldTarget	
SpinnerYellowLockIn	
SpinnerYellowLocked	
SpinnerYellowLockedPulse	
SpinnerYellowSelectTarget	
Streaming	
StuckOnEdgeGetIn	
StuckOnEdgeIdle	
StuckOnEdgeLeftGetIn	
StuckOnEdgeLeftIdle	
StuckOnEdgeRightGetIn	
StuckOnEdgeRightIdle	
SuccessfulWheelie	
TakeAPictureCapture	
TakeAPictureFocusing	
TapResponsePulse	
TemperatureDoubleDig	

Trigger Name	Description
TemperatureNegDoubleDig	
TemperatureNegSingleDig	
TemperatureNegTripleDig	
TemperatureSingleDig	
TemperatureTripleDig	
TestAllLeds	
TestOffset	
TestRotation	
TextToSpeechGetIn	
TextToSpeechGetLoop	
TextToSpeechGetOut	
TimerCancelGetIn	
TimerCancelTimer	
TimerCheckTimeGetIn	
TimerCheckTimeGetOut	
TimerRing	
TimerRingGetIn	
TimerRingGetOut	
TimerSetGetIn	
TimerSetGetOut	
UnitTestAnim	
VC_IntentNeutral	
VC_ListeningGetIn	
VC_ListeningGetOut	
VC_ListeningLoop	
VC_SleepingToListeningGetIn	
VC_SleepingToListeningGetOut	
VC_SleepingToListeningLoop	
Visible	
VolumeLevel1	
VolumeLevel2	
VolumeLevel3	
VolumeLevel4	
VolumeLevel5	
WakeUp	

Trigger Name	Description
WakeupGetout	
WeatherCondCloudy_01	
WeatherCondColdClear_01	
WeatherCondRain_01	
WeatherCondSnow_01	
WeatherCondStars_01	
WeatherCondSunny_01	
WeatherCondThunderstorms_01	
WeatherCondWindy_01	

13.2 Backpack Lights

Summary of the back lights

Things that this could do:

- Diagram of the backpack lights

- Show the FAC lights

- Changes to the backpack lights in the custom software

See also DDL.

13.3 Purple circle light

- For the first few seconds at boot, this is normal and should get fixed later on in the body board boot process. However, if your Vector is stuck on it and he shows an error code (801, 898, 899), there could be an issue. First try to reboot by holding the button for 5-6 seconds. If that doesn't work, leave him and let the battery die. This will probably take a few hours. After the battery dies, turn him back on. If he still shows a purple light after being turned back on, then there is a hardware issue on the bodyboard.

13.4 Behaviour IDs

Behavior ID	Description
AcknowledgeCharger	
AcousticTestMode	This behavior is the first behavior called when Vector starts in an acousting testing mode
ActiveLookForFaces	
AlexaSignInOut	
Alexa	
AskForHelpOnSide	
AskForHelp	
Asleep	
BasicVoiceCommands	
BeQuietAnims	This behavior is used to animate Vector going into a quiet state, and animate his eyes looking around. See quiet mode
BeQuietLoop	This behavior is used to animate Vector's eyes looking around. See quiet mode
BlackJackGoodLuckTTS	
BlackJackHandleRTPResponses	
BlackJackHitOrStandPrompt	
BlackJackLookAtFaceInFront	
BlackJackRequestToPlayAgain	
BlackJackRequestToPlay	
BlackJackTextToSpeech	
BlackJackVoiceCommand	
BlackJack	
ChangeEyeColor	
CheckForAndReactToHand	
ClearChargerArea	
ComeHereVoiceCommand	
ConfirmCharger	
ConfirmCube	
ConfirmHabitat	
ConnectToCube	
CoordinateGlobalInterrupts	
CoordinateInHabitat	
CoordinateWhileHeldInPalm	
CoordinateWhileInAir	
CubeSpinnerConnectionGate	
CubeSpinnerLookAroundInPlace	
CubeTrickDispatcher	

Behavior ID	Description
DanceBig	
DanceForwardBackFlower	
DanceFrontRightLeftPoint	
DanceSTwoways	
DanceSwell	
DanceToTheBeatCoordinator	
DanceToTheBeatVoiceCommand	
DanceToTheBeat	
DanceWiggleForwardWiggleBack	
DefaultTextToSpeechLoop	
DemoTimerUtilityCoordinator	
DevBaseBehavior	
DevBatteryLogging	
DevCubeSpinnerConsole	
DevCubeSpinner	
DevDesignCubeLights	
DevDisplayReadingsOnFace	
DevEventSequenceCapture	
DevImageCapture_PetsAndHands	
DevImageCapture	
DevPlannerTest	
DevSquawkBoxTest	
DevTestBlackjackViz	
DevTestConnectToCube	
DevTestPersonDetectorBehavior	
DevTestPromptUser	
DevTouchDataCollection	
DevTurnInPlaceTest	
DevViewCubeBackpackLights	
DoATrickVoiceCommand	
DockingTestSimple	
DriveOffChargerCube	
DriveOffChargerFace	
DriveOffChargerIntoSocializing	
DriveOffChargerRandomlyAnim	

Behavior ID	Description
DriveOffChargerRandomly	
DriveOffChargerStraight	
EmergencyModeAnimDispatcher	This behavior gives a visual animation why Vector about the emergency mode.
EmergencyModeFindAndGoToHome	This specific behavior initiates the *FindAndGoToHome* behavior when in emergency mode.
EmergencyModeInAir	This behavior animates a request to be picked up an placed in the dock when he has fallen or picked up.
EmergencyModeOffCharger	This behavior coordinates driving back to the charging dock
EmergencyModeTriggerWord	This animates a response to the trigger word, usually why it can't respond to commands right now.
EmergencyMode	This behavior coordinates Vector driving to the charging dock, or requesting help.
ExploringBumpObject	
ExploringExamineObstacle	
ExploringGetIn	
ExploringReferenceHuman	
ExploringVoiceCommand	
Exploring	
FactoryCentroidExtractor	
FetchCubeVoiceCommand	
FetchCube	
FindAndGoToHome	
FindAndRequestHome	
FindCubeAndPlayKeepaway	
FindCubeAndThen	
FindCube	
FindFacesFetchCube	
FindFacesPhoto	
FindHomeForSleeping	
FindHomeInHabitat	
FindHome	
FindYourCubeVoiceCommand	
FistBumpVoiceCommand	
FistBump	
ForceStuckOnEdge	
FrameFaces	
GlobalInterruptions	
GoHomeVoiceCommand	

Behavior ID	Description
GoHome	
GoToSleep	
GreetAfterLongTime	
HabitatMutedDispatcher	
HabitatMutedVoiceCommandResponse	
HeldInPalmDispatcher	
HeldInPalmResponses	
HighLevelAI	
HowOldAreYouCounting	
HowOldAreYou	
InitNormalOperation	This behavior is the first behavior called when Vector starts normally.
InitPRDemo	This behavior is the first behavior called when Vector starts in a PR demo mode.
InitialHeldInPalmReaction	
InitialPickupAnimation	
IntentUnmatched	
InteractWithFaces	
InteractWithStaticCube	
InterruptingVoiceReactions	
InvestigateCubeConnectionGate	
InvestigateHeldCube	
KeepawayVoiceCommand	
Keepaway	
KnowledgeGraphQuestion	
KnowledgeGraphTTS	
LeaveAMessage	
LiftLoadTest	
ListenForBeatsLong	
ListenForBeatsVoiceCommand	
ListenForBeats	
LookAtMeVoiceCommand	
LookInPlaceHeadDownInAir	
LookInPlaceHeadUpInAir	
LookInPlaceHeadUp	
LookOverThereVoiceCommand	
MandatoryPhysicalReactions	

Behavior ID	Description
MeetVictorAlreadyKnowYouPrompt	
MeetVictor	
MessagingPlaybackTTS	
MessagingRecordTTS	
ModeSelector	Top level dispatcher; this is called by many different start up modes. See power management for a description.
MoveCube	
MovementBackward	
MovementForward	
MovementTurnAround	
MovementTurnLeft	
MovementTurnRight	
NoCloud	The behavior is invoked when Vector can't reach the voice server; see Communication trouble behaviors
NoWifi	The behavior is invoked when Vector can't connect to a Wifi SSID; see Communication trouble behaviors
NormalWakeUp	This is called by *InitNormalBehavior* on start. It plays the wake up animation --- if it isn't night time, and this isn't a maintenance reboot.
NothingToDo_Idle	
ObservingDriveOffCharger	
ObservingEyeContact	
ObservingFindFaces	
ObservingLookAtFacesInAir	
ObservingLookAtFaces	
ObservingOffChargerHeadOnly	
ObservingOnChargerEyeContact	
ObservingOnChargerGetIn	
ObservingOnChargerGetOut	
ObservingOnChargerIdleAnim	
ObservingOnChargerIdle	
ObservingOnCharger	
Observing	
OnboardingComeHere	
OnboardingEmulate1p0WaitForVC	
OnboardingLookAtPhone	
OnboardingLookAtUserOffCharger	
OnboardingLookAtUserOnCharger	

Behavior ID	Description
OnboardingLookAtUser	
OnboardingPowerOff	
OnboardingTeachComeHere	
OnboardingTeachMeetVictor	
OnboardingTeachWakeWord	
OnboardingWakeUp	
Onboarding	This behavior is the first behavior called when Vector starts "fresh" (new from factory or a clearing of user data) and is now onboarding a new human companion.
PRDemoBigGreeting	
PRDemoComeHere	
PRDemoExploring	
PRDemoObserving	
PRDemoSleeping	
PRDemoStateMachine	
PickUpCubeVoiceCommand	
PickupCubeNoInitialReaction	
PickupCube	
PlaceCubeByCharger	
PlayAGameVoiceCommand	
PlayRollBlock	
PlayWithCube	
PlaybackMessage	
PlaypenCameraCalibration	
PlaypenDistanceSensor100mm	
PlaypenDistanceSensor300mm	
PlaypenDistanceSensor80mm	
PlaypenDriftCheck	
PlaypenDriveForwards	
PlaypenEndChecks	
PlaypenInitChecks	
PlaypenMotorCalibration	
PlaypenPickupCube	
PlaypenSoundCheck	
PlaypenTest	
PlaypenWaitToStart	
PopAWheelieVoiceCommand	

Behavior ID	Description
PopAWheelie	
PowerSaveStressTest	
PowerSaveTest	
ProceduralTurnToMicDirection	
PutDownBlockAtPose	
PutDownBlock	
PutDownDispatch_LookForFaceAndCube	
PuzzleMaze	
QuietModeEmergencyModeGoHome	This behavior coordinates driving back to the charging dock when the battery is low or overheated in quiet mode. See power managent for more details.
QuietMode	The *QuietMode* behavior is when Vector's has been asked to be silent. See quiet mode
ReactToAbuse	
ReactToAffirmative	
ReactToApology	
ReactToBatteryTooHotToCharge	
ReactToBody	
ReactToCliffDuringFetch	
ReactToCliff	
ReactToDarkness	
ReactToFrustrationMajor	
ReactToGazeDirectionSurface	
ReactToGazeDirection	
ReactToGoodBye	
ReactToGoodMorning	
ReactToHand	
ReactToHello	
ReactToJoltInPalm	
ReactToLove	
ReactToMotion	
ReactToMotorCalibration	
ReactToNegative	
ReactToObstacle	
ReactToPalmTilt	
ReactToPickupFromPalm	
ReactToPlacedOnSlope	
ReactToPutDownFromPalm	

Behavior ID	Description
ReactToPutDown	
ReactToRobotOnBack	
ReactToRobotOnFace	
ReactToRobotOnSide	
ReactToRobotShakenSnowGlobe	
ReactToRobotShaken	
ReactToSoundAsleep	
ReactToSoundAwake	
ReactToSoundDirectionAsleep	
ReactToSoundDirectionAwake	
ReactToTouchPetting	
ReactToTriggerDirectionAwake	
ReactToUncalibratedHeadAndLift	
ReactToUnclaimedIntent	
ReactToUnexpectedMovement	
RequestHomeBecauseStuck	
RequestToGoHome	
ResetSafely	
RespondToRenameFace	
RollBlockIfNotVertical	
RollCubeVoiceCommand	
SDKDefault	
SDKOverrideAll	
SayName	
SearchWithinBoundingBox	
SeasonalHappyHolidays	
SeasonalHappyNewYear	
SelfTestButton	
SelfTestDockWithCharger	
SelfTestDriftCheck	
SelfTestDriveForwards	
SelfTestInitChecks	
SelfTestLookAtCharger	
SelfTestMotorCalibration	
SelfTestPickup	

Behavior ID	Description
SelfTestPutOnCharger2	
SelfTestPutOnCharger	
SelfTestScreenAndBackpack	
SelfTestSoundCheck	
SelfTestTouch	
SelfTest	
ShortLookAroundForFaceAndCube	
ShowWallTime	
ShutUpAnims	This behavior is used to animate Vector going into a quiet state (after beint told to shutup), and animate his eyes looking around. See quiet mode
ShutUpMode	The *ShutUpMode* behavior is when Vector's has been asked to "shut up." See quiet mode
SingletonAnticShowClock	
SingletonCancelTimer	
SingletonFindFaceInFrontWallTime	
SingletonICantDoThat	
SingletonPounceApproachWithProx	
SingletonPounceDispatcher	
SingletonPounceTurnLeft	
SingletonPounceTurnRight	
SingletonPounceWithProx	
SingletonPoweringRobotOff	This behavior is active when Vector is powering down. See power management
SingletonTimerAlreadySet	
SingletonTimerAntic	
SingletonTimerCheckTime	
SingletonTimerRinging	
SingletonTimerSet	
SingletonWallTimeCoordinator	
SleepCycle	This behavior is manages Vector going to sleep, playing and interacting. See power management
SleepingPersonCheck	
SleepingTriggerWord	
SleepingWakeUpLights	
SleepingWakeUp	
SocializeGame	
Socialize	
StayOnChargerUntilCharged	

Behavior ID	Description
StuckOnEdge	
TakeAPhotoCoordinator	
TestStackMonitors	
TimerRingingPRDemo	
TimerUtilityCoordinator	
TrackCubeTest	
TrackCube	
TrackFaceTest	
TrackingEyeContact	
TriggerWordDetected	
TriggerWordWithoutIntent	
TurnToLastFace	
UserDefinedBehaviorSelector	
UserDefinedBehaviorTreeConfirmNewBehavior	
UserDefinedBehaviorTreeRouter	
UserDefinedBehaviorTreeTextToSpeech	
VectorPlaysCubeSpinner	
Volume	
Wait	
WeatherCloudyGeneric	
WeatherColdClearGeneric	
WeatherRainGeneric	
WeatherResponses	
WeatherSnowGeneric	
WeatherStarsGeneric	
WeatherSunnyGeneric	
WeatherTextToSpeech	
WeatherThunderstormsGeneric	
WeatherWindyGeneric	
WhatsMyNameVoiceCommand	
WhileInAirDispatcher	
WhileInAirResponsesPRDemo	
WhileInAirResponses	
WiggleBackOntoChargerFromPlatform	

13.5 Behavior Classes

Behavior Classes	Description
AdvanceClock	
AestheticallyCenterFaces	
Alexa	
AlexaSignInOut	
AnimGetInLoop	
AnimSequence	This kind of behavior plays an animation.
AnimSequenceWithFace	
AnimSequenceWithObject	
AskForHelp	
AttentionTransferIfNeeded	
BlackJack	
BumpObject	
CheckForAndReactToSalientPoint	
ClearChargerArea	
ConfirmHabitat	
ConfirmObject	
ConnectToCube	
CoordinateGlobalInterrupts	
CoordinateInHabitat	
CoordinateWeather	
CoordinateWhileHeldInPalm	
CoordinateWhileInAir	
CountingAnimation	
DanceToTheBeat	
DanceToTheBeatCoordinator	
DevBatteryLogging	
DevCubeSpinnerConsole	
DevDesignCubeLights	
DevDisplayReadingsOnFace	
DevEventSequenceCapture	
DevImageCapture	
DevSquawkBoxTest	
DevTestBlackjackViz	
DevTouchDataCollection	
DevTurnInPlaceTest	

Behavior Classes	Description
DevViewCubeBackpackLights	
DispatchAfterShake	
DispatcherPassThrough	
DispatcherQueue	This behavior runs each of the behaviors in the `behavior` array in order.
DispatcherRandom	
DispatcherStrictPriority	This behavior runs each of the behaviors in the `behavior` array in order.
DispatcherStrictPriorityWithCooldown	This behavior runs each of the behaviors in the `behavior` array in order. Behaviors still in a cooldown period are skipped.
DisplayWallTime	
DisplayWeather	
DockingTestSimple	
DriveOffCharger	
DriveToFace	
EnrollFace	
Exploring	
ExploringExamineObstacle	
EyeColor	
FactoryCentroidExtractor	
FetchCube	
FindCube	
FindCubeAndThen	
FindFaceAndThen	
FindFaces	
FindHome	
FistBump	
GoHome	
GreetAfterLongTime	
HighLevelAI	
HowOldAreYou	
InspectCube	
InteractWithFaces	
Keepaway	
KnowledgeGraphQuestion	
LeaveAMessage	
LiftLoadTest	
ListenForBeats	

Behavior Classes	Description
LookAroundInPlace	
LookAtFaceInFront	
LookAtMe	
LookForFaceAndCube	
MoveHeadToAngle	
ObservingLookAtFaces	
ObservingWithoutTurn	
OnboardingCoordinator	
OnboardingEmulate1p0WaitForVC	
OnboardingLookAtPhone	
OnboardingLookAtUser	
OnboardingTeachWakeWord	
OnboardingWakeUp	
PickUpCube	
PlaceCubeByCharger	
PlannerTest	
PlaybackMessage	
PlaypenCameraCalibration	
PlaypenDistanceSensor	
PlaypenDriftCheck	
PlaypenDriveForwards	
PlaypenEndChecks	
PlaypenInitChecks	
PlaypenMotorCalibration	
PlaypenPickupCube	
PlaypenSoundCheck	
PlaypenTest	
PlaypenWaitToStart	
PopAWheelie	
PounceWithProx	
PowerSaveStressTest	
PowerSaveTest	
PoweringRobotOff	This behavior is active when Vector is powering down. See power management
PRDemo	
PRDemoBase	

Behavior Classes	Description
ProceduralClock	
PromptUserForVoiceCommand	
ProxGetToDistance	
PutDownBlock	
PutDownBlockAtPose	
PuzzleMaze	
QuietModeCoordinator	
ReactToBatteryTooHotToCharge	
ReactToBody	
ReactToCliff	
ReactToDarkness	
ReactToFrustration	
ReactToGazeDirection	
ReactToHand	
ReactToMicDirection	
ReactToMotion	
ReactToMotorCalibration	
ReactToPlacedOnSlope	
ReactToPutDown	
ReactToRobotOnBack	
ReactToRobotOnFace	
ReactToRobotOnSide	
ReactToRobotShaken	
ReactToSound	
ReactToTouchPetting	
ReactToUncalibratedHeadAndLift	
ReactToUnclaimedIntent	
ReactToUnexpectedMovement	
ReactToVoiceCommand	
RequestToGoHome	
ResetState	
RespondToRenameFace	
RollBlock	
SayName	
SDKInterface	

Behavior Classes	Description
SearchWithinBoundingBox	
SelfTest	
SelfTestButton	
SelfTestDockWithCharger	
SelfTestDriftCheck	
SelfTestDriveForwards	
SelfTestInitChecks	
SelfTestLookAtCharger	
SelfTestMotorCalibration	
SelfTestPickup	
SelfTestPutOnCharger	
SelfTestScreenAndBackpack	
SelfTestSoundCheck	
SelfTestTouch	
SleepCycle	This behavior is manages Vector going to sleep, playing and interacting. See power management
Sleeping	
StayOnChargerUntilCharged	
TakeAPhotoCoordinator	
TextToSpeechLoop	
TimerUtilityCoordinator	
TrackCube	
TrackFace	
Turn	
TurnToFace	
UserDefinedBehaviorSelector	
UserDefinedBehaviorTreeRouter	
VectorPlaysCubeSpinner	
Volume	
Wait	
WallTimeCoordinator	
WiggleOntoChargerContacts	

13.6 Behavior Tree

Figure: Vector Behavior Tree v1.0.1

Note: all of the behavior related files are in the following directory, and sub-directories:

```
/anki/data/assets/cozmo_resources/config/engine/behaviorComponent
```

Habitat: I am not sure if this term refers to the desk / table in general, or is specifc to the Vector Field (tray).

These are not referred to by the rest of the behavior tree, but the names are in `libcozmo_engine`. I am not sure if these are unused, or invoked internally by the C++. If they are, the C++ should be refactored to use the behavior tree.

13.6.1 Self-maintenance behaviors

Self-maintenance behaviors:

- Startup related to Vector turning on and settig up the behavior tree.

- Power management related to turning on and off, initiating return to charger when the battery is low, as well as self-protection behaviors like very low battery, and over temperature.

- Reacting to environmental conditions while driving around

- Drive home

- Communication trouble behaviors are invoked when Vector can't connect to a Wifi SSID or can't reach the voice server.

- Quiet mode related to Vector being quiet -- not interacting with a person or toy, but also not asleep.

- Sleep

13.6.2 Social behaviors

- Petting

- Being held

- Playing

- Helping out

  - Question-Answer

  - Timer

  - Weather

13.6.3 Toys and Accessories

13.6.4 Change history synopsis

Date	Change
2020-12-1	Created

13.7 Console Variables

Console Variables are part of the developer build. They allow the developer to test, diagnose, and tweak (inject data into) the various modules.

This note is to help gather a description of each of console variables. This format lets us gather information on them, and help understand where they fit in.

These tables are not suitable for the TRM at this time; they may go better in the software design description in the future.

Note: the k seems to be dropped or optional in matching

A/B Testing console variables

Variable	Type	Units	Description
kForceDisableABTesting			

AIWhiteboard console variables

Variable	Type	Default	Description
kAI_MaxExtraExploringCooldown_s	float	800.0	
kBW_DebugRenderPossibleObjects	bool	true	
kBW_DebugRenderPossibleObjectsZ	float	35.0	
kBW_MaxHeightForPossibleObject_mm	float	30.0	
kBW_MaxPossibleObjects	unsigned	10	
kBW_PossibleObjectClose_mm	float	50.0	
kBW_PossibleObjectClose_rad	float	3.141592653	Current objects flip due to distance
kExploringCooldownUpdatePeriod_s	float	60.0	
kFlatPosisbleObjectTol_deg	float	10.0	

13.7.1 Animation

Animation console variables

Variable	Type	Default	Description
kEyeDartFocusValue_pix	float	1.0	
kIgnoreAnimWhitelist	bool	false	
kShouldPreCacheSprites	bool	false	

AnimationStreamer console variables

Variable	Type	Default	Description
kEnableBackpackLightsTrack	bool	false	
kShouldDisplayPlaybackTime	bool	false	
kDisplayCPUThrottling	bool	false	Whether or not to display CPU throttling. This is disabled by default since current OS doesn't throttle for thermal reasons and sporadic idle throttling is not worth alerting the dev about.
kDisplayHighTemperature	bool	true	Whether or not to display high temperature indicator on face

AnimationStreamer.System console variables

Variable	Type	Default	Description
kDisplayCPUThrottling			
kDisplayHighTemperature			
kDisplayMemoryPressure	bool	true	
kThermalAlertTemp_C	unsigned	90	Temperature beyond which the thermal indicator is displayed on face.

ManualAnimationPlayback console variables

Variable	Type	Default	Description
kNumberOfFramesToIncrement	unsigned	1	
kShouldDisplayKeyframeNumber	bool	false	

BackpackLights console variables

Variable	Type	Default	Description
kOfflineCheckFreq_ms	unsigned	5000	
kOfflineTimeBeforeLights_ms	unsigned	120000	

DevViewLights console variables

Variable	Type	Default	Description
kBackpackTriggerIdx	unsigned	0	
kCubeTriggerIdx	unsigned	0	

GlitchLights console variables

Variable	Type	Default	Description
kGlitchLightDelay_ms	unsigned	200	How long to wait before the lights should start glitching.
kGlitchLightDuration_ms	unsigned	60	Duration of each glitchy backpack light keyframe.

Face.KeepAlive console variables

Notes: The Medium distance eye dart params are used when the dart's length is larger than threshold. These darts have a single interpolation frame (with associated dart distance and squash fractions)

The Long distance eye dart params (when dart's length is larger than threshold). These darts have two interpolation frames (with associated dart distance and squash fractions).

Variable	Type	Default [Min .. Max]	Description
kKeepAliveBlink_SpacingMaxTime_ms	int	10000 [0 .. 30000]	
kKeepAliveBlink_SpacingMinTime_ms	int	3000 [0 .. 30000]	
kKeepAliveEyeDart_DownMinScale	float	0.9 [0.5 .. 1.0]	
kKeepAliveEyeDart_HotSpotPositionMultiplier	float	1.5 [0.5 .. 10.0]	
kKeepAliveEyeDart_LongDistanceThresh_pix	int	10 [0 .. 92]	
kKeepAliveEyeDart_LongShiftFraction1	float	0.2 [0.0 .. 1.0]	
kKeepAliveEyeDart_LongShiftFraction2	float	0.4 [0.0 .. 1.0]	
kKeepAliveEyeDart_LongSquashFraction1	float	0.7 [0.5 .. 1.0]	
kKeepAliveEyeDart_LongSquashFraction2	float	0.85 [0.5 .. 1.0]	
kKeepAliveEyeDart_MaxDistFromCenter_pix	int	15 [0 .. 92]	
kKeepAliveEyeDart_MaxDistFromCenterFocused_pix	int	1 [0 .. 92]	
kKeepAliveEyeDart_MediumDistanceThresh_pix	int	5 [0 .. 92]	
kKeepAliveEyeDart_MediumShiftFraction	float	0.2 [0.0 .. 1.0]	
kKeepAliveEyeDart_MediumSquashFraction	float	0.85 [0.5 .. 1.0]	
kKeepAliveEyeDart_OuterEyeScaleIncrease	float	0.03 [0.0 .. 0.2]	
kKeepAliveEyeDart_ShiftLagFraction	float	0.4 [0.0 .. 1.0]	
kKeepAliveEyeDart_SpacingMaxTime_ms	int	2250 [0 .. 10000]	
kKeepAliveEyeDart_SpacingMinTime_ms	int	1000 [0 .. 10000]	Global keep-alive eye dart params (spacing, distance, scaling)
kKeepAliveEyeDart_UpMaxScale	float	1.05 [1.0 .. 1.2]	
kMaxBlinkSpacingTimeForScreenProtection_ms	float	30000	

Face.ParameterizedFace console variables

Variable	Type	Default [Min .. Max]	Description
kProcFace_AntiAliasingFilter		Filter	Box Filter
kProcFace_AntiAliasingSigmaFraction	float	0.5 [0.0 .. 1.0]	
kProcFace_AntiAliasingSize	int	3 [0 .. 15]	full image antialiasing, 3 will use NEON
kProcFace_DefaultScanlineOpacity	float	1.0 [0.0..1.0]	
kProcFace_Display	FaceDisplayType	Normal	This overrides whatever faces we're sending with a with ConsoleVars edited parameters. The FaceDisplayType for values and their meaning.
kProcFace_EllipseDelta	int	10 [1 .. 90]	
kProcFace_EnableAntiAliasing	bool	true	
kProcFace_EyeLightnessMultiplier	float	1.0 [0.0 .. 2.0]	
kProcFace_Gamma	float	1.0 [1.0 .. 4.0]	
kProcFace_GammaType	FaceGammaType	None	
kProcFace_GlowFilter	Filter	Box Filter	
kProcFace_GlowLightnessMultiplier	float	1.0 [0.0 .. 10.0]	
kProcFace_GlowSizeMultiplier	float	1.0 [0.0 .. 1.0]	
kProcFace_HotspotFalloff	float	0.48 [0.05 .. 1.0]	
kProcFace_HotspotRender	bool	true	If true, the glow is rendered
kProcFace_InterpolationType	unsigned	1	Nearest, Linear, Cubic, Area, Lanczos, LinearExact, Max, WarpFillOutliers
kProcFace_LineType	unsigned	1	
kProcFace_NoiseMaxLightness	float	1.14 [0.0 .. 2.0]	
kProcFace_NoiseMinLightness	float	0.92 [0.0 .. 2.0]	
kProcFace_NoiseNumFrames	int	5 [0 .. 7]	
kProcFace_NominalEyeSpacing	int	92 [-184 .. 184]	
kProcFace_Scanlines	bool	false	

ENUM FILTER

Name	Value	Description
None	0	
Box Filter	1	
Gaussian Filter	2	

ENUM FACEDISPLAYTYPE

Name	Value	Description
Normal	0	
Test	1	Displays pattern showcasing RGB at different intensities
Full White	2	Displays a completely white image
Override Individually	3	each eyes parameters operate on their respective eye
Override Together	4	left eye parameters drive both left and right eyes

ENUM FACEGAMMATYPE

Name	Value	Description
None	0	
FromLinear	1	
ToLinear	2	
AddGamma	3	Use value of kProcFace_Gamma
RemoveGamma	4	Use value of kProcFace_Gamma
Custom	5	

Face.ScanlineDistortion console variables

Variable	Type	Default [Min .. Max]	Description
kProcFaceScanline_MaxShiftNoise	int	3	Max amount to randomly shift control-point distortion shifts left and right, per scanline.
kProcFaceScanline_OffNoiseMaxWidth	int	3	Max width of each "off" noise bar.
kProcFaceScanline_OffNoiseProb	float	0.1 [0.0 ..1.0]	Fraction of (nominal) eye area to be off (note: does not consider "Width" parameter below).

FaceInfoScreenManager console variables

Variable	Type	Default [Min .. Max]	Description
kAlexaNotificationTimeout_s	float	2.0 [0.001 .. 3.0]	
kButtonPressDurationForShutdown_ms	unsigned	500	How long the button needs to be pressed for before it should trigger shutdown animation.
kFakeButtonPressType	int	0	Fake one of several types of button presses. This value will get reset immediately, so to run it again from the web interface, first set it to NoOp. NoOp, singlePressDetected, doublePressDetected
kToggleMuteTimeout_s	float	1.2 [0.001 .. 3.0]	

FaceSelection console variables

Variable	Type	Default	Description
kFaceSelectionDebugging	bool	false	

13.7.2 Audio Input

MicData console variables

Variable	Type	Default [Min .. Max]	Description
kBeatDetectorUseProcessedAudio	bool	true	
kDevForceProcessState	MicProcessingState	NormalOperation	See enumeration below
kMicData_ClipRecordTime_ms	unsigned	4000 [500 .. 15000]	
kMicData_CollectRawTriggers	bool	false	
kMicData_ForceDisableMicDataProc	bool	false	
kMicData_ForceEnableMicDataProc	bool	false	
kMicData_QuietTimeCooldown_ms	unsigned	1000 [500 .. 10000]	Time necessary for the VAD logic to wait when there's no activity, before we begin skipping processing forperformance. Note that this probably needs to at least be as long as the trigger, which is ~ 500-750ms.
kMicData_SaveRawFullIntent	bool	false	
kMicData_SaveRawFullIntent_WakeWordless	bool	false	
kMicData_SpeakerNoiseDisablesMics	bool	true	
kSaveNotches	bool	false	
kTempoCorrectionScaleFactor	float	0.988	

ENUM MICPROCESSINGSTATE

Name	Value	Description
None	0	Raw single mic data
NoProcessingSingleMic	1	Cheap single mic processing
SigEsBeamformingOff	2	Signal Essence fall back policy, clean & mix mics
SigEsBeamformingOn	3	Signal Essence beamforming processing

Audio.Microphone console variables

Variable	Type	Default [Min .. Max]	Description
kNoiseFloorMin	float	1.5 [0.0 .. 10.0]	
kNoiseFloorRange	float	5.5 [0.0 .. 10.0]	

Alexa console variables

Variable	Type	Default [Min .. Max]	Description
kAcousticTestMode	bool	false	
kAlexaEnabledInAU	bool	true	
kAlexaEnabledInUK	bool	true	
kAlexaHackCheckForSystemClockSyncPeriod_s	float	5.0	
kAlexaIdleDelay_s	float	2.0 [0.0 .. 10.0	
kAlexaMaxIdleDelay_s	float	3.0 [0.0 .. 10.0]	
kAllowAudioOnCharger	bool	true	
kDEV_ONLY_EnableAlexaTemplateRendererStub	bool	false	
kLogAlexaDirectives	bool	false	
kNotchPower	float	-0.41 [-1 .. 0]	
kSaveAlexaAudio	bool	false	

Alexa.Init console variables

Variable	Type	Default	Description
kDumpAlexaTriggerAudio	bool	false	

Alexa.Messaging console variables

Variable	Type	Default	Description
kLogAlexaMessages			
kStealAlexaWakewordAudio	bool	false	

BeatDetectorComponent

Variable	Type	Default [Min .. Max]	Description
kBeatHistoryWindowSize_sec	float	10. [1 .. 60]	
kConfidenceThreshold	float	0.18 [0.01 .. 1.]	
kFakeBeat_bpm	float	-1. [-1. .. 200.]	
kFakeBeatConfidence	float	0.50 [0. .. 100.]	
kHighConfidenceThreshold	float	0.75 [0.01 .. 20.]	
kMinNumBeatsInHistory	int	6 [2 .. 50]	
kPossibleBeatWindow_sec	float	9. [1. .. 10.]	
kTempoSteadyThreshold_bpm	float	5 [1 .. 25]	

SoundReaction console variables

Variable	Type	Default [Min .. Max]	Description
kRTS_AbsolutePowerThreshold_display	float	2.9	
kRTS_MaxReactionTime_s	float	1.00	"Wwe have this much time to respond to a sound
kRTS_MinPowerThreshold_display	float	1.5	
kRTS_PowerAvgNumSamples	unsigned	100 1 .. 250]	
kRTS_ReReactionCooldown_s	float	0.25	we have this much time to respond to a sound
kRTS_WebVizUpdateInterval	float	0.2 [0 .. 1]	

13.7.3 SpeechRecognizer console variables

Variable	Type	Default	Description
kSuppressTriggerResponse	bool	false	

SpeechRecognizer.Alexa console variables

Variable	Type	Default [Min .. Max]	Description
kAlexaRecognizerModel	int		enUS, enUK, enAU, frFR, deDE
kDefaultDetectThreshold	int	250 [0 .. 1000]	
kForceRunNotchDetector	unsigned int	0 [0 .. 2]	
kSaveRawMicInput	bool	false	

SpeechRecognizer.AlexPlayback console variables

Variable	Type	Default [Min .. Max]	Description
kAlexaPlaybackRecognizerModel	int		
kPlaybackRecognizerSampleCountThreshold	unsigned	5000 [1000 .. 10000]	

SpeechRecognizer.Vector console variables

Variable	Type	Units	Description
kVectorRecognizerModel			
kVectorRecognizerModelSensitivity			

TriggerWord console variables

Variable	Type	Default [Min .. Max]	Description
kDirStreamingConfToIgnore	int	500 [0 .. 10000]	Confidence levels below this will be ignored.
kDirStreamingTimeToIgnoreBegin	float	0.5 [0 .. 2]	When our streaming begins/ends there is a high chance that we will record some non-intent sound, these values allow us to chop off the front and back of the streaming window when determining the intent direction
kDirStreamingTimeToIgnoreEnd	float	1.25 [0 .. 2]	Ignore mic direction with confidence below this when trying to determine streaming direction.
kMaxStreamingDuration_s	float	10.0 [0.0 .. 20.0]	This is the maximum duration we'll wait from streaming begin.
kMinListeningTimeout_s	float	5.0 [0.0 .. 30.0]	The behavior will always "listen" for at least this long once it hears the wakeword, even if we receive an error sooner than this. Note that the behavior will also consider the intent to be an error if the stream doesn't open within this amount of time, so don't lower this number too much.
kRecentDirFallbackTime	real	1 [0 .. 10]	If we cannot determine the mic direction, we fall back to the most recent direction this allows you to specify how far back we sample for the most recent direction.
kRespondsToTriggerWord	bool	true	
kTriggerWord_FakeError	bool	false	Pretend all responses are errors: NOTE intents may still get processed with this set true, recommendation is to use silence or a known mismatch intent (my favorite happens to be "potatoes").
kTriggerWord_FakeError_HasWifi	bool	false	

13.7.4 Audio Output

Audio.AnimationStream console variables

Variable	Type	Default [Min .. Max]	Description
kAudioAnimationOffset_ms	unsigned	200 [0 .. 300]	

13.7.5 Audio.Controller

Variable	Type	Default [Min .. Max]	Description
kWriteAudioOutputCapture	bool	false	
kWriteAudioOutputMaxLogCount	unsigned	1 [1 .. 5]	
kWriteAudioProfilerCapture	bool	false	
kWriteAudioProfilerMaxLogCount	unsigned	3 [1 .. 5]	

Audio.KeepAlive console variables

Variable	Type	Default	Description
kEnableKeepAliveEyeBlinkAudioEvents	bool	true	
kEnableKeepAliveEyeDartAudioEvents	bool	true	
kEnableKeepAliveEyeSquintAudioEvents	bool	true	

Audio.Procedural console variables

Variable	Type	Default [Min .. Max]	Description
kEnableHeadProceduralMovement	bool	false	
kEnableLiftProceduralMovement	bool	false	
kEnableRobotStateLog	bool	false	
kEnableTreadProceduralMovement	bool	true	
kHeadCoolDown_ms	unsigned	65 [0 .. 250]	
kHeadMovementThreshold_rpms	float	0.0 [0.0 .. 0.01]	
kLiftCoolDown_ms	unsigned	65 [0 .. 250]	
kLiftMovementThreshold_rpms	float	0.0 [0.0 .. 0.01]	
kMaxHeadAccel_rpms2	float	0.0001 [0.0 .. 0.001]	
kMaxHeadSpeed_rpms	float	0.005 [0.0 .. 0.025]	
kMaxLiftAccel_rpms2	float	0.0001 [0.0 .. 0.001]	
kMaxLiftSpeed_rpms	float	0.0025 [0.0 .. 0.05]	
kMaxTreadAccel_mmpms2	float	5.0 [0.0 .. 10.0]	
kMaxTreadSpeed_mmps	float	220.0	The maximum allowed speed (in mm/sec) a wheel can spin.
kMaxTurnSpeed_mmps	float	220.0	The maximum allowed speed (in mm/sec) a wheel can spin.
kTreadCoolDown_ms	unsigned	65 [0 .. 250]	
kTreadMovementThreshold_mmps	float	0.0 [0.0 .. 0.01]	

SayNameProbability console variables

Variable	Type	Default [Min .. Max]	Description
kSayNameMinProb	float	0.1 [0.0 .. 1.0]	
kSayNameProbDecayFactor	float	0.75 [0.0 .. 1.0]	
kSayNameSpacing_sec	float	10.0	

TextToSpeech console variables

Variable	Type	Default [Min .. Max]	Description
kEnablePausePrams	bool	TTS_ENABLEPAUSEPARAMS	
kLeadingSilence_ms	unsigned	TTS_LEADINGSILENCE_MS [0 .. 5000]	
kMinPlayableFrames	unsigned	8192 [0 .. 65536]	
kPauseBracket_ms	unsigned	TTS_PAUSEBRACKET_MS [50 .. 4000]	
kPauseComma_ms	unsigned	TTS_PAUSECOMMA_MS [50 .. 4000]	
kPausePunctuation_ms	unsigned	TTS_PAUSEPUNCTUATION_MS [50 .. 4000]	
kPauseSemicolon_ms	unsigned	TTS_PAUSESEMICOLON_MS [50 .. 4000]	
kPauseSpelling_ms	unsigned	TTS_PAUSESPELLING_MS [50 .. 4000]	
kTrailingSilence_ms	unsigned	TTS_TRAILINGSILENCE_MS [0 .. 5000]	
kVoicePitch	int	100 [70 .. 160]	
kVoiceShaping	int	100 [70 .. 140]	
kVoiceSpeed	int	100 [30 .. 300]	
kWriteTTSFile	bool	false	

kTextToSpeechPath console variables

Variable	Type	Default [Min .. Max]	Description
kDurationScalar	float	1.0 [0.25 .. 4.0]	
kPitchScalar	float	0.0 [-1.0 .. 1.0]	
kVoiceStyle	unsigned	0	

UserIntentComponent console variables

Variable	Type	Default	Description
kPlayGetInAfterDevWakeWord	bool	false	
kStreamAfterDevWakeWord	bool	false	

13.7.6 Behaviors

DevBaseBehavior console variables

Variable	Type	Default [Min .. Max]	Description
kDevDispatchAfterShake	unsigned int	0	
kShakeTime	float	0.1 [0.01 .. 2.0]	

BehaviorCountingAnimation console variables

Variable	Type	Default	Description
kSlowLoopBeginSize_loops	int	-1	
kSlowLoopEndSize_loops	int	-1	

BehaviorDanceToTheBeatCoordinator console variables

Variable	Type	Default [Min .. Max]	Description
kDancingCooldown_sec	float	20.0 [0.0 .. 3600.0]	
kListeningCooldown_sec	float	20.0 [0.0 .. 3600.0]	
kMinIntraDancingPeriod_sec	float	10.0 [0.0 .. 3600.0]	

Behavior.PuzzleMaze console variables

Variable	Type	Default [Min .. Max]	Description
kPuzzleTimeout_sec	float	24000.0 [0.0 .. 24000.0]	

BehaviorHighLevelAI console variables

Variable	Type	Default [Min .. Max]	Description
kHLAI_MinObservingBeforeExploring_s	float	10.0	
kTimeMultiplier	float	1.0 [1.0 .. 300.0]	

BehaviorPlannerTest console variables

Variable	Type	Default [Min .. Max]	Description
kCubeDistance_mm	float	25.0 [0.0 .. 100.0]	
kDistance_mm	float	1000.0 [0.0 .. 2000.0]	
kOnlyUseOriginalGoal	bool	true	

BehaviorReactToMotion console variables

Variable	Type	Default	Description
kTurnFirst	bool	true	

Behaviors.ActivationState console variables

Variable	Type	Default	Description
kDebugActivationState	bool	false	

Behaviors.BehaviorSystemManager console variables

Variable	Type	Default	Description
kDebugBehaviorStack	bool	false	

Behaviors.CheckForAndReactToSalientPoint console variables

Variable	Type	Default	Description
kCFARTSP_CooldownOverride_sec	float	0.0	

Behaviors.ConditionFactory console variables

Variable	Type	Default	Description
kDebugConditionFactory	bool	false	

Behaviors.InternalStatesBehavior console variables

Variable	Type	Default	Description
kDebugInternalStatesBehavior	bool	false	

Behaviors.RobustChargerObservation console variables

Variable	Type	Default	Description
kFakeLowlightCondition	bool	false	
kRobustChargerObservation_SaveImages	bool	false	

Behaviors.TakeAPhoto console variables

Variable	Type	Default	Description
kHeadAngleDeg	float	25	
kReadyToTakePhotoTimeout_sec	float	3.0	
kTakingPhotoTimeout_sec	float	6.0	

Habitat console variables

Variable	Type	Default	Description
kDevForceBeginConfirmHabitat	bool	false	

Photography console variables

Variable	Type	Default	Description
kDevIsStorageFull	bool	false	
kTakePhoto_UseRawPhotos	bool	false	

StimFace console variables

Variable	Type	Default [Min .. Max]	Description
kStimFace_ema_N	int	20 [0 .. 100]	
kStimFace_enabled	bool	false	
kStimFace_minSaturation	float	0.25 [0.0 .. 1.0]	
kStimFace_sendThresh	float	0.01 [0.0 .. 1.0]	

TimerUtility.AdvanceAnticSeconds console variables

Variable	Type	Default	Description
kAdvanceAnticSeconds	unsigned	10	

TimerUtility.AdvanceTimerAndAnticSeconds console variables

Variable	Type	Default	Description
kAdvanceTimerAndAnticSeconds	unsigned	60	

TimerUtility.AdvanceTimerSeconds console variables

Variable	Type	Default	Description
kAdvanceTimerSeconds	unsigned	60	

UserDefinedBehaviorTree console variables

Variable	Type	Default	Description
kEnableUserDefinedBehaviorTree	bool	false	

VoiceMessage console variables

Variable	Type	Default	Description
kRequireKnownUser	bool	true	

13.7.7 Interacting with people and faces

AcknowledgementBehaviors console variables

Variable	Type	Default [Min .. Max]	Description
kDistanceToConsiderClose_gap_mm	float	100.0 [0.0 .. 1000.0]	
kDistanceToConsiderClose_mm	float	300.0 [0.0 .. 1000.0]	
kFaceReactCooldown_s	float	4.0 [0.0 .. 60.0]	

BasicActions.TurnInPlace console variables

Variable	Type	Default [Min .. Max]	Description
kMaxUnexpectedMoveCountHeldInPalm	unsigned	11 [1 .. 200]	

BasicActions.TurnTowardsFace console variables

Variable	Type	Default	Description
kMaxTimeToWaitForRecognition_sec	float	3.0	

BasicActions.TurnTowardsObject console variables

Variable	Type	Default	Description
kInsertWaitsInTurnTowardsObjectVerify	bool	false	

BasicActions.WaitForImages console variables

Variable	Type	Default	Description
kDefaultNumFramesToWait	unsigned	3	

BehaviorBigGreeting console variables

Variable	Type	Default	Description
kBigGreetingDriveOffCharger	bool	true	

Behavior.EnrollFace console variables

Variable	Type	Default [Min .. Max]	Description
kEnrollFace_DefaultMaxFacesVisible	int	1	If Vector sees more than this number of faces, it is "too many" and will not enroll the face.
kEnrollFace_DefaultTooManyFacesRecentTime_sec	float	0.5	
kEnrollFace_DefaultTooManyFacesTimeout_sec	float	2.0	
kEnrollFace_DriveForwardIntentDist_mm	float	14.0	Amount to drive forward once face is found to signify intent
kEnrollFace_DriveForwardIntentSpeed_mmps	float	75.0	
kEnrollFace_FailOnWrongFace	bool	true	
kEnrollFace_MaxBackup_mm	float	15.0	Max distance to backup while looking for a face, up to max total amount
kEnrollFace_MaxInterruptionBeforeReset_ms	unsigned	10000	
kEnrollFace_MaxTotalBackup_mm	float	50.0	Max total distance to backup while looking for a face, up to max total amount
kEnrollFace_MaxTurnTowardsFaceAngle_rad	float	3.141592653	Max angle to turn while looking for a face
kEnrollFace_MinBackup_mm	float	5.0	Min distance to backup while looking for a face, up to max total amount
kEnrollFace_MinTrackingPanAngle_deg	float	4.0	Minimum angles to turn during tracking to keep the robot moving and looking alive
kEnrollFace_MinTrackingTiltAngle_deg	float	4.0	
kEnrollFace_NumImagesToWait	int	5	
kEnrollFace_NumImagesToWaitInPlace	int	25	
kEnrollFace_SayWrongNameMode	SayWrongNameMode	Long	This only matters if kEnrollFace_FailOnWrongFace==false
kEnrollFace_ScoreThresholdToFailOnWrongFace	int	800	If the max score for any observation of a "wrong face" is above this threshold, we will fail enrollment. If, however, it is below this threshold, we will go ahead and enroll this named face as a new person with the new name. Set to 0 to always fail when wrong face is seen.
kEnrollFace_TicksForKnownNameBeforeFail	unsigned	15	Number of times to see a named "wrong face" before either failing or going ahead and enrolling it.
kEnrollFace_Timeout_sec	float	25.0	
kEnrollFace_TimeoutForReLookForFace_ms	TimeStamp_t	3000	
kEnrollFace_TimeoutMax_sec	float	45.0	
kEnrollFace_UpdateFaceAngleThreshold_deg	float	45.0	

Variable	Type	Default [Min .. Max]	Description
kEnrollFace_UpdateFacePositionThreshold_mm	float	100.0	Thresholds for when to update face ID based on pose

ENUM SAYWRONGNAMEMODE

Name	Value	Description
Off	0	Don't say name at all, just go back to looking for faces
Short	1	Just say the name
Long	2	You are "X" not "Y"

Behavior.InteractWithFaces console variables

Variable	Type	Default [Min .. Max]	Description
kInteractWithFaces_DoGlanceDown	bool	false	If true, do a glance down before the memory map check (only valid if we are doing the check)
kInteractWithFaces_DoMemoryMapCheckForDriveForward	bool	true	If false, always drive the "ideal" distance without checking anything. If true, check memory map to determine which distance to drive
kInteractWithFaces_DriveForwardIdealDist_mm	float	40.0 [0.0 .. 200.0]	How far forward to check and ideally drive.
kInteractWithFaces_DriveForwardMinDist_mm	float	-15.0 [-100.0 .. 100.0]	How far forward to move in case the check fails.
kInteractWithFaces_DriveForwardSpeed_mmps	float	40.0 [0.0 .. 200.0]	
kInteractWithFaces_MinTrackingPanAngle_deg	float	4.0 [0.0 .. 30.0]	Minimum angle to turn during tracking to keep the robot moving and looking alive.
kInteractWithFaces_MinTrackingTiltAngle_deg	float	4.0 [0.0 .. 30.0]	Minimum angle to turn during tracking to keep the robot moving and looking alive.
kInteractWithFaces_VizMemoryMapCheck	bool	false	
kWiggle_BackupDist_mm	float	15.0 [0.0 .. 20.0]	
kWiggle_BackupSettleTime_s	float	0.4 [0.0 .. 2.0]	
kWiggle_BackupSpeed_mmps	float	100.0 [0.0 .. 200.0]	
kWiggle_ForwardDist_mm	float	6.0 [0.0 .. 20.0]	
kWiggle_ForwardSpeed_mmps	float	120 [0.0 .. 200.0]	
kWiggle_VerifyWaitTime_s	float	0.25 [0.0 .. 2.0]	

Behavior.LookAroundInPlace console variables

Variable	Type	Default	Description
kVizConeOfFocus	bool	false	

Behavior.ReactToHand console variables

Variable	Type	Default [Min .. Max]	Description
kHandReaction_DriveForwardSpeed_mmps	float	100.0 [0.0 .. MAX_SAFE_WHEEL_SPEED_MMPS]	
kReactToHand_DriveDistanceFraction	float	1.0 [0.0 .. 1.0]	
kReactToHand_PitchAngleThresh_deg	float	2.0 [0.0 .. 10.0]	

Behavior.ReactToPalmEdge console variables

Variable	Type	Default	Description
kMaxNumInitialReactAttemptsBeforeGivingUp	unsigned	2	If the behavior encounters this many failures with the initial animation/action while activated, then just give up and go to ForceStuckOnPalmEdge.

Behaviors.ConditionEyeContact console variables

Variable	Type	Default	Description
kMaxTimeSinceTrackedFaceUpdated_ms	unsigned	500	

Behaviors.FindFaceAndThen console variables

Variable	Type	Default [Min .. Max]	Description
kMinTimeLookInMicDirection_s	float	0.5 [0.0 .. 2.0]	

Vision.EyeContact console variables

Variable	Type	Default	Description
kBlinkAmountThreshold	float	.73	
kDistanceFromCameraThresholdSq_mm	float	2590	
kExpireThreshold	unsigned	50	
kEyeContactDistanceSq	float	64.0	
kHistorySize	unsigned	6	
kInlierDistanceSq	float	100.0	
kMinNumberOfInliers	unsigned	3	
kPitchAngleThreshold_rad	float	1.5707963265	
kYawAngleThreshold_rad	float	1.5707963265	

Vision.FaceDetection console variables

These give loose constraints on how fast Vector can move and still trust the tracker (which has no knowledge of or access to camera movement). Rough means of deciding the angles below:

1. look at angle created by distance between two faces seen close together at the max distance we care about seeing them from.

2. If robot turns by that angle between two consecutve frames, it is possible the tracker will be confused and jump from one to the other.

Variable	Type	Default [Min .. Max]	Description
kFaceTrackingCropWidthFraction	float	0.66667 [0.0 .. 1.0]	The percentage of the width of the image that will remain after cropping.
kFaceTrackingMaxBodyAngleChange_deg	float	8.0	
kFaceTrackingMaxHeadAngleChange_deg	float	8.0	
kFaceTrackingMaxPoseChange_mm	float	10.0	

Vision.FaceDetectorCommon console variables

Variable	Type	Default [Min .. Max]	Description
kAdjustEyeDistByYaw	bool	true	
kDetectionMode	DetectionMode	Movie	
kFaceDetectionThreshold	int	500 [1 .. 1000]	
kKeepUndistortedFaceFeatures	bool	false	
kMaxDetectedFaces	int	10 [1 .. 1023]	
kMaxFaceSize	int	640 [20 .. 8192]	
kMinFaceSize	int	48 [20 .. 8192]	
kPoseAngle	PoseAngle	Front	
kReinitDetector	bool	false	Use this to trigger a reinitialization on next Update()
kRollAngle	RollAngle	UpperPm45	
kSearchDensity	SearchDensity	Normal	
kUseUndistortionForFacePose	bool	true	

Vision.FaceDetectorMovie console variables

Variable	Type	Default [Min .. Max]	Description
kDelayCount	int	1 [0 .. 10]	
kDirectionMask	bool	false	
kEnableAngleExtension	bool	false	
kEnablePoseExtension	bool	true	
kLostMaxHold	int	2 [0 .. 300]	
kLostMaxRetry	int	2 [0 .. 300]	
kSearchInitialCycle	int	2 [1 .. 45]	
kSearchNewCycle	int	2 [1 .. 45]	
kSearchNewInterval	int	5 [-1 .. 45]	
kSteadinessPosition	int	10 [0 .. 30]	
kSteadinessSize	int	10 [0 .. 30]	
kTrackingAccuracy	int	Okao::TrackingAccuracy	High
kTrackingSwapRatio	int	400 [100 .. 10000]	
kUseHeadTracking	bool	false	When setting this to true, we were seeing worse part detection performance while tracking. The nPose field in the DetectionInfo struct was sometimes "HEAD" (meaning back of head). From the Omron team: "It returned "Head" because you set bUseHeadTracking as TRUE of OKAO_DT_MV_SetPoseExtension(). (It's default value is FALSE.) Face Detection engine output "Head" only by tracking, not from the first frame or Still Mode. It is good for keeping tracking, but not good for Facial Parts Detection. If you give priority to Facial Parts Detection over tracking, you should turn bUseHeadTracking off or skip the face." So I'm defaulting this to false, and it seems to help in testing.

Vision.FaceRecognition console variables

Variable	Type	Default [Min .. Max]	Description
kDisplayDebugEnrollmentImages	bool	false	
kEnableEnrollmentAfterFull	bool	false	
kEnableMergingOfSessionOnlyAlbumEntries	bool	false	
kEnrollmentThumbnailSize	int	64	
kFaceRecMaxDebugResults	unsigned	3 [2 .. 10]	
kFaceRecognitionExtraDebug	bool	false	
kFaceRecognitionGuessThreshold	int	350 [0 .. 1000]	
kFaceRecognitionSimulatedDelay_ms	unsigned	0	
kFaceRecognitionThreshold	int	575 [0 .. 1000]	
kFaceRecognitionThresholdMarginForAdding	int	200 [0 .. 1000]	
kFaceRecognitionThresholdMarginForUsing2ndBest	int	50 [0 .. 1000]	
kGatherDebugEnrollmentImages	bool	false	
kGetEnrollmentTimeFromImageTimestamp	bool	false	
kTimeBetweenFaceEnrollmentUpdates_sec	float	0.5	

Vision.FaceTracker console variables

Variable	Type	Default	Description
kCloseDistanceBetweenEyesMax	float	128.0	
kCloseDistanceBetweenEyesMin	float	64.0	
kFaceDetectionDelay_ms	int	100	
kFaceDetectionDelayDuringEnrollment_ms	int	1000	
kFaceRecognitionDelay_ms	int	0	
kFarDistanceBetweenEyesMax	float	32.0	
kFarDistanceBetweenEyesMin	float	16.0	
kFramesToCompleteEnrollment	int	50	
kFramesToLoseFaceAfterEnrollment	int	2000	
kLookingDownMaxAngle_deg	float	-25.0	
kLookingDownMinAngle_deg	float	-10.0	
kLookingLeftRightMaxAngle_deg	float	20.0	
kLookingLeftRightMinAngle_deg	float	10.0	
kLookingStraightMaxAngle_deg	float	25.0	
kLookingUpMaxAngle_deg	float	45.0	
kLookingUpMinAngle_deg	float	25.0	
kMinDetectionConfidence	int	500	Faces are not enrollable unless the tracker is above this confidence. NOTE: It appears the returned track confidence is set to the fixed value of whatever the OKAO detection threshold is set to when in default tracking accuracy mode, so this parameter will have no effect unless the high-accuracy tracker is used.
kNumberOfFramesBeforeUpdatedFace	int	200	

Vision.FaceWorld console variables

Variable	Type	Default	Description
kDeletionTimeout_ms	unsigned	600000	
kHeadCenterPointThreshold_mm	float	220.0	
kIgnoreFacesBelowRobot	bool	true	
kNumTimesToSeeFrontalToBeStable	unsigned	30	
kTimeUnobservedBeforeReLoggingToDAS_ms	unsigned	10000	

Vision.GazeDirection console variables

Variable	Type	Default	Description
kConeFor180TurnForFaceSearch_deg	float	40.0	
kFaceDirectedAtRobotMaxXThres_mm	float	20.0	
kFaceDirectedAtRobotMaxYThres_mm	float	100.0	
kFaceDirectedAtRobotMinXThres_mm	float	-25.0	
kFaceDirectedAtRobotMinYThres_mm	float	-100.0	
kGazeDirectionExpireThreshold_ms	unsigned	1000	
kGazeDirectionHistorySize	unsigned	6	
kGazeDirectionInlierXThreshold_mm	float	300.0	
kGazeDirectionInlierYThreshold_mm	float	100.0	
kGazeDirectionInlierZThreshold_mm	float	20.0	
kGazeDirectionMinNumberOfInliers	unsigned	2	
kGazeDirectionSecondPointTranslationY_mm	float	1500.0	"This value was chosen to be sufficiently large that the difference in the z coordinates of the two points used to find the intersection with the ground plane weren't too close as to cause numerical instabilities. 500 was too small."
kGazeDirectionShiftOutputPointX_mm	float	100.0	
kMaxPanAccel_radPerSec2	float	10.0	
kMaxPanSpeed_radPerSec	float	MAX_BODY_ROTATION_SPEED_RAD_PER_SEC	
kMaxTimeSinceTrackedFaceUpdated_ms	unsigned	500	
kNumberOfTurnsForSurfacePoint	int	1	
kRenderGazeDirectionPoints	bool	false	
kSearchForFaceNumberOfImagesToWait	int	5	
kSearchForFaceTurnAroundAngle_deg	float	180.0	
kSearchForFaceTurnSideAngle_deg	float	-90.0	
kSleepTimeAfterActionCompleted_s	float	2.0	

Variable	Type	Default		Description
kTurnWaitAfterFinalTurn_s	float	1.0		
kUseExistingFacesWhenSearchingForFaces	bool	false		
kUseEyeContact	bool	true		

"WasRotatingTooFast.Dock.Body_deg/s" console variables

Variable	Type	Default		Description
kDockingRotatingTooFastThresh_degPerSec	float	RAD_TO_DEG	0.4	
kBodyTurnSpeedThreshFace_degs	float	30.0		

"WasRotatingTooFast.Face.Head_deg/s" console variables

Variable	Type	Default	Description
kHeadTurnSpeedThreshFace_degs	float	10.0	

"WasRotatingTooFast.Face.NumToLookBack" console variables

Variable	Type	Default	Description
kNumImuDataToLookBackFace	unsigned	5	

"WasRotatingTooFast.Pet.Body_deg/s" console variables

Variable	Type	Default	Description
kBodyTurnSpeedThreshPet_degs	float	30.0	

"WasRotatingTooFast.Pet.Head_deg/s" console variables

Variable	Type	Default	Description
kHeadTurnSpeedThreshPet_degs	float	10.0	

"WasRotatingTooFast.Pet.NumToLookBack" console variables

Variable	Type	Default	Description
kNumImuDataToLookBackPet	unsigned	5	

13.7.8 Cube interaction

Behavior.PutDownBlock console variables

Variable	Type	Default	Description
kBPDB_finalHeadAngle_deg	float	-20.0	
kBPDB_kBackupDistanceMax_mm	float	-75	
kBPDB_kBackupDistanceMin_mm	float	-45	
kBPDB_putDownBackupSpeed_mm	float	100.0	
kBPDB_verifyBackupDist_mm	float	-30.0	

CubeAccelComponent console variables

Variable	Type	Default	Description
kCanAccelDirtyPoses	bool	false	

CubeLightDesign console variables

Variable	Type	Default [Min .. Max]	Description
kLED1_s1_alpha	unsigned	255 [0 .. 255]	
kLED1_s1_blue	unsigned	0 [0 .. 255]	
kLED1_s1_green	unsigned	0 [0 .. 255]	
kLED1_s1_hold_ms	unsigned	0 [0 .. 7650]	
kLED1_s1_hold_offset_ms	unsigned	0 [0 .. 30600]	
kLED1_s1_red	unsigned	0 [0 .. 255]	
kLED1_s1_transition_s2_ms	unsigned	0 [0 .. 7650]	
kLED1_s2_alpha	unsigned	255 [0 .. 255]	
kLED1_s2_blue	unsigned	0 [0 .. 255]	
kLED1_s2_green	unsigned	0 [0 .. 255]	
kLED1_s2_hold_ms	unsigned	0 [0 .. 7650]	
kLED1_s2_red	unsigned	0 [0 .. 255]	
kLED1_s2_transition_s1_ms	unsigned	0 [0 .. 7650]	
kLED2_s1_alpha	unsigned	255 [0 .. 255]	
kLED2_s1_blue	unsigned	0 [0 .. 255]	
kLED2_s1_green	unsigned	0 [0 .. 255]	
kLED2_s1_hold_ms	unsigned	0 [0 .. 7650]	
kLED2_s1_hold_offset_ms	unsigned	0 [0 .. 30600]	
kLED2_s1_red	unsigned	0 [0 .. 255]	
kLED2_s1_transition_s2_ms	unsigned	0 [0 .. 7650]	
kLED2_s2_alpha	unsigned	255 [0 .. 255]	
kLED2_s2_blue	unsigned	0 [0 .. 255]	
kLED2_s2_green	unsigned	0 [0 .. 255]	
kLED2_s2_hold_ms	unsigned	0 [0 .. 7650]	
kLED2_s2_red	unsigned	0 [0 .. 255]	
kLED2_s2_transition_s1_ms	unsigned	0 [0 .. 7650]	
kLED3_s1_alpha	unsigned	255 [0 .. 255]	
kLED3_s1_blue	unsigned	0 [0 .. 255]	
kLED3_s1_green	unsigned	0 [0 .. 255]	
kLED3_s1_hold_ms	unsigned	0 [0 .. 7650]	
kLED3_s1_hold_offset_ms	unsigned	0 [0 .. 30600]	
kLED3_s1_red	unsigned	0 [0 .. 255]	
kLED3_s1_transition_s2_ms	unsigned	0 [0 .. 7650]	
kLED3_s2_alpha	unsigned	255 [0 .. 255]	
kLED3_s2_blue	unsigned	0 [0 .. 255]	

Variable	Type	Default [Min .. Max]	Description
kLED3_s2_green	unsigned	0 [0 .. 255]	
kLED3_s2_hold_ms	unsigned	0 [0 .. 7650]	
kLED3_s2_red	unsigned	0 [0 .. 255]	
kLED3_s2_transition_s1_ms	unsigned	0 [0 .. 7650]	
kLED4_s1_alpha	unsigned	255 [0 .. 255]	
kLED4_s1_blue	unsigned	0 [0 .. 255]	
kLED4_s1_green	unsigned	0 [0 .. 255]	
kLED4_s1_hold_ms	unsigned	0 [0 .. 7650]	
kLED4_s1_hold_offset_ms	unsigned	0 [0 .. 30600]	
kLED4_s1_red	unsigned	0 [0 .. 255]	
kLED4_s1_transition_s2_ms	unsigned	0 [0 .. 7650]	
kLED4_s2_alpha	unsigned	255 [0 .. 255]	
kLED4_s2_blue	unsigned	0 [0 .. 255]	
kLED4_s2_green	unsigned	0 [0 .. 255]	
kLED4_s2_hold_ms	unsigned	0 [0 .. 7650]	
kLED4_s2_red	unsigned	0 [0 .. 255]	
kLED4_s2_transition_s1_ms	unsigned	0 [0 .. 7650]	
kRotate	bool	false	

CubeSpinner console variables

Variable	Type	Default	Description
kAdjustHeightOfSpinnerLift	int	81	
kDedupTimeAfterLock_ms	int	1000	
kIReallyReallyWantToBreakCubeSpinner	bool	false	
kShouldLockPulseTargetColor	bool	true	

DoubleTap console variables

Variable	Type	Default	Description
kCanDoubleTapDirtyPoses	bool	false	
kIgnoreMovementWhileWaitingForDoubleTap	bool	false	

TapFilter.DoubleTapTime console variables

Variable	Type	Default	Description
kDoubleTapTime_ms	unsigned	500	

TapFilter.IgnoreMoveTimeAfterDoubleTap console variables

Variable	Type	Default	Description
kIgnoreMoveTimeAfterDoubleTap_ms	unsigned	500	

TapFilter.IntesityMin console variables

Variable	Type	Default	Description
kTapIntensityMin	int16_t	60	

TapFilter.WaitOffsetTime console variables

Variable	Type	Default	Description
kTapWaitOffset_ms	Anki::TimeStamp_t	75	

13.7.9 Emotion and Mood

Mood.Emotion console variables

Variable	Type	Default	Description
kMaxEmotionHistorySamples	unsigned	128	

MoodManager console variables

Variable	Type	Default	Description
kMoodManager_AppPeriod_s	float	1.0	
kMoodManager_AudioSendPeriod_s	float	0.5	
kMoodManager_WebVizPeriod_s	float	1.0	

13.7.10 Exploring

Behavior.BehaviorGoHome console variables

Variable	Type	Default	Description
kGoHome_VisualVerification_SaveImages	bool	false	

Behavior.ReactToCliff console variables

Variable	Type	Default	Description
kEnableVisualCliffExtension	bool	true	
kMaxNumCliffReactAttemptsBeforeGivingUp	unsigned	2	
kMaxNumRobotStopsBeforeGivingUp	unsigned	5	
kMinViewingDistanceToCliff_mm	float	80.0	

BehaviorExploring console variables

Variable	Type	Default [Min .. Max]	Description
kExploringPostBumpReferenceProb	float	1.0 [0.0 .. 1.0]	
kMaxObjectWidthToBump_rad	float	1.39626 [0 .. 6.2831853]	
kMinObjectWidthToBump_rad	float	0.01745 [0 .. 3.141592653]	For bumping an object. The robot is usually around 5-8cm from the object at this point, but may not be facing it perfectly, so only bump if the object seems to have an appropriate width. The delegated behavior decides if it is close enough.
kMoveLiftAboveProx	bool	false	
kProbReferenceBeforeBump	float	0.0 [0.0 .. 1.0]	This is disabled because it looks much nicer when he bumps right after the scan, so instead it is set to reference *after* the bump from within the bump behavior.
kProbReferenceOnResume	float	1.0 [0.0 .. 1.0]	
kResumeReferenceCooldown_s	float	20.0 [0.0 .. 60.0]	

13.7.11 Motion control

Robot console variables

Variable	Type	Default	Description
kCreateUnexpectedMovementObstacles	bool	true	
kDebugTrackLocking	bool	false	
kMaxUnexpectedMovementCountWhileHeldInPalm	unsigned	200	

DockingMethod(B:0 T:1 H:2) console variables

Variable	Type	Default	Description
kDefaultDockingMethod	DockingMethod	BLIND_DOCKING	
kPickupDockingMethod	DockingMethod	HYBRID_DOCKING_BEELINE	
kRollDockingMethod	DockingMethod	BLIND_DOCKING	
kStackDockingMethod	DockingMethod	BLIND_DOCKING	

DriveToActions console variables

Variable	Type	Default	Description
kDriveToPoseTimeout	float	30.0	
kEnablePredockDistanceCheckFix	bool	true	
kEnableDrivingAnimations	bool	true	

PlaceRelObjectAction console variables

Variable	Type	Default	Description
kPlaceRelUseMaxOffset	bool	true	

PoseConfirmation console variables

Variable	Type	Default [Min .. Max]	Description
kDefaultMaxObservationDistance_mm	float	500.0 [50.0 .. 1000.0]	

13.7.12 Navigation

MapComponent console variables

Variable	Type	Default	Description
kCliffTimeout_ms	float	1200000	20 minutes
kMapRenderRate_sec	float	0.25	
kMergeOldMaps	bool	false	
kObjectPositionChangeToReport_mm	float	5.0	
kObjectRotationChangeToReport_deg	float	10.0	
kProxExploredTriangleHalfWidth_mm	float	50.0	
kProxExploredTriangleLength_mm	float	300.0	
kProxTimeout_ms	float	600000	
kRobotPositionChangeToReport_mm	float	8.0	
kRobotRotationChangeToReport_deg	float	20.0	
kTimeoutUpdatePeriod_ms	float	5000	
kUnrecognizedTimeout_ms	float	20000	
kVisionTimeout_ms	float	120000	

MapComponent.VisualEdgeDetection console variables

Variable	Type	Default	Description
kEdgeLineLengthToInsert_mm	float	200.0	
kHoughAccumThreshold	int	20	
kHoughAngleResolution_deg	float	2	
kHoughMaxLineGap_mm	float	10	
kHoughMinLineLength_mm	float	40	
kMaxPixelsUsedForHoughTransform	int	160000	400 x 400 max size
mapComponent	kVisionCliffPadding_mm	float	20.0

Planner console variables

Variable	Type	Default	Description
kXYTPlanner_PointTurnTolOverride_deg	float	2.0	

XYPlanner console variables

Variable	Type	Default [Min .. Max]	Description
kArtificialPlanningDelay_ms	int	0 [0 .. 3900]	

QuadTreeProcessor console variables

Variable	Type	Default	Description
kDebugFindBorders	bool	false	Enables printing debug information in console
kRenderBorder3DLines	bool	false	Enables rendering borders returned as 3D lines instead of quads
kRenderBordersFrom	bool	false	Enables rendering detected borders, origin quad
kRenderBordersToDot	bool	false	Enables rendering detected borders, border center as dots
kRenderBordersToQuad	bool	false	Enables rendering detected borders, destination quad
kRenderSeeds	bool	false	Renders seeds differently for debugging purposes
kRenderZOffset	float	20.0	Adds Z offset to all quads

13.7.13 Power management

Charger console variables

Variable	Type	Default	Description
kChargerMaxObservationDistance_mm	float	500.0	

BatteryComponent console variables

Variable	Type	Default [Min .. Max]	Description
kFakeDisconnectedBattery	bool	false	
kFakeLowBattery	bool	false	
kFakeLowBatteryAfterOffChargerTimeout_sec	unsigned	0	
kPeriodicDebugDASLogging	bool	false	
kRequiredChargeTime_s	float	300.0 [10.0 .. 9999.0]	Must be set before low battery and then not changed

PowerSave console variables

Variable	Type	Default	Description
kForceCalmMode	bool	false	
kPowerSave_CalmMode	bool	true	
kPowerSave_Camera	bool	true	
kPowerSave_CameraStopCameraStream	bool	false	
kPowerSave_LCDBacklight	bool	true	
kPowerSave_ProxSensorMap	bool	true	
kPowerSave_ThrottleCPU	bool	true	

Sleeping.Behavior console variables

Variable	Type	Default [Min .. Max]	Description
kSleepingBoutNumStirs_max	unsigned	10 [1 .. 10]	
kSleepingBoutNumStirs_min	unsigned	5 [1 .. 10]	
kSleepingBoutSpacing_max_s	float	5.0 [0.0 .. 7200.0]	
kSleepingBoutSpacing_min_s	float	1.5 [0.0 .. 30.0]	
kSleepingStirSpacing_max_s	float	5400.0 [0.0 .. 7200.0]	
kSleepingStirSpacing_min_s	float	2700.0 [0.0 .. 7200.0]	

Sleeping.SleepCycle console variables

Variable	Type	Default	Description
kSleepCycle_ComatoseLength_s	float	30.0	
kSleepCycle_DeepSleep_PersonCheckInterval_s	float	14400.0	
kSleepCycle_EnableWiggleWhileSleeping	bool	true	
kSleepCycle_LightSleep_PersonCheckInterval_s	float	3600.0	
kSleepCycle_MinSleepDebt_s	float	3000.0	
kSleepCycle_RecentSleepLength_s	float	600.0	
kSleepCycle_TooLongOnChargerNotChargingDuration_sec	float	300.0	
kSleepCycleForceLightSleep	bool	false	
kSleepCycleForceSleep	bool	false	

Sleeping.SleepTracker console variables

Variable	Type	Default [Min .. Max]	Description
kSleepTracker_awakeSleepDebtRate	float	5.0	
kSleepTracker_debtToConsiderSleepy_awake	float	3600.0	
kSleepTracker_debtToConsiderSleepy_fromSleep	float	1800.0	
kSleepTracker_maxSleepDebt_hours	float	20	
kSleepTracker_moning_hour	int	7 [0 .. 23]	The hour portion of the time of day to wake up.
kSleepTracker_moning_minute	int	0 [0 .. 59]	The minute portion of the time of day to wake up.
kSleepTracker_night_hour	int	21 [0 .. 23]	The hour portion of the time of day to return to the charger to sleep.
kSleepTracker_night_minute	int	0 [0 .. 59]	The minute portion of the time of day to return to the charger to sleep.
kSleepTracker_updatePeriod_s	float	60	

StayOnCargerUntilCharged console variables

Variable	Type	Default	Description
kCooldown_s	float	1200.0	
kMinTimeAtNominal_s	float	4.0	>= time for any drive-off-charger anim to clear charger platform
kSafeguardTimeout_s	float	1800	

13.7.14 Sensing

HeldInPalm.Coordinator console variables

Variable	Type	Default [Min .. Max]	Description
kMaxTimeForInitialHeldInPalmReaction_ms	unsigned	1000 [0 .. 5000]	
kCliffValHeldInPalmSurface	float	500.0 [0.0 .. 1000.0]	
kEnableDebugTransitionPrintouts	bool	false	
kMinTimeToConfirmRobotHeldInPalm_ms	unsigned	500 [0 .. 10000]	If no cliffs have been detected since the robot was picked up, but the robot has been reporting that it has been picked up and held upright for this amount of time, go ahead and declare the robot to be held in a palm anyways. This is essentially a fallback for the normal detection mechanism for the tracker.
kTrackerWebVizUpdatePeriod_s	float	60	

ProxSensorComponent console variables

Variable	Type	Default	Description
kMapPerformanceTestsEnabled	bool	false	
kMapPerformanceTestsSampleWindow	int	128	
kMaxObstacleWidth_mm	float	18.0	
kMaxObsThreshold_mm	unsigned	400	
kMinObsThreshold_mm	unsigned	30	
kMinQualityThreshold	float	0.01	
kObsPadding_x_mm	float	6.0	
kObsPadding_y_mm	float	0.0	
kRenderProxBeliefs	bool	false	
kSensorAperture	float	0.4	

13.7.15 Touch console variables

Variable	Type	Default	Description
kTestOnlyLoggingEnabled	bool	false	

13.7.16 Profiling (low level information)

Console console variables

Variable	Type	Default	Description
kSaveModifiedConsoleVarsOnly	bool	false	

CpuProfiler console variables

Variable	Type	Default [Min .. Max]	Description
kAnimEngine_TimeLogging	unsigned	0	
kAnimEngine_TimeMax_ms	float	33 [2 .. 33]	
kCozmoEngine_Logging	unsigned	0	
kDrawFace_Logging	unsigned	0	
kMessageProfilerDuration	float	0.0 [0.0 .. 3600.0]	
kMicDataProcessorRaw_Logging	unsigned	0	
kMicDataProcessorTrigger_Logging	unsigned	0	
kProfilerLogOutput	int	0	
kProfilerLogSlowTicks	bool	false	
kVisionComponent_Logging	unsigned	0	
maxDrawTime_ms	float	5 [5 .. 32]	
maxProcessingTimePerDrop_ms	float	5 [5 .. 32]	
maxTriggerProcTime_ms	float	10 [10 .. 32]	

Dev console variables

Variable	Type	Default	Description
kForceDisableAnkiDevFeatures	bool	false	

DevLogging console variables

Variable	Type	Default	Min	Max	Description
kSaveImageFrequency	unsigned	0	0	75	

FeatureGate console variables

Variable	Type	Default	Description
kFeatureToEdit	unsigned	0	

Firmware console variables

Variable	Type	Default	Description
kAlwaysDoFirmwareUpdate	bool	false	
kSkipFirmwareAutoUpdate	bool	false	

JdocsManager console variables

Variable	Type	Default	Description
kJdocType	unsigned	0	

Logging console variables

Variable	Type	Default	Description
kEnableCladLogger	bool	true	

Network console variables

Variable	Type	Default [Min .. Max]	Description
kEnableVerboseNetworkLogging	bool	false	
kMaxPingTimesToTrackOverride	unsigned	0 [0 .. 1000]	
kPrintNetworkStats	bool	false	
kPrintNetworkStatsTimeSpacingMS	unsigned	1000 [0 .. 10000]	

Network.Emulator console variables

Variable	Type	Default [Min .. Max]	Description
gUDPMaxLatency	unsigned	0 [0 .. 5000]	
gUDPMinLatency	unsigned	0 [0 .. 5000]	
gUDPNetEmulatorEnabled	bool	false	
gUDPNetEmulatorRuntimeToggling	bool	false	
gUDPRandomPacketLossPercentage	float	-1.0 [-1.0 .. 101.0]	

Network.Stats console variables

Variable	Type	Default	Description
gNetStat1NumConnections	int		
gNetStat2LatencyAvg	float		
gNetStat3LatencySD	float		
gNetStat4LatencyMin	float		
gNetStat5LatencyMax	float		
gNetStat6PingArrivedPC	float		
gNetStat7ExtQueuedAvg_ms	float		
gNetStat8ExtQueuedMin_ms	float		
gNetStat9ExtQueuedMax_ms	float		
gNetStatAQueuedAvg_ms	float		
gNetStatBQueuedMin_ms	float		
gNetStatCQueuedMax_ms	float		
kLogMessageLatencyOnce	bool	false	
kNetConnStatsUpdate	bool	true	

OSState.Boot console variables

Variable	Type	Default	Description
kFakeIsReboot	bool	false	

OSState.DiskInfo console variables

Variable	Type	Default [Min .. Max]	Description
kHighDiskPressureMultiple	unsigned	10 [0 .. 100]	
kMediumDiskPressureMultiple	unsigned	5 [0 .. 100]	

OSState.MemoryInfo console variables

Variable	Type	Default [Min .. Max]	Description
kHighMemPressureMultiple	unsigned	10 [0 .. 100]	
kMediumMemPressureMultiple	unsigned	5 [0 .. 100]	

OSState.Temperature console variables

Variable	Type	Default	Description
kFakeCpuTemperature_degC	unsigned	20	
kSendFakeCpuTemperature	bool	false	

OSState.Timezone console variables

Variable	Type	Default	Description
kOSState_FakeNoTime	bool	false	
kOSState_FakeNoTimezone	bool	false	

OSState.Webviz console variables

Variable	Type	Default [Min .. Max]	Description
kWebvizUpdatePeriod	int	0 [0 ..]	The duration, in milliseconds, between successive updates to the web visualization

OSState.WifiInfo console variables

Variable	Type	Default [Min .. Max]	Description
kHighWifiErrorRate	unsigned	2 [0 .. 100]	
kMediumWifiErrorRate	unsigned	1 [0 .. 100]	

Robot console variables

Variable	Type	Default	Description
kDebugPossibleBlockInteraction	bool	false	
kEnableTestFaceImageRGBDrawing	bool	false	
kUseVisionOnlyWhileOnTreads	bool	false	

RobotDataLoader console variables

Variable	Type	Default	Description
kStressTest_numThreads	int	5	
kStressTestThreadedPrintsDuringLoad	bool	false	

RobotStats console variables

Variable	Type	Default	Description
kRobotStats_AliveUpdatePeriod_s	float	60.0	
kRobotStats_OverrideAliveHours	float	-1.0	

Settings.Debug console variables

Variable	Type	Default	Description
kHttpRequestTimeOutMSec	int	10000	0

"unit tests" console variables

Variable	Type	Default	Description
kTestBEIConsoleVar	unsigned	0	

UiComms console variables

Variable	Type	Default	Description
kAcceptMessagesFromUI	bool	true	
kPingSendFreq_ms	double	1000	0 = never
kPrintUiMessageLatency	bool	false	
kSdkStatusSendFreq	unsigned	1	0 = never

VizDebug console variables

Variable	Type	Default	Description
kSendAnythingToViz	bool	true	

WallTime console variables

Variable	Type	Default	Description
kFakeWallTimeIsSynced	bool	false	

13.7.17 Factory Test

DevSquawkBoxBehavior console variables

Variable	Type	Default [Min .. Max]	Description
kHeadMovementDuration_s	float	0.5 [0.1 .. 2.5]	
kLiftMovementDuration_s	float	0.5 [0.1 .. 2.5]	
kLoopingAnimationState	unsigned	0	NONE, Move Head, Move Lift
kTreadMovementSpeed_mmps	float	200.0 [20.0 .. 220.0]	
kUseRestrictedMotionAnim	bool	true	

DockingTest console variables

Variable	Type	Default	Description
kAlignInsteadOfPickup	bool	false	
kDoDeepRoll	bool	false	
kDriveToAndPickupBlockOneAction	bool	true	
kJustPickup	bool	false	
kMaxAngleAwayFromPreDock_deg	float	10	
kMaxConsecFails	unsigned	3	
kMaxNumAttempts	unsigned	30	
kMaxXAwayFromPreDock_mm	float	50	
kMaxYAwayFromPreDock_mm	float	250	
kNumRandomObstacles	unsigned	10	
kRollInsteadOfPickup	bool	false	
kTestDockingMethod	DockingMethod	HYBRID_DOCKING	
kUseClosePreActionPose	bool	false	

LiftLoadTest console variables

Variable	Type	Default	Description
kNumLiftRaises	unsigned	50	

Playpen console variables

Variable	Type	Default	Description
kCalibMarkerCubeSize_mm	float		
kCalibMarkerSize_mm	float		
kCenterTolerance	unsigned		
kCheckFirmwareVersion	bool		
kCliffSensorThreshold	unsigned		
kCliffSpeed_mmps	float		
kDefaultTimeout_ms	float		
kDisconnectAtEnd	bool		
kDistanceSensorBiasAdjustment_mm	float		
kDistanceSensorReadingThresh_mm	float		
kDistanceToDriveOverCliff_mm	float		
kDistanceToTriggerBackCliffs_mm	float		
kDistanceToTriggerFrontCliffs_mm	float		
kDurationOfAudioToRecord_ms	unsigned		
kDurationOfTouchToRecord_ms	unsigned		
kExpectedCubePoseAngleThresh_rad	float		
kExpectedCubePoseDistThresh_mm	float		
kExpectedCubePoseHeightThresh_mm	float		
kExpectedCubePoseX_mm	float		
kExpectedCubePoseY_mm	float		
kExposure_ms	unsigned		
kFFTExpectedFreq_hz	unsigned		
kFFTFreqTolerance_hz	unsigned		
kFocalLengthTolerance	unsigned		
kGain	float		
kHeadAngleForDriftCheck	float		
kHeadAngleToPlaySound	float		
kHeadAngleToSeeTarget_rad	float		
kIgnoreFailures	bool		
kIMUDriftAngleThreshDeg	float		
kIMUDriftDetectPeriod_ms	unsigned		
kMarkerToTriggerCalibration	CustomObjectMarker		
kMaxExpectedTouchValue	unsigned		
kMaxRobotAngleChangeDuringBackup_rad	float		
kMfgIDTimeout_ms	unsigned		

Variable	Type	Default	Description
kMinBatteryVoltage	float		
kMinExpectedTouchValue	unsigned		
kMinFirmwareVersion	unsigned		
kMinHardwareVersion	int		
kMotorCalibrationTimeout_ms	unsigned		
kNumDistanceSensorReadingsToRecord	unsigned		
kPlaypenCalibTarget	unsigned		
kRadialDistortionTolerance	float		
kSkipActiveObjectCheck	bool		
kSoundVolume	float		
kTangentialDistortionTolerance	float		
kTimeoutForCalibration_ms	unsigned		
kTimeoutWaitingForTarget_ms	unsigned		
kTimeToDisplayResultOnFace_ms	float		
kTimeToWaitForCliffEvent_ms	float		
kTouchDurationToStart_ms	unsigned		
kUseButtonToStart	bool		
kUseTouchToStart	bool		
kVisualDistanceToDistanceSensorObjectThresh_mm	float		
kWriteToStorage	bool		

SelfTest console variables

Variable	Type	Default	Description
kChargerMarkerLastObservedTimeThresh_ms	unsigned		
kDefaultTimeout_ms	float		
kDistanceSensorBiasAdjustment_mm	float		
kDistanceSensorReadingThresh_mm	float		
kDistanceToDriveForwards_mm	float		
kDriveBackwardsDist_mm	unsigned		
kDriveBackwardsSpeed_mmps	unsigned		
kDriveSpeed_mmps	float		
kDurationOfAudioToRecord_ms	unsigned		
kFFTExpectedFreq_hz	unsigned		
kFFTFreqTolerance_hz	unsigned		
kHeadAngleForDriftCheck	float		
kHeadAngleToPlaySound	float		
kIgnoreFailures	bool		
kIMUDriftAngleThreshDeg	float		
kIMUDriftDetectPeriod_ms	unsigned		
kMaxExpectedTouchValue	unsigned		
kMinBatteryVoltage	float		
kMinExpectedTouchValue	unsigned		
kMotorCalibrationTimeout_ms	unsigned		
kNumDistanceSensorReadingsToRecord	unsigned		
kSoundVolume	float		
kTimeToBeUpsideDown_ms	unsigned		
kTimeToDisplayResultOnFace_ms	float		
kVisualDistanceToDistanceSensorObjectThresh_mm	float		

Vision.Calibration console variables

Variable	Type	Default	Description
kCalibTargetType	CalibTargetType	CHECKERBOARD	
kCheckerboardHeight	unsigned	4	
kCheckerboardSquareSize_mm	float	0.05	
kCheckerboardWidth	unsigned	11	
kDrawCalibImages	bool	false	
kMaxCalibBlobPixelArea	float	800.0	
kMinCalibBlobPixelArea	float	20.0	
kMinCalibPixelDistBetweenBlobs	float	5.0	
kMinNumCalibImages	unsigned	1	
kNumMarkersNeededForCalibration	unsigned	10	
kSingleTargetReprojErr_pix	float	1.5	

13.7.18 RobotSettings console variables

Variable	Type	Default	Description
kButtonWakeWord	unsigned	0	kButtonWakeWords
kDebugDemoLocaleIndex	int	0	
kEyeColor	unsigned	0	kEyeColors
kMasterVolumeLevel	unsigned	0	kMasterVolumeLevels

13.7.19 Vision Processing

NeuralNets console variables

Variable	Type	Default [Min .. Max]	Description
kNeuralNets_MaxNumSceneDescriptionTags	int	5 [3 .. 10]	

TrackingActions console variables

Variable	Type	Default [Min .. Max]	Description
kOverride_ClampSmallAngles	bool	false	
kOverride_ClampSmallAnglesMaxPeriod_s	float	-1.0 [0.0 .. 5.0]	
kOverride_ClampSmallAnglesMinPeriod_s	float	-1.0 [0.0 .. 5.0]	
kOverride_PanDuration_s	float	-1.0 [0.0 .. 1.0]	
kOverride_PanTolerance_deg	float	-1.0 [0.0 .. 20.0]	
kOverride_TiltDuration_s	float	-1.0 [0.0 .. 1.0]	
kOverride_TiltTolerance_deg	float	-1.0 [0.0 .. 20.0]	

Vision.Benchmark console variables

Variable	Type	Default	Description
kVisionBenchmark_DisableAllModes	bool	false	
kVisionBenchmark_DisplayImages	bool	false	Only works if running synchronously
kVisionBenchmark_EnableAllModes	bool	false	
kVisionBenchmark_PrintFrequency_ms	int	3000	
kVisionBenchmark_ScaleMultiplier	int	2	
kVisionBenchmark_ToggleMode	Benchmark::Mode	0	

Vision.CropScheduler console variables

Variable	Type	Default [Min .. Max]	Description
kCropScheduler_MaxMarkerDetectionDist_mm	float	500.0 [1.0 .. 1000.0]	

Vision.General console variables

Variable	Type	Default [Min .. Max]	Description
kDisplayEyeContactInMirrorMode	bool	false	
kDisplayMarkerNames	bool	false	
kDisplayUndistortedImages	bool	false	
kKeepDrawingSalientPointsFor_ms	unsigned	0	
kMaxExpectedTimeBetweenCapturedFrames_ms	unsigned	500	
kSendDebugImages	bool	true	
kSendUndistortedImages	bool	false	
kSimulateDroppedFrameFraction	float	0.0 [0.0 .. 1.0]	
kVisionSystemSimulatedDelay_ms	unsigned	0	For testing artificial slowdowns of the vision thread
kVisualizeObservedMarkersIn3D	bool	false	

Vision.IAsyncRunner console variables

Variable	Type	Default [Min .. Max]	Description
kIAsyncRunner_OrigImageSubsample	int	1 [1 .. 2]	
kIAsyncRunner_SaveImages	int	0	Off,Save Resized,Save Original Size

Vision.Illumination console variables

Variable	Type	Default	Description
kEnableExtraIlluminationDetectorDebug	bool	false	

Vision.ImageCompositor console variables

Variable	Type	Default	Description
kImageHistogramSubsample	unsigned	4	

Vision.LaserPointDetector console variables

Variable	Type	Default [Min .. Max]	Description
kLaser_darkSurroundRadiusFraction	float	2.5	
kLaser_darkThresholdFraction_darkExposure	float	0.7 [0.0 .. 1.0]	
kLaser_darkThresholdFraction_normalExposure	float	0.9 [0.0 .. 1.0]	
kLaser_DrawDetectionsInCameraView	bool	false	
kLaser_highThreshold_darkExposure	unsigned	160	
kLaser_highThreshold_normalExposure	unsigned	240	
kLaser_lowThreshold_darkExposure	unsigned	128	
kLaser_lowThreshold_normalExposure	unsigned	235	
kLaser_maxRadius_pix	float	25.0	
kLaser_MaxSurroundStdDev	int	25	
kLaser_minRadius_pix	float	2.0	
kLaser_saturationBoundingBoxFraction	float	1.25	
kLaser_saturationThreshold_green	float	15.0	
kLaser_saturationThreshold_red	float	30.0	
kLaser_scaleMultiplier	int	2 [1 .. 8]	
kLaserDetectionDebug	int	0	

Vision.MarkerDetection console variables

Variable	Type		Default [Min .. Max]	Description
kBodyTurnSpeedThreshBlock_degs	float		30.0	
kHeadTurnSpeedThreshBlock_degs	float		10.0	
kMarkerDetector_CropWidthFraction	float		0.65 [0.5 .. 1.0]	This is fraction of full width we use with the CropScheduler to crop the image for marker detection.
kMarkerDetector_ScaleMultiplier	int		2	
visionSystem	kMarkerDetector_VizCropScheduler	bool	false	

Vision.MirrorMode console variables

Variable	Type	Default [Min .. Max]	Description
kDisplayExposureInMirrorMode	bool	true	
kDisplayFacesInMirrorMode	bool	true	
kDisplayMarkerNamesScale	float	0.0 [0.0 .. 1.0]	
kDisplayMarkersInMirrorMode	bool	true	
kDisplaySalientPointsInMirrorMode	bool	true	
kDrawMirrorModeSalientPointsFor_ms	int	0	
kMirrorModeFaceDebugFontScale	float	0.5 [0.1 .. 1.0]	
kMirrorModeGamma	float	1.0	

Vision.MotionDetection console variables

Variable	Type	Default [Min .. Max]	Description
kMotionDetection_BlurFilterSize_pix	unsigned	21	
kMotionDetection_CentroidPercentileX	float	0.5	In image coordinates
kMotionDetection_CentroidPercentileY	float	0.5	In image coordinates
kMotionDetection_DrawGroundDetectionsInCameraView	bool	false	
kMotionDetection_GroundCentroidPercentileX	float	0.05	In robot coordinates. Most important for pounce: distance from robot
kMotionDetection_GroundCentroidPercentileY	float	0.50	In robot coordinates
kMotionDetection_LastMotionDelay_ms	unsigned	500	
kMotionDetection_MaxBodyAngleChange_deg	float	0.1	
kMotionDetection_MaxHeadAngleChange_deg	float	0.1	
kMotionDetection_MaxPoseChange_mm	float	0.5	
kMotionDetection_MinAreaForMotion_pix	unsigned	500	
kMotionDetection_MinAreaFraction	float	1./225.	1/15 of each image dimension
kMotionDetection_MinBrightness	unsigned		
kMotionDetection_MorphologicalSize_pix	unsigned	20	
kMotionDetection_RatioThreshold	float		
kMotionDetection_ScaleMultiplier	int	4 [1 .. 8]	
kMotionDetectionDebug	bool	false	

Vision.NeuralNetRunner console variables

Variable	Type	Default [Min .. Max]	Description
kNeuralNetRunner_PrintTimingFrequency	int	1	
kFakeCatDetectionProbability	float	0. [0 .. 1.]	Fake pet detections for testing behaviors while we don't have reliable neural net models.
kFakeDogDetectionProbability	float	0. [0 .. 1.]	Fake pet detections for testing behaviors while we don't have reliable neural net models.
kFakeHandDetectionProbability	float	0. [0 .. 1.]	Fake hand for testing behaviors while we don't have reliable neural net models.
petTracker	kRuntimePetDetectionThreshold	int	-1 [-1 .. 1000]

Vision.PreProcessing console variables

Variable	Type	Default [Min .. Max]	Description
kClaheClipLimit	int	32	
kClaheTileSize	int	4	
kClaheWhenDarkThreshold	unsigned	80	In MarkerDetectionCLAHE::WhenDark mode. Only use CLAHE when image average is less than this
kExposure_TargetPercentile	float	0 [0 .. 1.0]	0 to disable
kExposure_TargetValue	int	128 [0 .. 255]	
kLinearizeForAutoExposure	bool	false	
kMaxFractionOverexposed	float	0.8 [0 .. 1]	
kMeteringHoldTime_ms	unsigned	2000	How long to disable auto exposure after using detections to meter.
kMinCameraGain	float	0.1	
kOverExposedAdjustmentFraction	float	0.5 [0 .. 1]	
kOverExposedThreshold	unsigned	240	
kPostClaheSmooth	int	-3	0: off, +ve: Gaussian sigma, -ve & odd: Box filter size
kRollingShutterCorrectionEnabled	bool	true	
kUnderExposedThreshold	unsigned	15	
kUseCenterWeightedMetering	bool	true	
kUseCLAHE_u8	unsigned	0 [0 .. 4]	One of MarkerDetectionCLAHE enum

VisionSystem.Statistics console variables

Variable	Type	Default [Min .. Max]	Description
kImageMeanSampleInc	int	10 [1 .. 32]	Sample rate for estimating the mean of an image (increment in both X and Y)

13.8 Software Repositories

There are many software repositories for the Vector, cloud servers, and interoperating with Vector. This pages lists a few. The diagram below summarises some of the main ones:
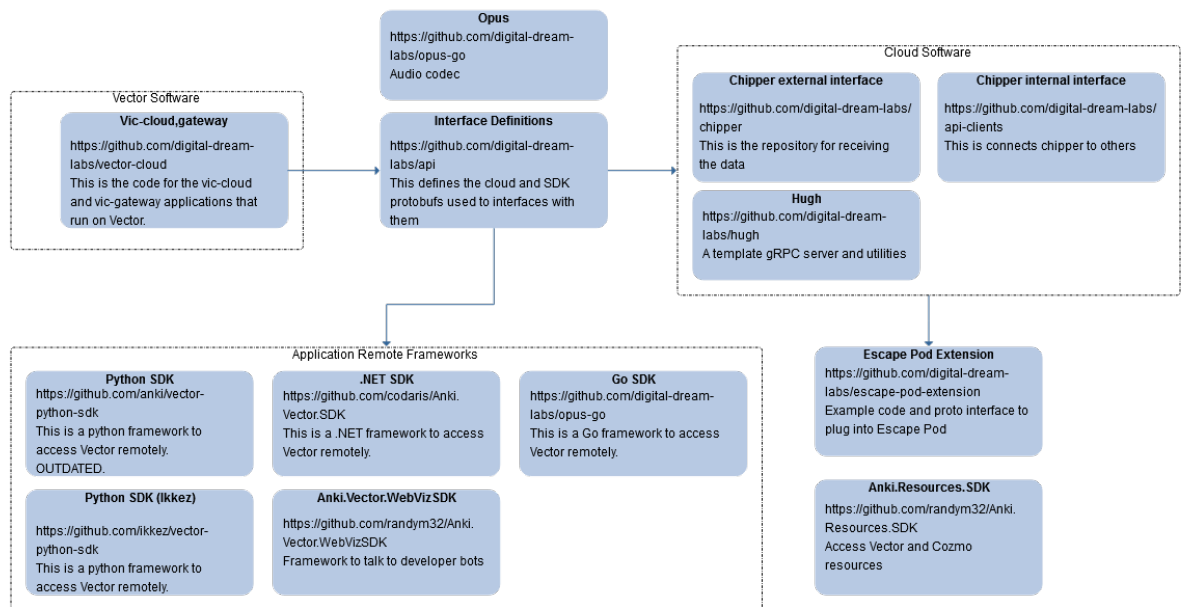
Figure: Synopsis of the main repositories for Vector's software, the cloud software, and interacting with him via an SDK

13.8.1 Programmers API, Guides and Examples

The main PC/Mobile SDK (HTTPS API) include:

- Python Communication SDK: Vector - Python SDK

- The original python SDK This is a python framework to access Vector remotely. *OUTDATED*

- C# Communication SDK: Anki.Vector.SDK This is a .NET framework to access Vector remotely from a Windows, Linux or Mac OS computer.

- C# Anki.Vector.WebVizSDK to access the WebViz related information in developer builds.

- C# Anki.Resources.SDK to access, analyze local (that is, on your computer) copies of the Vectors' application resources/ assets

- Go SDK

  This is a Go-based API to access Vector remotely.

- See the SDKs above for examples how to use each

13.8.2 Other interface-related repositories

- The Escape Pod Extension framework allows extending the EscapePod to support other voice-command features / connections to cloud servers. The repository includes examples.

- The https://github.com/digital-dream-labs/api repository defines the cloud and SDK protobufs used to interfaces with them. The information in this repository is used by both the robot and the cloud.

- The OPUS audio code is used to encode and transport the spoken audio to the cloud (Chipper) and then decode it on the cloud server.

13.8.3 The software running on Vector

- Vector-cloud This is the code for the vic-cloud and vic-gateway applications that run on Vector.

- Chipper This is the repository for go-based server receiving data from Vector.

- The api-clients reposistory holds the interfaces and tools that connect Chipper to others modules on the cloud server.

- The hugh repository holds a framework that acts as a template gRPC server and utilities.

Source Code Location for each Program

Main service	repo	location
vic-cloud	repo	https://github.com/digital-dream-labs/vector-cloud
	Offboard Vision	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/offboard_vision
	Token client	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/token
	Voice stream	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/voice
vic-gateway	repo	https://github.com/digital-dream-labs/vector-cloud in /gateway
vic-switchboard		
vic-aim		
vic-engine		
vic-robot		

13.8.4 Bluetooth LE tools

Bluetooth LE implementations. There isn't an SDK for the Bluetooth LE protocol, but there are a few implementations that you might wish to look at/reuse:

- OS-X Objective-C

- linux & C

- Chrome & Javascript and here

Bluetooth LE implementations. There isn't an SDK for the Bluetooth LE protocol, but there are a few implementations that you might wish to look at/reuse:

- OS-X Objective-C

- linux & C

- Chrome & Javascript and here

13.9 Source Code Location for each Program

Main service	repo	location
vic-cloud	repo	https://github.com/digital-dream-labs/vector-cloud
	Offboard Vision	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/offboard_vision
	Token client	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/token
	Voice stream	https://github.com/digital-dream-labs/vector-cloud/tree/main/internal/voice
vic-gateway	repo	https://github.com/digital-dream-labs/vector-cloud in /gateway
vic-switchboard		
vic-aim		
vic-engine		
vic-robot		

13.10 URLs listed in the code

There are a bunch of URLs in Anki binary files… these may be part of schemas, random comments, etc.

13.10.1 Servers

- http://s3.amazonaws.com/doc/2006-03-01/
- https://developer.amazon.com/docs/alexa-voice-service/settings.html#settingsupdated The listed documentation for Alexa services
- anki.com/v github.com/anki/sai-token-service/proto/tokenpb
- support.anki.com

13.10.2 Github repos

- github.com/anki/sai-chipper-voice/client/chipper
- github.com/anki/sai-chipper-voice/proto/anki/chipperpb
- github.com/anki/opus-go/libopus
- github.com/anki/opus-go/ogg
- github.com/aws/aws-sdk-go/private/protocol/query
- github.com/aws/aws-sdk-go/private/protocol/query
- github.com/aws/aws-sdk-go/aws/credentials/ec2rolecreds
- github.com/aws/aws-sdk-go/private/protocol/eventstream
- github.com/aws/aws-sdk-go/private/protocol/xml/xmlutil
- github.com/aws/aws-sdk-go/vendor/github.com/go-ini/ini
- github.com/aws/aws-sdk-go/aws/credentials/ec2rolecreds
- github.com/aws/aws-sdk-go/private/protocol/eventstream
- github.com/aws/aws-sdk-go/private/protocol/xml/xmlutil
- github.com/aws/aws-sdk-go/vendor/github.com/go-ini/ini
- github.com/aws/aws-sdk-go/private/protocol/query/queryutil
- github.com/aws/aws-sdk-go/vendor/github.com/jmespath/go-jmespath
- github.com/grd/ogg
- github.com/google/uuid
- github.com/cenkalti/backoff
- github.com/dgrijalva/jwt-go
- github.com/gwatts/rootcerts
- github.com/aws/aws-sdk-go/aws
- github.com/golang/protobuf/proto
- github.com/aws/aws-sdk-go/aws/csm

- github.com/golang/protobuf/ptypes

13.10.3 Other

- google.golang.org/genproto/googleapis/rpc/status
- http://logo.verisign.com/vslogo.gif
- google.golang.org/grpc/peer
- google.golang.org/grpc/status
- google.golang.org/grpc/balancer
- google.golang.org/grpc/encoding
- google.golang.org/grpc/metadata
- google.golang.org/grpc/resolver
- google.golang.org/grpc/keepalive
- google.golang.org/grpc/transport
- google.golang.org/genproto/googleapis/rpc/status

13.10.4 Some built in certificates?

- http://www.certplus.com/CRL/class2.crl
- http://fedir.comsign.co.il/crl/ComSignCA.crl
- http://crl.securetrust.com/STCA.crl
- http://crl.netsolssl.com/NetworkSolutionsCertificateAuthority.crl
- http://www.trustdst.com/certificates/policy/ACES-index.html
- http://crl.comodoca.com/COMODOCertificationAuthority.crl
- http://crl.xrampsecurity.com/XGCA.crl
- www.xrampsecurity.com
- http://crl.comodoca.com/AAACertificateServices.crl
- http://crl.comodo.net/AAACertificateServices.crl
- http://www.usertrust.com
- http://crl.usertrust.com/UTN-USERFirst-ClientAuthenticationandEmail.crl
- http://logo.verisign.com/vslogo.gif
- http://www.chambersign.org
- http://repository.swisssign.com/
- https://ocsp.quovadisoffshore.com
- http://www.quovadis.bm
- http://www.firmaprofesional.com/cps
- http://www.certicamara.com/dpc/0Z
- http://www.quovadisglobal.com/cps
- http://www.startssl.com/policy.pdf

• http://www.startssl.com/intermediate.pdf

13.11 Channels

I'm not sure what these are. They may be part of the logging of information and routing it internally and to a log file.

This note is to help gather a description of each of the channels. This format lets us gather information on them, and help understand where they fit in.

Channel	Description
Actions	
AIWhiteboard	
Alexa	
Audio	
Behaviors	
BlockPool	
BlockWorld	
CpuProfiler	
FaceRecognizer	
FaceWorld	
JdocsManager	the cloud storage?
Keyboard	
MessageProfiler	
Microphones	
NeuralNets	
PerfMetric	
PoseConfirmer	
SpeechRecognizer	
VisionComponent	
VisionSystem	

13.12 Communication trouble behaviors

Summary: These behaviors play animations when there is communication problems. (These are self-maintenance behaviors)

These behaviors are not called by a behavior tree configuration file. Instead they are are invoked by the internal behavior implementation, in the *BehaviorReactToVoiceCommand* class.

13.12.1 No Wifi behavior

The *NoWifi* (class *DispatcherQueue*) behavior is used to animate Vectors face when he is unable to connect to a Wifi SSID.
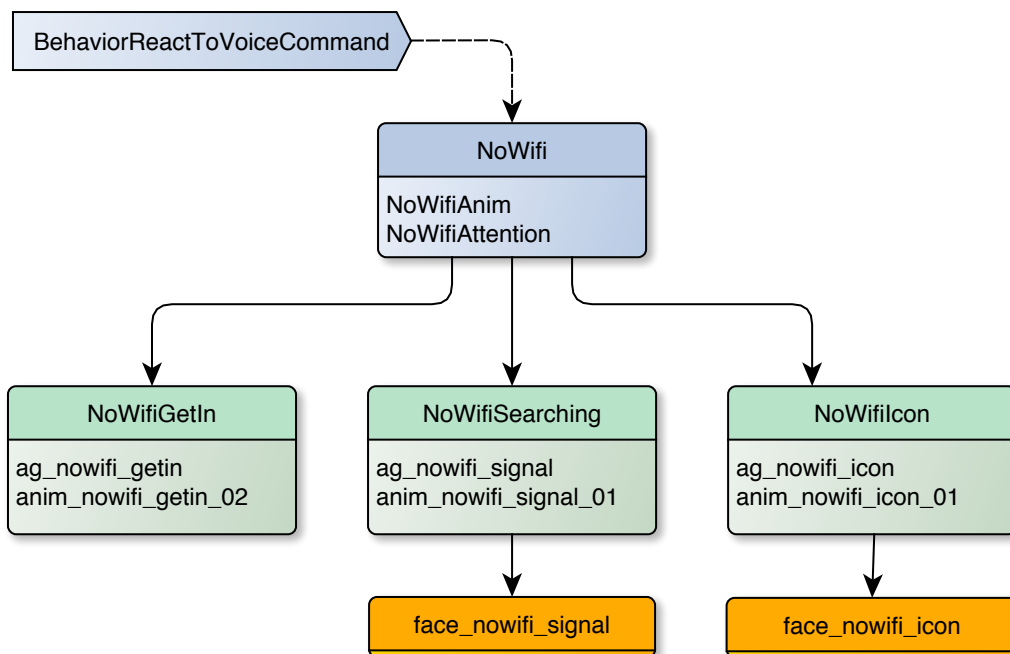

Figure: The No Wifi behavior tree

The behavior file is located at:

```
behaviors/victorBehaviorTree/noWifi.json
```

The behavior will play three animations:

1. The *NoWifiGetIn* animation when the behavior starts

2. The *NoWifiSearching" animation while Vector is looking for a Wifi SSID; This will play the* face_nowifi_signal* sprite sequence.

3. The*NoWifiIcon* animation when the above animation completes This will play the *face_nowifi_icon* sprite sequence.

4. Then the loop will repeat from step 2.

13.12.2 No Cloud behavior

The *NoCloud* (class *DispatcherQueue*) behavior is used to animate Vectors face when he is able to connect to a Wifi SSID, but unable to connect to the remote server.
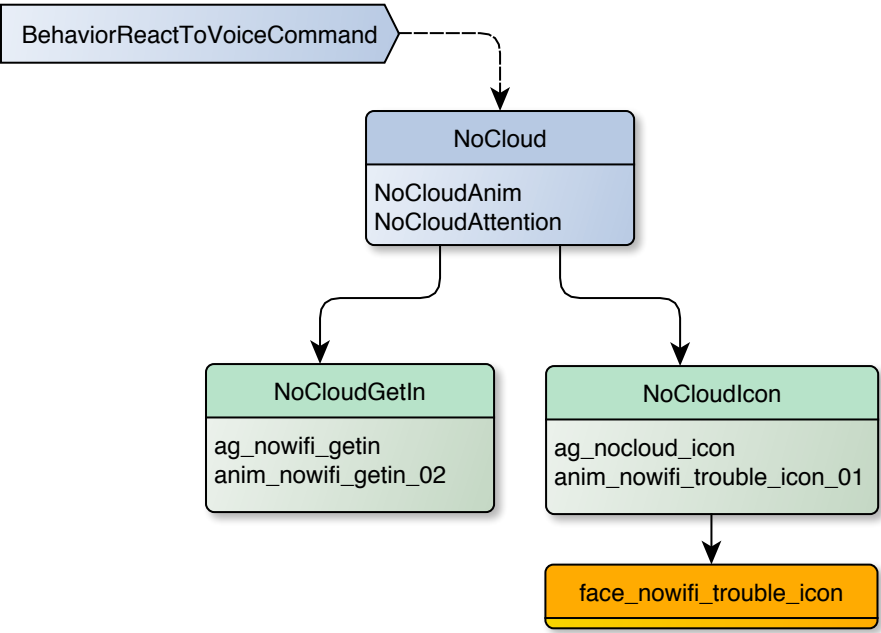
Figure: The No Cloud behavior tree

The behavior file is located at:

```
behaviors/victorBehaviorTree/noCloud.json
```

The *NoCloudAnim* behavior is used to animate the face. The behavior will play two animations:

1. The *NoCloudGetIn* animation when the behavior starts. Note that this is same as the *NoWifiGetIn* animation group.

2. The *NoCloudIcon* animation will loop thereafter. This will play the *face_nowifi_trouble_icon* sprite sequence.

The *NoCloudAttention* is used to transfer attention back(?) to the previous task.

13.12.3 Other variations

Curiosly there is another animation -- the *anim_cloud_icon* animation -- this not used. It is not part of an animation group, but probably was part of the *NoCloudIcon* animation group. It uses a *face_nocloud_icon* sprite animation.

13.12.4 Change history synopsis

Date	Change
2020-12-2	Created

13.13 Source Files referenced in the binaries

The following source code files were referenced in the binaries:

File
../../../animProcess/src/cozmoAnim/alexa/alexa.cpp
../../../animProcess/src/cozmoAnim/alexa/alexaClient.cpp
../../../animProcess/src/cozmoAnim/alexa/alexaImpl.cpp
../../../animProcess/src/cozmoAnim/alexa/media/alexaMediaPlayer.cpp
../../../animProcess/src/cozmoAnim/animation/animationStreamer.cpp
../../../animProcess/src/cozmoAnim/audio/sdkAudioComponent.cpp
../../../animProcess/src/cozmoAnim/faceDisplay/faceInfoScreenManager.cpp
../../../animProcess/src/cozmoAnim/micData/micDataSystem.cpp
../../../animProcess/src/cozmoAnim/micData/micImmediateDirection.cpp
../../../animProcess/src/cozmoAnim/showAudioStreamStateManager.cpp
../../../animProcess/src/cozmoAnim/speechRecognizer/speechRecognizerTHFSimple.cpp
../../../cannedAnimLib/baseTypes/keyframe.cpp
../../../cannedAnimLib/baseTypes/track.h
../../../cannedAnimLib/spriteSequences/spriteSequenceLoader.cpp
../../../coretech/common/engine/math/pose.cpp
../../../coretech/common/engine/math/poseBase_impl.h
../../../coretech/common/engine/math/poseOriginList.cpp
../../../coretech/common/engine/math/poseTreeNode.h
../../../coretech/common/engine/utils/recentOccurrenceTracker.cpp
../../../coretech/common/robot/array2d.h
../../../coretech/common/robot/arrayPatterns.h
../../../coretech/common/robot/arraySlices.h
../../../coretech/common/robot/interpolate.h
../../../coretech/common/robot/matrix.h
../../../coretech/common/robot/memory.cpp
../../../coretech/common/robot/sequences.h
../../../coretech/common/robot/serialize.h
../../../coretech/vision/engine/camera.cpp
../../../coretech/vision/engine/enrolledFaceEntry.cpp
../../../coretech/vision/engine/faceRecognizer_okao.cpp
../../../coretech/vision/engine/faceTrackerImpl_okao.cpp
../../../coretech/vision/engine/imageBuffer/imageBuffer.cpp
../../../coretech/vision/engine/imageCompositor.cpp
../../../coretech/vision/engine/markerDetector.cpp
../../../coretech/vision/engine/undistorter.cpp

File
../../../coretech/vision/robot/computeCharacteristicScale.cpp
../../../coretech/vision/robot/computeCharacteristicScale_binomial.cpp
../../../coretech/vision/robot/computeQuadrilaterals.cpp
../../../coretech/vision/robot/connectedComponents.h
../../../coretech/vision/robot/detectFiducialMarkers.cpp
../../../coretech/vision/robot/fiducialMarkers.cpp
../../../coretech/vision/robot/filtering.cpp
../../../coretech/vision/robot/histogram.cpp
../../../coretech/vision/robot/imageProcessing.h
../../../coretech/vision/robot/integralImage.cpp
../../../coretech/vision/robot/laplacianPeaks.cpp
../../../coretech/vision/robot/nearestNeighborLibrary.cpp
../../../coretech/vision/robot/quadRefinement.cpp
../../../coretech/vision/robot/traceBoundary.cpp
../../../coretech/vision/robot/transformations.cpp
../../../coretech/vision/shared/compositeImage/compositeImage.cpp
../../../coretech/vision/shared/compositeImage/compositeImageBuilder.cpp
../../../coretech/vision/shared/compositeImage/compositeImageLayer.cpp
../../../coretech/vision/shared/hueSatWrapper.cpp
../../../coretech/vision/shared/spriteCache/spriteWrapper.cpp
../../../coretech/vision/shared/spritePathMap.cpp
../../../coretech/vision/shared/spriteSequence/spriteSequence.cpp
../../../cubeBleClient/cubeBleClient.cpp
../../../engine/actions/actionContainers.cpp
../../../engine/actions/basicActions.cpp
../../../engine/actions/dockActions.cpp
../../../engine/actions/trackGroundPointAction.cpp
../../../engine/aiComponent/aiWhiteboard.cpp
../../../engine/aiComponent/alexaComponent.cpp
../../../engine/aiComponent/behaviorComponent/activeBehaviorIterator.cpp
../../../engine/aiComponent/behaviorComponent/asyncMessageGateComponent.cpp
../../../engine/aiComponent/behaviorComponent/behaviorComponentMessageHandler.cpp
../../../engine/aiComponent/behaviorComponent/behaviorContainer.h
../../../engine/aiComponent/behaviorComponent/behaviorExternalInterface/behaviorEventComponent.cpp
../../../engine/aiComponent/behaviorComponent/behaviorExternalInterface/behaviorExternalInterface.cpp

File
../../../engine/aiComponent/behaviorComponent/behaviorStack.cpp
../../../engine/aiComponent/behaviorComponent/behaviorSystemManager.cpp
../../../engine/aiComponent/behaviorComponent/behaviorTimers.cpp
../../../engine/aiComponent/behaviorComponent/behaviorTypesWrapper.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/alexa/behaviorAlexa.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/animationWrappers/behaviorAnimGetInLoop.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/animationWrappers/behaviorAnimSequence.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/animationWrappers/behaviorAnimSequenceWithFace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/animationWrappers/behaviorCountingAnimation.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/animationWrappers/behaviorTextToSpeechLoop.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/attentionTransfer/behaviorAttentionTransferIfNeeded.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/basicWorldInteractions/behaviorDriveOffCharger.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/basicWorldInteractions/behaviorInteractWithFaces.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/behaviorHighLevelAI.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/behaviorLookAroundInPlace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/behaviorResetState.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/behaviorStayOnChargerUntilCharged.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/blackjack/behaviorBlackJack.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/blackjack/blackJackVisualizer.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/coordinators/behaviorCoordinateWhileHeldInPalm.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/coordinators/behaviorCoordinateWhileInAir.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/coordinators/behaviorQuietModeCoordinator.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/cubeSpinner/behaviorVectorPlaysCubeSpinner.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/danceToTheBeat/behaviorDanceToTheBeat.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/devBehaviors/behaviorDevViewCubeBackpackLights.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/devBehaviors/behaviorDispatchAfterShake.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/devBehaviors/behaviorReactToBody.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/dispatch/behaviorDispatcherRerun.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/dispatch/behaviorDispatcherScoring.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/dispatch/behaviorDispatcherStrictPriorityWithCooldown.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/dispatch/iBehaviorDispatcher.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/freeplay/putDownDispatch/behaviorLookForFaceAndCube.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/freeplay/userInteractive/behaviorPuzzleMaze.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/habitat/behaviorConfirmHabitat.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/iCozmoBehavior.cpp

File
../../../engine/aiComponent/behaviorComponent/behaviors/internalStatesBehavior.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/meetCozmo/behaviorEnrollFace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/meetCozmo/behaviorRespondToRenameFace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/photoTaking/behaviorAestheticallyCenterFaces.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/photoTaking/behaviorTakeAPhotoCoordinator.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/prDemo/behaviorPRDemoBase.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorCheckForAndReactToSalientPoint.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToCliff.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToDarkness.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToMotion.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToPutDown.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToUnexpectedMovement.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/reactions/behaviorReactToVoiceCommand.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/robotDrivenDialog/behaviorPromptUserForVoiceCommand.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/sdkBehaviors/behaviorSDKInterface.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/simpleFaceBehaviors/behaviorDriveToFace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/simpleFaceBehaviors/behaviorFindFaceAndThen.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/simpleFaceBehaviors/behaviorSayName.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/sleeping/behaviorSleepCycle.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/timer/behaviorProceduralClock.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/timer/behaviorTimerUtilityCoordinator.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/timer/behaviorWallTimeCoordinator.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/userDefinedBehaviorTree/behaviorUserDefinedBehaviorSelector.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/victor/behaviorReactToTouchPetting.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/victor/behaviorReactToUnclaimedIntent.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/victor/behaviorTrackFace.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/weather/behaviorCoordinateWeather.cpp
../../../engine/aiComponent/behaviorComponent/behaviors/weather/behaviorDisplayWeather.cpp
../../../engine/aiComponent/behaviorComponent/behaviorsBootLoader.cpp
../../../engine/aiComponent/behaviorComponent/iBehavior.cpp
../../../engine/aiComponent/behaviorComponent/stackMonitors/stackCycleMonitor.cpp
../../../engine/aiComponent/behaviorComponent/userDefinedBehaviorTreeComponent/userDefinedBehaviorTreeComponent.cpp
../../../engine/aiComponent/behaviorComponent/userIntentComponent.cpp
../../../engine/aiComponent/behaviorComponent/userIntentMap.cpp
../../../engine/aiComponent/behaviorComponent/weatherIntents/weatherConditionRemaps.cpp

File
../../../engine/aiComponent/beiConditions/beiConditionFactory.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionAnyStimuli.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionBatteryLevel.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionBecameTrueThisTick.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionBehaviorTimer.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionCliffDetected.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionCompound.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionEmotion.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionFeatureGate.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionObjectKnown.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionOffTreadsState.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionProxInRange.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionRobotPitchInRange.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionRobotRollInRange.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionSalientPointDetected.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionSettingsUpdatePending.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionSimpleMood.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionTimedDedup.cpp
../../../engine/aiComponent/beiConditions/conditions/conditionTimerInRange.cpp
../../../engine/aiComponent/beiConditions/conditions/iConditionUserIntent.cpp
../../../engine/aiComponent/timerUtility.cpp
../../../engine/block.cpp
../../../engine/blockWorld/blockWorld.cpp
../../../engine/comms/robotConnectionData.cpp
../../../engine/components/animationComponent.cpp
../../../engine/components/battery/batteryComponent.cpp
../../../engine/components/cubes/cubeCommsComponent.cpp
../../../engine/components/cubes/cubeConnectionCoordinator.cpp
../../../engine/components/cubes/cubeInteractionTracker.cpp
../../../engine/components/cubes/cubeLights/cubeLightAnimationHelpers.cpp
../../../engine/components/mics/beatDetectorComponent.cpp
../../../engine/components/pathComponent.cpp
../../../engine/components/powerStateManager.cpp
../../../engine/components/sdkComponent.cpp
../../../engine/components/variableSnapshot/variableSnapshotComponent.h

File
../../../engine/components/visionComponent.cpp
../../../engine/drivingAnimationHandler.cpp
../../../engine/faceWorld.cpp
../../../engine/moodSystem/emotion.cpp
../../../engine/moodSystem/moodDecayEvaluator.cpp
../../../engine/moodSystem/moodManager.cpp
../../../engine/moodSystem/staticMoodData.cpp
../../../engine/navMap/mapComponent.cpp
../../../engine/petWorld.cpp
../../../engine/robot.cpp
../../../engine/robot.h
../../../engine/robotDataLoader.cpp
../../../engine/vision/imageSaver.cpp
../../../engine/vision/visionSystem.cpp
../../../generated/proto/external_interface/alexa.pb.cc
../../../generated/proto/external_interface/behavior.pb.cc
../../../generated/proto/external_interface/cube.pb.cc
../../../generated/proto/external_interface/messages.pb.cc
../../../generated/proto/external_interface/nav_map.pb.cc
../../../generated/proto/external_interface/response_status.pb.cc
../../../generated/proto/external_interface/settings.pb.cc
../../../generated/proto/external_interface/shared.pb.cc
../../../lib/das-client/src/DAS.cpp
../../../lib/das-client/src/dasAppender.cpp
../../../lib/util/source/anki/util/../util/cladHelpers/cladEnumToStringMap.h
../../../lib/util/source/anki/util/../util/entityComponent/componentWrapper.h
../../../lib/util/source/anki/util/../util/entityComponent/dependencyManagedEntity.h
../../../lib/util/source/anki/util/../util/entityComponent/entity.h
../../../lib/util/source/anki/util/../util/entityComponent/iDependencyManagedComponent.h
../../../lib/util/source/anki/util/entityComponent/dependencyManagedEntity.h
../../../lib/util/source/anki/util/random/randomIndexSampler.cpp
../../../platform/switchboard/anki-wifi/connmanbus.c
pffft.cpp
randombytes/randombytes.c
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/config_linux.go

File
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/ipc_manager.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/main.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/message_handler.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/multilimiter.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/switchboard_proxy.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/gateway/tokens.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/cloudproc/cloudproc.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/cloudproc/opts.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/cloudproc/server.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/config/urls.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/ipc/baseconn.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/ipc/baseserver.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/ipc/dgram.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/ipc/path_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/ipc/unixgram.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/jdocs/client.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/jdocs/jdocs.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/jdocs/opts.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/jdocs/server.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/jdocs/translate.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/log/das.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/log/log.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/log/log_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/logcollector/cladhandler.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/logcollector/logcollector.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/logcollector/opts.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/logcollector/server.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/cert.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/crash_reporter.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/error.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/esn.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/esn_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/robot/loguploader/loguploader.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/accessor.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/client.go

File
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/errorhandler.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/handlers.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/identity/getcert_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/identity/identity.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/options.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/queue.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/refresher.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/sts.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/token/token.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/util/grpc.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/util/grpc_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/util/multierror.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/util/util.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/options.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/process.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/receive.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/send.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/api.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/conn_chipper.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/connect.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/context.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/init.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/opts.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/anki/voice/stream/opts_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/libopus/decoder.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/libopus/encoder.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/libopus/errors.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/ogg/ogg.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/opus/header.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/opus-go/opus/opus.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-chipper-voice/client/chipper/chipper.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-chipper-voice/client/chipper/options.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-chipper-voice/proto/anki/chipperpb/chipperpb.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-go-util/testutils/testtime/time.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-jdocs/proto/jdocspb/jdocs.pb.go

File
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-token-service/client/clienthash/hash.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-token-service/model/token.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/anki/sai-token-service/proto/tokenpb/token.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/go/src/github.com/aws/aws-sdk-go/aws/awserr/error.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/process/main.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/cloud/process/platform_vicos.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/cloud/common.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/cloud/docs.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/cloud/logcollector.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/cloud/mic.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/cloud/token.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/gateway/messageExternalToRobot.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/gateway/messageRobotToExternal.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/gateway/shared.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/cladgo/src/clad/gateway/switchboard.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/alexa.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/behavior.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/cube.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/extensions.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/external_interface.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/external_interface.pb.gw.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/messages.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/nav_map.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/response_status.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/settings.pb.go
/mnt/devhomes/build/work/83941694d19f355d/anki/victor/generated/go/src/proto/external_interface/shared.pb.go
_cgo_gotypes.go

13.13.1 Signal Essence files

{Am I the only one who things "Signal Essence" sounds like a perfume name?}

File
../../../EXTERNALS/anki-thirdparty/signalEssence/v008/vicos/project/anki_victor/mmif_proj.c
../../../EXTERNALS/anki-thirdparty/signalEssence/v008/vicos/project/anki_victor/policy_actions.c
../../../EXTERNALS/anki-thirdparty/signalEssence/v008/vicos/project/anki_victor_vad/nfbin_f32_anki.c
../../../EXTERNALS/anki-thirdparty/signalEssence/v008/vicos/project/anki_victor_vad/svad.c
../../../EXTERNALS/opencv/vicos/include/opencv2/core/mat.inl.hpp
../../../se_lib/aec_common.c
../../../se_lib/aec_msu.c
../../../se_lib/aec_pbfd.c
../../../se_lib/aec_stereo.c
../../../se_lib/aec_tapered_wts.c
../../../se_lib/aec_td.c
../../../se_lib/aecmonitor.c
../../../se_lib/avepower_i16.c
../../../se_lib/buffer_composer.c
../../../se_lib/cl_agc.c
../../../se_lib/cl_agc_i16.c
../../../se_lib/conv.c
../../../se_lib/dcremove.c
../../../se_lib/dcremove_f32.c
../../../se_lib/decimate31.c
../../../se_lib/downsampn.c
../../../se_lib/fdanalyze.c
../../../se_lib/fdechomodel.c
../../../se_lib/fdemphasis.c
../../../se_lib/fdsearch.c
../../../se_lib/fdsearch_winner.c
../../../se_lib/float_dft.c
../../../se_lib/float_dft_fftpack.c
../../../se_lib/float_dft_pffft.c
../../../se_lib/frdelay.c
../../../se_lib/gainest.c
../../../se_lib/highpass_filter_array.c
../../../se_lib/interpn.c
../../../se_lib/leakyave.c
../../../se_lib/lec.c

File
../../../se_lib/lrhpf.c
../../../se_lib/meta_aec.c
../../../se_lib/meta_fda.c
../../../se_lib/mmfx.c
../../../se_lib/mmfxcalibactions.c
../../../se_lib/mmfxspatialfilter.c
../../../se_lib/mmif_helper.c
../../../se_lib/mmpreprocessor.c
../../../se_lib/mmvalidate.c
../../../se_lib/morpho.c
../../../se_lib/multiaec.c
../../../se_lib/multichan_delay.c
../../../se_lib/multichan_delay_f32.c
../../../se_lib/narrowband_noisegen.c
../../../se_lib/nfbin_f32.c
../../../se_lib/nrgainv.c
../../../se_lib/output_injector.c
../../../se_lib/ref_proc.c
../../../se_lib/rfir.c
../../../se_lib/rfir_f.c
../../../se_lib/sampledelayqueue.c
../../../se_lib/sampledelayqueue_f32.c
../../../se_lib/sat_detector.c
../../../se_lib/scratch_mem.c
../../../se_lib/se_crossover.c
../../../se_lib/se_dft.c
../../../se_lib/se_dft_fftpack.c
../../../se_lib/se_dft_fxp.c
../../../se_lib/se_dft_pffft.c
../../../se_lib/se_dft_qf.c
../../../se_lib/se_diag.c
../../../se_lib/se_nr.c
../../../se_lib/se_rcv.c
../../../se_lib/subbandsplitter.c
../../../se_lib/system_tests.c

File
../../../se_lib/tdinterp.c
../../../se_lib/trackfilters.c
../../../se_lib/upsamplen.c
../../../se_lib/vadd.c
../../../se_lib/vavepower.c
../../../se_lib/vavepowerrms_i16.c
../../../se_lib/vcmul.c
../../../se_lib/vcrmul_i16_i32.c
../../../se_lib/vdotproduct_i16_i32.c
../../../se_lib/vdotproductq15_i16.c
../../../se_lib/vdotproductswithleftshift_q15_i16.c
../../../se_lib/vfill_i16.c
../../../se_lib/vfill_i32.c
../../../se_lib/vfloatlib.c
../../../se_lib/vgen_exp_ramp.c
../../../se_lib/vgenctone.c
../../../se_lib/vgentone.c
../../../se_lib/vgetindex.c
../../../se_lib/vgetvalue.c
../../../se_lib/vinvertorder.c
../../../se_lib/vleftshifts_i16.c
../../../se_lib/vlimitmin.c
../../../se_lib/vmax.c
../../../se_lib/vmin.c
../../../se_lib/vmmadd.c
../../../se_lib/vmove_i16.c
../../../se_lib/vmove_i32.c
../../../se_lib/vmovesrcstride_i16.c
../../../se_lib/vmul.c
../../../se_lib/vpower_i16_i32.c
../../../se_lib/vpowerwithexponent_i16_i32.c
../../../se_lib/vscale.c
../../../se_lib/vsub_i16.c
../../../se_lib/vsum_i32.c
../../../se_lib/vtrackupavedown.c

File
../../../se_lib/win_fcns.c
../../../se_lib/winbufdft.c
../../../se_lib/wola.c

13.13.2 Google BreakPad and Minidump

File
./src/client/linux/handler/minidump_descriptor.h
./src/client/linux/minidump_writer/directory_reader.h
./src/client/linux/minidump_writer/line_reader.h
./src/client/linux/minidump_writer/proc_cpuinfo_reader.h
./src/client/minidump_file_writer-inl.h
./src/common/linux/elfutils-inl.h

13.14 Mandatory physical reactions

Summary: Reactions to physical circumstances that can't be skipped.
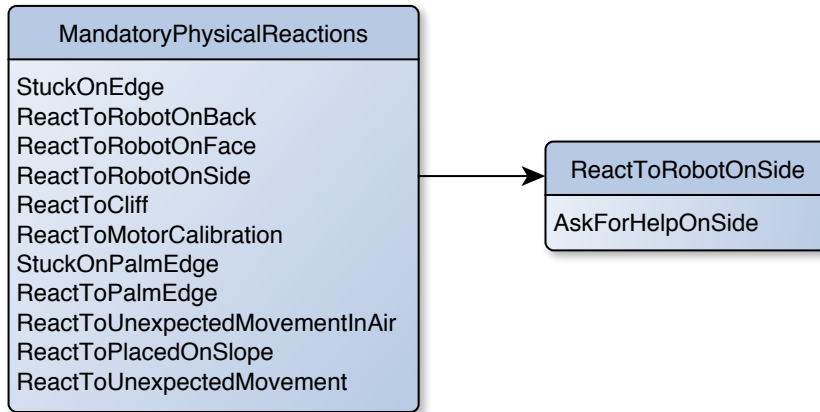

Figure: The mandatory physical reactions behavior tree

The *MandatoryPhysicalReactions* is used to react when Vector is:

- On his side, face, or back

- Encounters a cliff

- Is stuck on cliff edge

- Needs to calibrate the head or lift motors

- Some interaction with being held in the palm of a hand

- Is falling or is being picked up

- Is on a slope

- Is being moved around

The behavior file is located at:

```
behaviors/victorBehaviorTree/reactions/mandatoryPhysicalReactions.json
```

For the most part, the *MandatoryPhysicalReactions* provides a way to prioritize the internal C++ implementation; most of these behavior references do not link to other behaviors or animations.

The only one of interest is *ReactToRobotOnSide* which initiates *AskForHelpOnSide* behavior to call for someone to come and help.

- *ReactToCliff* has configuration for how fast and far Vector backs up in responds to a a cliff.

- *ReactToUnexpectedMovement* has configuration for how fast and far Vector backs up when he experiences being moved by someone; as well as how much he can respond.

13.14.1 Change history synopsis

Date	Change
2020-12-1	Created

13.15 Power management behaviors

Summary: The behaviors related to turning off, handling very low-battery, sleeping, and other battery-related items. (These are self-maintenance behaviors)

This note describes the power management-related behaviors. Please refer to the Technical Reference Manual for a description of Vector's internal power states, management, and sleep debt.

The main power management behaviors are launched at a very high-level. These are launched by the *ModeSelector* (class *DispatcherStrictPriority*) behavior. This behavior is invoked by:

- *InitNormalOperation* behavior during start of normal behavior

- *AcousticTestMode* behavior during aoustic testing at the factory

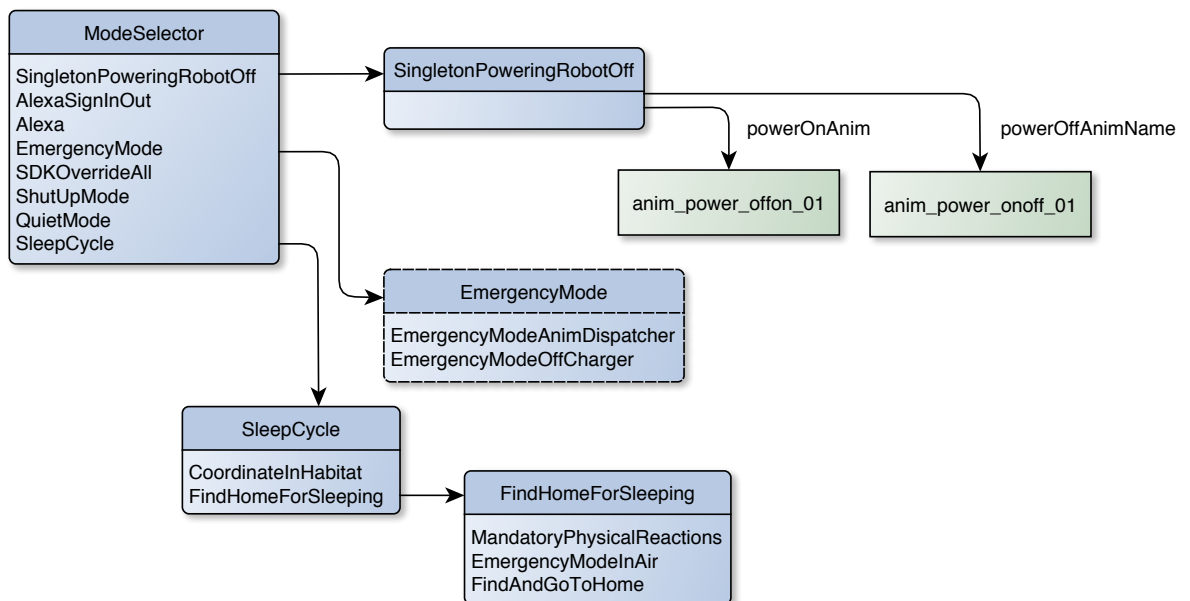- *DevBaseBehaviorInternal* behavior during start of developer mode

Figure: The power management behavior tree

The behavior file is located at:

```
behaviors\victorBehaviorTree\modeSelector.json
```

The behavior tree system gives things in explicit priority order. The higher item in the list has more priority than the lower; when an item calls out to other nodes, all of those still have higher priority than the items lower than the original one.

The mode selector:

- Handle powering off (the highest priority)

- Alexa-related behaviors (not related to power management)

- Handle overheating (in emergency mode)

- SDK override (not related to power management)

- Being quiet related behaviors (not related to power management)

- Handle low battery and attempt to return to charger... if unable, turn off

- A sleep/quiet state manager. Regular behaviors and interactions are started in this behavior tree, so are at the lowest priority.

13.15.1 Powering off

The highest priority behavior is the *SingletonPoweringRobotOff* (class *PoweringRobotOff*) behavior, that animates Vector while he is the process of turning off.

Name	Type	Value	Description
powerButtonHeldToActivate_ms	int	250 ms	The minimum time that the power button must be held down to activate the power off sequence.
powerOnAnimName	animation	anim_power_offon_01	The animation to play if the button is released and Vector will resume.
powerOffAnimName	animation	anim_power_onoff_01	The animation to play while powering off.

Table: PoweringRobotOff configuration parameters

If the power button is released before Vector has turned off, the behavior begins the process of resuming,

The animation reference is unusual. It doesn't refer to the trigger of an animation group. Instead it refers to a specific animation.

13.15.2 Sleep

At the opposite end -- the *lowest* priority -- is the *SleepCycle* (class *SleepCycle*) behavior. The behavior file is located at:

```
behaviors/victorBehaviorTree/highLevelDelegates/sleeping/sleepCycle.json
```

This behavior arbitrates between:

- Vector autonomously exploring

- Interacting with a person (outside of Alexa).

- And going into a sleep state

If Vector has no reason to sleep, this behavior lets the *CoordinateInHabitat* behavior. If it decides to sleep (the decision is made in the C++ code) It initiates the *FindHomeForSleeping* behavior to drive to the charging dock, if possible, to sleep. (In turn it invokes *MandatoryPhysicalReactions* to respond to environmental hazards while driving around.)

This has a condition that keeps it in sleep, even if there is a higher priority interaction, if:

- The battery level is low,

- The temperature is high, or

- Charging is stopped because it is too hot.

Depending, it will initiate the looking for home to go to sleep. This is the *FindHomeForSleeping* behavior.

Things that wake Vector up from the different kinds of sleep:

Sleep state	Things that wake Vector
HeldInPalmSleep	Being jolted, touched, picked up (out of the hand), SDK interaction, the timer, and voice commands.
LightSleep	Being jolted, poked, touched, or picked up; any sound or the lights coming on; SDK interaction, the timer, and voice commands.
DeepSleep	Being touched, or picked up; SDK interaction, the timer, and voice commands.

Table: That wake Vector from sleep.

This behavior includes a decision tree that sets a `reason` code for based on sensors that have trigger. That reason code is used above. By editting this behavior's decision tree, you can adjust how sensitive he is to conditions like touch, poking, illumination to wake him from sleep.

Note: In power save mode -- a lower sleep state -- the camera is turned off, so Vector is not sensitive to light.

Driving to the charging dock to sleep

When Vector is going into a sleep state, *FindHomeForSleeping* (class *DispatcherStrictPriority*) behavior to drive to the charging dock, if possible, to sleep. The behavior file is located at:

```
behaviors/victorBehaviorTree/highLevelDelegates/sleeping/findHomeForSleeping.json
```

This behavior stops whatever else is going on, and runs a subset of navigation and driving related behaviors.

13.15.3 Emergency Mode

Emergency mode is quite complex, and handles conditions where:

- The battery level is low,

- The temperature is high, or

- Charging is stopped because it is too hot.

The way to exit emgency mode is for the battery level to rise above the low threshold, and for temperature to cool below the hot threshold.

This is controlled by the *EmergencyMode* (class *DispatcherQueue*) behavior.
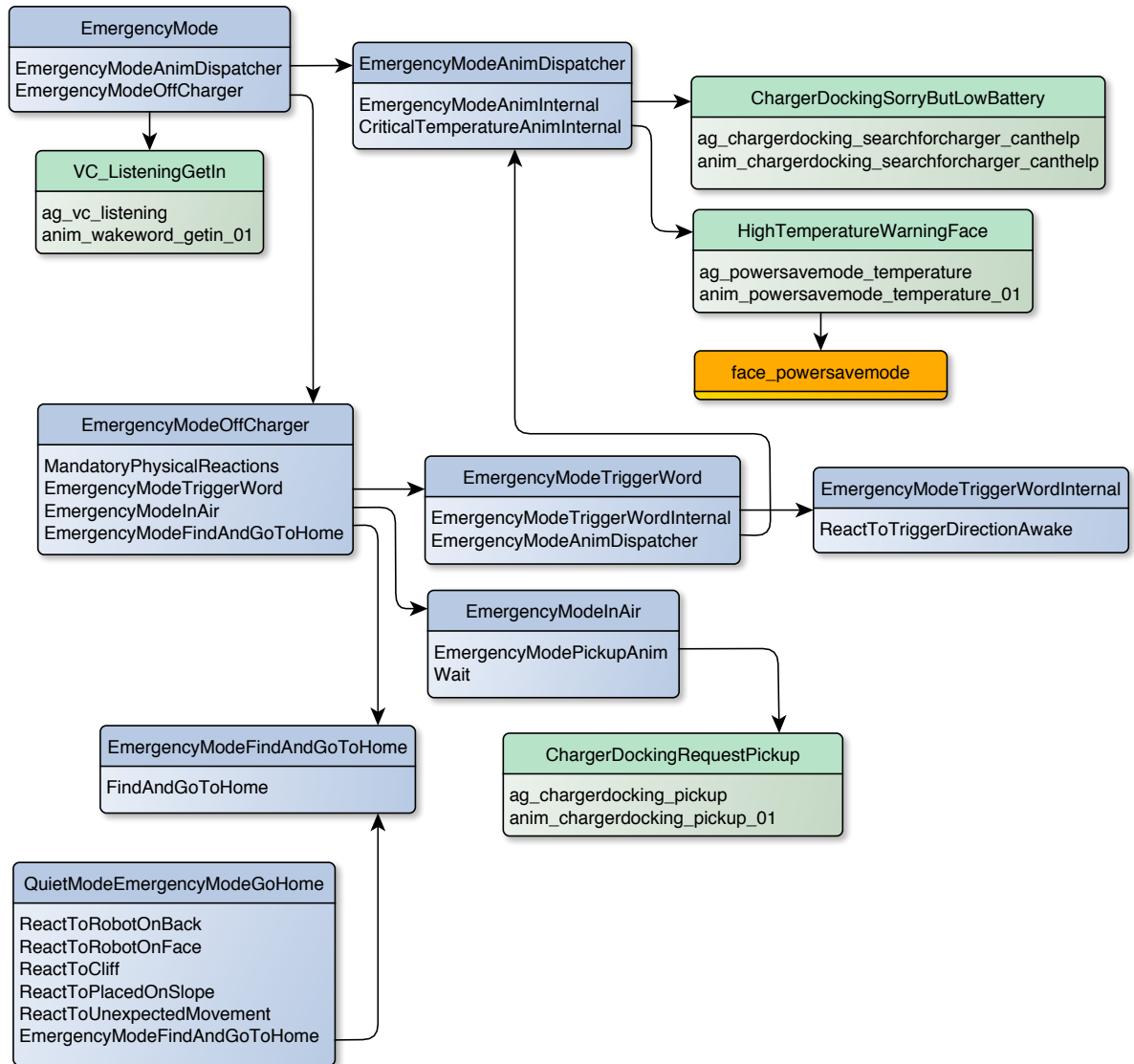
Figure: The emergency behavior tree

The behavior file is located at:

```
behaviors/victorBehaviorTree/emergencyMode/emergencyMode.json
```

Animation feedback

The behavior coordinates with working with other social interactions, albeit in a restricted manner. If the trigger word is heard, it doesn't stream the audio to the voice server. Instead it:

1. Plays the *VC_ListeningGetIn* animation, then

2. Plays the *StreamingDisabledButWithLight* animation to indicate that streaming is disabled, and filly

3. Sends the *Play__Robot_Vic_Sfx__Wake_Word_On* audio event to play a feedback sound.

4. It also links with the following animation to show the reason why

Emergency mode uses the *EmergencyModeAnimDispatcher* (class *DispatcherQueue*) behavior to play different animations based on the the emergency condition:

- If the battery is low, it will trigger the *ChargerDockingSorryButLowBattery* animation.

- If the temperature is high, or too hot charge, it will trigger the *HighTemperatureWarningFace* animation; this will play the *face_powersavemode* sprite sequence.

The behavior file is located at:

```
behaviors/victorBehaviorTree/emergencyMode/emergencyModeAnimDispatcher.json
```

If Vector is in picked up, or otherwise off his treads (but not being held), he plays the *ChargerDockingRequestPickup* animation. This is done in the *EmergencyModeInAir* behavior. The behavior file is located at:

```
behaviors/victorBehaviorTree/emergencyMode/emergencyModeInAir.json
```

Returning to the charging dock

There are a couple of behaviors that try to cause Vector to drive back to the charging dock. These run a subset of navigation and driving related behaviors to drive to the charging dock. The first is the *EmergencyModeOffCharger* (class *DispatcherStrictPriority*). The behavior file is located at:

```
behaviors/victorBehaviorTree/emergencyMode/emergencyModeOffCharger.json
```

The second is the *QuietModeEmergencyModeGoHome* (class *DispatcherStrictPriority*), which would be invoked while in quiet mode. *Commentary: This behavior appears like it should not run; the same conditions that would trigger it would also trigger the much higher priority EmergencyMode behavior. This behavior should be checked out and considered for removal.*

13.15.4 Change history synopsis

Date	Change
2020-11-30	Created
2020-12-1	Quiet mode's emergency mode, and trigger word animation

13.16 Quiet mode behaviors

Summary: The behaviors related to turning off, handling very low-battery, sleeping, and other battery-related items. (These are self-maintenance behaviors)

The quiet mode is when Vector's has been asked to be silent, either nicely ("be quiet") or abusively ("shut up").

13.16.1 Quiet Mode

The *QuietMode* behavior is when Vector's has been asked to be silent, nicely ("be quiet") using the `imperative_quiet` user intent.
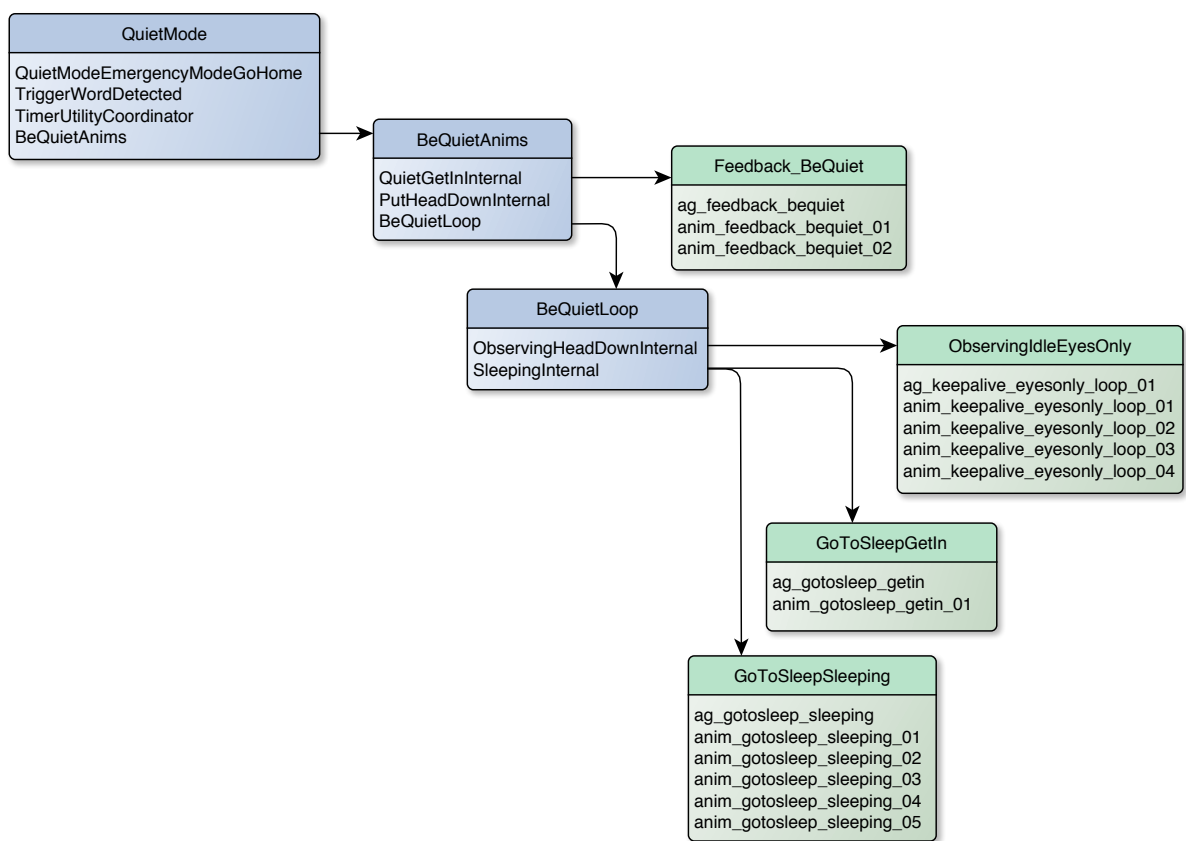
Figure: Quiet mode behavior tree

It *QuietModeEmergencyModeGoHome* see power management for a description of emergency mode.

The *BeQuietAnims* behavior is used to trigger the *Feedback_BeQuiet* animation, and lowers the Vector's drowsy head, using the *PutHeadDownInternal* behavior.

Thereafter the *BeQuietLoop* is used play one of three animations:

- The *ObservingIdleEyesOnly* animation is played while Vector sits quietly and looks around.

- The *GoToSleepGetIn* animation is played when Vector goes to sleep,

- The *GoToSleepSleeping* animation is played while Vector sleeps.

13.16.2 ShutUp mode

The *ShutUpMode* behavior is variation of quiet mode, used when Vector's has been asked to "shut up" (the `imperative_shutup` user intent).
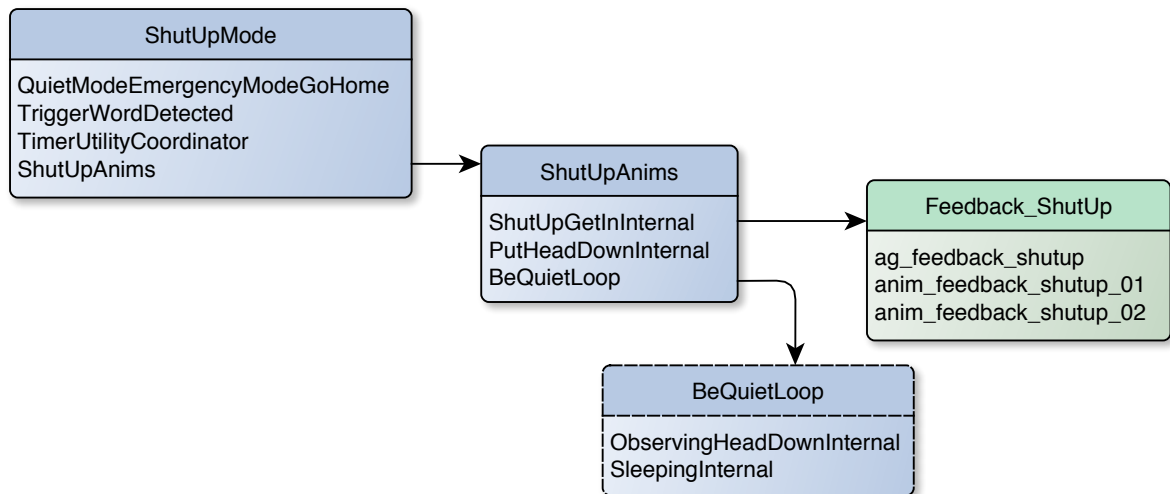
Figure: Shut up mode behavior tree

The main difference is the intent that triggers is, and the animation response. Thereafter, this mode reused the same *BeQuietLoop* used by the quiet mode.

13.16.3 Change history synopsis

Date	Change
2020-12-1	Created

13.17 Software Classes

Vector's software has a lot of modules -- I'm assuming these are C++ classes. They are not all annotated or understood. Here are some that we've spotted:

Module	Description
AIComponent	
AIWhiteboard	
AccountSettingsManager	
ActionList	
ActionQueue	
Actions	
ActiveFeatureComponent	
AddActiveObject	
AdvertisementService	
AkAlsaSink	
Alexa	
AlexaAudioInput	
AlexaClient	
AlexaComponent	
AlexaImpl	
AlexaMediaPlayer	
AlexaObserver	
AlexaPlaybackRecognizerComponent	
AlignWithObjectAction	
AnimComms	
AnimContext	
AnimEngine	
AnimProcessMessages	
Animation	
AnimationAudioClient	
AnimationComponent	
AnimationGroup	
AnimationGroupContainer	
AnimationGroupEntry	
AnimationStreamer	
Animations	
AnkiLab	
AppCubeConnectionSubscriber	
Array2d	
AttentionTransferComponent	

Module	Description
AudienceTags	
Audio	
AudioBehaviorStackListener	
AudioEngineController	
AudioEventGroupRef	
AudioMultipleFileLocation	
AudioMultiplexer	
AudioMuxClient	
AudioScene	
AudioSceneEvent	
AudioSceneParameter	
AudioSceneStateGroup	
AudioWaveFileReader	
BEIConditionFactory	
BEIConditionMessageHelper	
BackpackLightAnimationContainer	
BackpackLightComponent	
BackpackLightsKeyFrame	
BackupOntoChargerAction	
Battery	
BatteryComponent	
BeatDetector	
BeatDetectorComponent	
Behavior	
BehaviorAcknowledgeFace	
BehaviorAcknowledgeObject	
BehaviorAlexa	
BehaviorAnimSequenceWithObject	
BehaviorAskForHelp	
BehaviorBlackJack	
BehaviorBumpObject	
BehaviorClearChargerArea	
BehaviorComponent	
BehaviorConfirmObject	
BehaviorConnectToCube	

Module	Description
BehaviorCoordinateInHabitat	
BehaviorCoordinateWeather	
BehaviorCountingAnimation	
BehaviorDanceToTheBeat	
BehaviorDanceToTheBeatCoordinator	
BehaviorDevCubeSpinnerConsole	
BehaviorDevSquawkBoxTest	
BehaviorDevTurnInPlaceTest	
BehaviorDispatchAfterShake	
BehaviorDispatcherPassThrough	
BehaviorDispatcherQueue	
BehaviorDispatcherRandom	
BehaviorDispatcherStrictPriorityWithCooldown	
BehaviorDisplayWallTime	
BehaviorDisplayWeather	
BehaviorDockingTest	
BehaviorDockingTestSimple	
BehaviorDriveOffCharger	
BehaviorEnrollFace	
BehaviorExploring	
BehaviorExploringExamineObstacle	
BehaviorEyeColorVoiceCommand	
BehaviorFactoryCentroidExtractor	
BehaviorFetchCube	
BehaviorFindCube	
BehaviorFindFaceAndThen	
BehaviorFindHome	
BehaviorFistBump	
BehaviorGoHome	
BehaviorGreetAfterLongTime	
BehaviorHowOldAreYou	
BehaviorInspectCube	
BehaviorInteractWithFaces	
BehaviorKeepaway	
BehaviorKnowledgeGraphQuestionc	

Module	Description
BehaviorLiftLoadTest	
BehaviorLookForFaceAndCube	
BehaviorObservingLookAtFaces	
BehaviorObservingWithoutTurn	
BehaviorOnboardingCoordinator	
BehaviorPRDemo	
BehaviorPlaceCubeByCharger	
BehaviorPlaypenCameraCalibration	
BehaviorPlaypenDistanceSensor	
BehaviorPlaypenDriftCheck	
BehaviorPlaypenEndChecks	
BehaviorPlaypenPickupCube	
BehaviorPlaypenTest	
BehaviorPopAWheelie	
BehaviorPounceOnMotion	
BehaviorPoweringRobotOff	
BehaviorPromptUserForVoiceCommand	
BehaviorPuzzleMaze	
BehaviorQuietModeCoordinator	
BehaviorReactToBody	
BehaviorReactToCliff	
BehaviorReactToCubeTap	
BehaviorReactToDarkness	
BehaviorReactToHand	
BehaviorReactToMicDirection	
BehaviorReactToMotion	
BehaviorReactToMotorCalibration	
BehaviorReactToPlacedOnSlope	
BehaviorReactToRobotOnBack	
BehaviorReactToRobotOnFace	
BehaviorReactToTouchPetting	
BehaviorReactToUncalibratedHeadAndLift	
BehaviorReactToUnexpectedMovement	
BehaviorReactToVoiceCommand	
BehaviorRequestToGoHome	

Module	Description
BehaviorResetState	
BehaviorRespondToRenameFace	
BehaviorRobustChargerObservation	
BehaviorSDKInterface	
BehaviorSDKLock	
BehaviorSayName	
BehaviorSelfTest	
BehaviorSelfTestDockWithCharger	
BehaviorSelfTestDriftCheck	
BehaviorSelfTestLookAtCharger	
BehaviorSleepCycle	
BehaviorSystem	
BehaviorSystemManager	
BehaviorTakeAPhotoCoordinator	
BehaviorTextToSpeechLoop	
BehaviorTrackCube	
BehaviorTrackFace	
BehaviorUserDefinedBehaviorTreeRouter	
BehaviorUserDefinedBehaviorTreeSelector	
BehaviorVolume	
Behaviors	
BehaviorsBootLoader	
BlackJackGame	
BlackJackSimulation	
BlackJackVisualizer	
Block	
BlockPool	
BlockTapFilterComponent	
BlockWorld	
BodyMotionKeyFrame	
CalculateExperimentHashBucket	
Camera	
CameraCalibrator	
CameraParamsController	
CannedAnimationContainer	

Module	Description
CannedAnimationLoader	
CardSimulation	
CarryingComponent	
ChannelFilter	
CladEnumToStringMap	
CliffAlignToWhiteAction	
CliffSensor	
CliffSensorComponent	
ColorRGBA	
CompositeImage	
CompositeImageLayer	
CompoundActionParallel	
CompoundActionSequential	
ComputePlacementApproachAngle	
ComputePreActionPoseDistThreshold	
ConditionCompound	
ConditionEngineErrorCodeReceived	
ConditionIlluminationDetected	
ConditionMotionDetected	
ConfirmHabitat	
ConnectionFlow	
Console	
ConsoleSystem	
Context	
ContinuityComponent	
CoreTech	
CozmoAPI	
CozmoAnimMain	
CozmoAudioController	
CozmoEngine	
CozmoGameImpl	
CropScheduler	
CubeAccelComponent	
CubeBatteryComponent	
CubeComms	

Module	Description
CubeCommsComponent	
CubeConnectionCoordinator	
CubeInteractionTracker	
CubeLightAnimationContainer	
CubeLightAnimationHelpers	
CubeLightComponent	
CubeLightController	
CubeSpinnerGame	
CustomObject	
DTRawPixelsClassifier	
DanceAnimMetadata	
DancePhrase	
DanceSession	
DasToSdkHandler	
DasToSdkManager	
DataPlatform	
Demo	
DevEventSequenceCapture	
DoleAvailableAnimations	
DriveAndFlipBlockAction	
DriveStraightAction	
DriveToActions	
DriveToFlipBlockPoseAction	
DriveToObjectAction	
DriveToPlaceCarriedObjectAction	
DriveToPlaceRelObjectAction	
DriveToPoseAction	
DrivingAnimationHandler	
EmotionAffector	
EmotionEvent	
EmotionEventMapper	
EmotionScorer	
EngineRobotAudioClient	
EngineRobotAudioInput	
EnrolledFaceEntry	

Module	Description
EraseAllFaces	
Error	
EventKeyFrame	
Expected	
Experiment	
FaceDisplay	
FaceInfoScreenManager	
FaceLayerManager	
FacePlantAction	
FaceRecognizer	
FaceTrackerImpl	
FaceWorld	
Factory	
FactoryTestLogger	
FeatureGate	
FileTransfer	
FindFaces	
FlipBlockAction	
FormatBytesAsHex	
GMMRawPixelsClassifier	
GetAnimationName	
GetBroadcastAddressFromIfAddr	
GetIPv6LinkLocalAddress	
GetLocalIpAddress	
GetLocalIpAddressFromIfAddr	
GetLocalIpv6LinkLocalAddress	
GetMaxOffsetObjectStillVisible	
GetNextAlbumEntryToUse	
GetNextPacketFromEngine	
GetNextPacketFromRobot	
GetRecognitionData	
GetSerializedAlbum	
GoogleBreakpad	
GraphEvaluator2d	
GroundPlaneClassifier	

Module	Description
HabitatDetectorComponent	
HandleAnimationEvent	
HandleMotorAutoEnabled	
HandleMotorCalibration	
HashStringTable	
HeldInPalmTracker	
HueSatWrapper	
IAction	
IActionRunner	
IBEICondition	
IBehavior	
IBehaviorPlaypen	
IBehaviorSelfTest	
ICompoundAction	
IConditionUserIntent	
ICozmoBehavior	
IDockAction	
IDriveToInteractWithObject	
IFormattedLoggerProvider	
IKeyFrame	
INeuralNetMain	
INeuralNetModel	
IPathPlanner	
IPv6	
IScoredBehavior	
ISensorComponent	
ITrackAction	
ITrackLayerManager	
IVisuallyVerifyAction	
Id	
IdCount	
Image	
ImageBase	
ImageBrightnessHistogram	
ImageCache	

Module	Description
ImageSaver	
ImageSensor	
ImuComponent	
InternalStatesBehavior	
Interruption	
IsCloseEnoughToPreActionPose	
IsExternalSdkConnection	
JdocsManager	
JsonTools	
KnowledgeGraph	
KnownMarker	
LOG	
LaserPointDetector	
LinearClassifier	
LocalUdpSocketComms	
Locale	
LocaleComponent	
Location	
Looking	
LoopBoundOverflow	
MapComponent	
Marker	
MarkerDetector	
Mask	
MenuConsoleChannel	
Message	
MicComponent	
MicDataInfo	
MicDataProcessor	
MicDataSystem	
MicDirectionHistory	
MicTriggerConfig	
Microphones	
MinimalAnglePlanner	
Mood	

Module	Description
MoodDecayEvaulator	
MoodManager	
MoodScorer	
MountChargerAction	
MoveHeadToAngleAction	
MoveLiftToAngleAction	
MoveLiftToHeightAction	
Movement	
MovementComponent	
MultiClientComms	
MusicConductor	
NVStorage	
NVStorageComponent	
NamedColors	
NativeAnkiUtilConsoleCallFunction	
NativeAnkiUtilConsoleIsDefaultValue	
NativeAnkiUtilConsoleResetValueToDefault	
NativeAnkiUtilConsoleSetValueWithString	
NativeAnkiUtilConsoleToggleValue	
NetEmulatorUDPSocket	
Network	
NeuralNetModel	
NeuralNetParams	
NeuralNetRunner	
NeuralNets	
ObjectInteractionInfoCache	
ObservableObject	
ObservableObjectLibrary	
ObservableObjectsLibrary	
OffboardModel	
OverheadMap	
PackMaskedId	
PackMaskedIds	
PanAndTiltAction	
ParamTraits	

Module	Description
PathComponent	
PathDolerOuter	
PerfMetric	
PetTracker	
PhotographyManager	
PickupObjectAction	
PlaceObjectOnGroundAction	
PlaceRelObjectAction	
Planner	
PlayAnimationAction	
PopAWheelieAction	
Pose3d	
PoseBase	
PowerStateManager	
PowerStates	
PreActionPose	
ProceduralFace	
ProcessRegistrationMsg	
ProxSensorComponent	
PublicStateBroadcast	
PublicStateBroadcaster	
Puzzle	
QuadTree	
QuadTreeNode	
QuestEngine	
Ran	
RandomGenerator	
RandomVectorSampler	
ReactionStrategyFacePositionUpdate	
ReadBMP	
RecentOccurrenceTracker	
RecognizeFace	
Rectangle	
RegisterNewUser	
RejectIfChargerOutOfView	

Module	Description
RejectIfWouldCrossCliff	
ReliableConnection	
ReliableTransport	
RemoveUser	
RequestAvailableAnimations	
ReselectingLoopAnimationAction	
RetryWrapperAction	
Robot	
RobotActionParams	
RobotAudioKeyFrame	
RobotConnectionManager	
RobotDataLoader	
RobotEventHandler	
RobotHealthReporter	
RobotImplMessaging	
RobotInitialConnection	
RobotManager	
RobotState	
RobotStateHistory	
RobotStats	
RobotStatsTracker	
RollObjectAction	
RollingShutterCorrector	
RotationMatrixBase	
RotationVector3d	
SDKComponent	
SayNameProbabilityTable	
SayTextAction	
SdkAudioComponent	
SdkComponent	
SdkLock	
SendPacketToRobot	
SetSerializedAlbum	
SetSockOpt	
SetThreadPriority	

Module	Description
Setting	
SettingsCommManager	
SettingsManager	
ShowAudioStreamStateManager	
Shutting	
SimpleMoodScorer	
SleepTracker	
SoundbankBundleInfo	
SoundbankLoader	
SpeechRecognizer	
SpeechRecognizerPryonLite	
SpeechRecognizerSystem	
SpeechRecognizerTHF	
SpeedChooser	
SpriteCache	
SpriteEntry	
SpritePathMap	
SpriteSequence	
SpriteSequenceContainer	
SpriteSequenceKeyFrame	
SpriteSequenceLoader	
SpriteWrapper	
StandardWaveDataContainer	
Starting	
StaticMoodData	
Stopping	
StreamingAnimationModifier	
StreamingWaveDataInstance	
TFLiteLogReporter	
TFLiteModel	
TId	
TextToSpeech	
TextToSpeechComponent	
TextToSpeechCoordinator	
TextToSpeechProvider	

Module	Description
TextToSpeechProviderImpl	
TimerUtility	
TouchBaselineCalibrator	
TouchSensor	
TouchSensorComponent	
Track	
TrackFaceAction	
TrackGroundPointAction	
TrackLayerManager	
TrackObjectAction	
TrackPetFaceAction	
TrackpetFaceAction	
TransportAddress	
TriggerAnimationAction	
TriggerEmotionEvent	
TurnInPlaceAction	
TurnTowardsFaceAction	
TurnTowardsObjectAction	
TurnTowardsPoseAction	
UDPTransport	
UdpSocketComms	
UiComms	
UiMessageHandler	
Undistorter	
Unfiltered	
UnpackMaskedIds	
Update	
UpdateExistingAlbumEntry	
UpdateRecognitionData	
UseLoadedAlbumAndEnrollData	
UseLoadedAlbumAndEnrollmentData	
User	
UserDefinedBehaviorTreeComponent	
UserEntitlementsManager	
UserIntentComponent	

Module	Description
UserIntentMap	
Util	
VariableSnapshotComponent	
VerifyDecayGraph	
VisionComponent	
VisionModeSchedule	
VisionProcessingResult	
VisionScheduleMediator	
VisionSystem	
VisuallyVerifyObjectAction	
VizManager	
VoiceMessage	
VoiceMessageSystem	
WeatherIntentParser	
WwiseComponent	
XYPlanner	

13.18 Sound Banks

13.18.1 Victor SFX sound bank

Most of the Vector's sound effects are in the "Victor_SFX" sound bank.
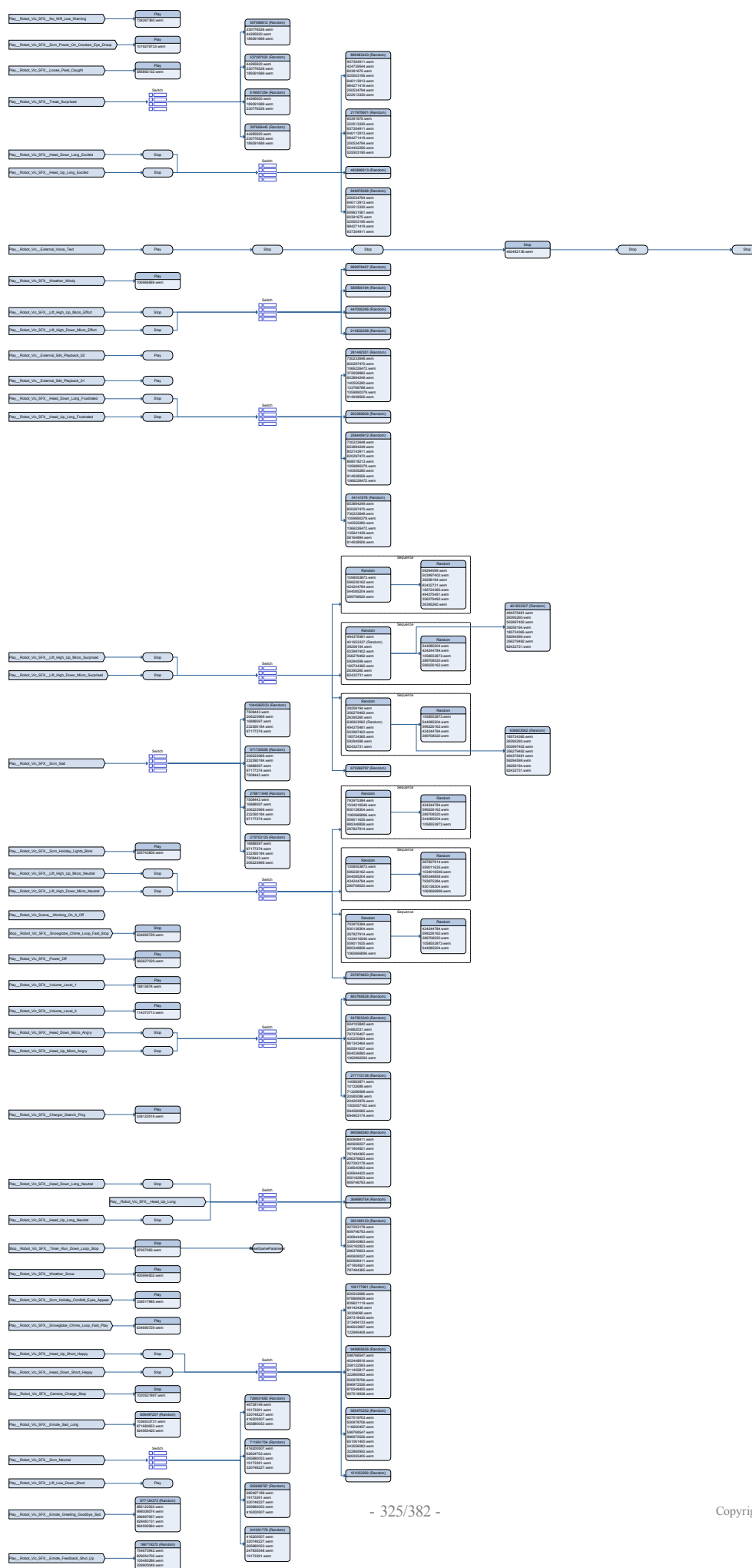
Figure: Victor SFX sound bank

A PDF of the soundbank diagram is also available as well.

13.18.2 Change history synopsis

Date	Change
2021-3-4	Created

Figure: Victor SFX sound bank

13.19 Sound Events

ID	Trigger	Description
100246366	Play__Robot_Vic_Sfx__Emote_Curious_Short_Stim	
1003390754	Play__Robot_Vic_SFX__Lift_High_Up_Short_Sad	
1019978791	Stop__Robot_Vic_Sfx__Shaking_Level_1_Stop	
1028223059	Play__Robot_Vic_Alexa__Sfx_Sml_Ui_Wakesound_Touch	
1034417959	Play__Dev_Robot__Tone_1k_5sec	
1040115983	Play__Robot_Vic_Alexa__External_Voice_Pause	
1078700783	Play__Robot_Vic_Sfx__Touch_React	
1102367108	Play__Robot_Vic_Sfx__Scrn_Curious_Long	
1110912082	Play__Robot_Vic_Sfx__Pet_Attention_Tone_Gen	
1119216913	Stop__Robot_Vic_Sfx__Snowglobe_Wind_Loop_Slow_Stop	
1129945440	Play__Robot_Vic_SFX__Head_Up_Short_Frustrated	
1135332675	Play__Robot_Vic_Sfx__Attention_Device_Loop_Play	
1140994213	Play__Robot_Vic_SFX__Lift_High_Up_Long_Neutral	
1150286419	Play__Robot_Vic_Alexa__Sfx_Sml_State_Privacy_Mode_Off	
1165565764	Play__Robot_Vic_SFX__Head_Up_Short_Effort	
1170663689	Play__Robot_Vic_SFX__Lift_High_Down_Short_Sad	
1175231617	Play_Robot_Vic_SFX_Down_Long_Neutral_Dev_01	
1181334695	Play__Robot_Vic_Sfx__Emote_Feedback_Shut_Up_Come_At_Me	
1193798168	Play__Robot_Vic_Sfx__Tread_Sad_Long	
1195953423	Play__Robot_Vic_SFX__Head_Down_Short_Excited	
1197687125	Play__Robot_Vic_Sfx__Lift_High_Up_Long	
1201929830	Play__Robot_Vic_Sfx__Lift_High_Dancing_Big	
1211422730	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Effort	
1216170354	Play__Robot_Sfx__Fist_Bump	
1234596125	Play__Robot_Vic_Sfx__Emote_Shaking_Level_3	
123523358	Play__Robot_Vic_Sfx__Tread_Happy_Long	
1241549428	Play__Robot_Vic_Sfx__Wake_Word_Fail	
1242048048	Play__Robot_Vic_SFX__Head_Down_Long_Neutral	
1248646962	Play__Robot_Vic_Alexa__Avs_System_Prompt_Error_Offline_Not_Connected_To_Service_Else	
1249715754	Play_Robot_Vic_SFX_Head_Up_Short_Neutral_Dev_02	
1251262896	Play__Robot_Vic_Sfx__Concentrate_Loop_Play	
1268088608	Play__Robot_Vic_SFX__Head_Up_Micro_Sad	
1284632326	Play__Robot_Vic_Sfx__Timer_Cancel	
1296983121	Play__Robot_Vic_SFX__Lift_High_Down_Short_Frustrated	
1304467293	Play__Robot_Vic_Sfx__Emote_Greeting_Hello	

ID	Trigger	Description
1312053763	Play__Robot_Vic_Sfx__Camera_Charge_Play	
1322012872	Play__Robot_Vic_Sfx__Lift_Up_Short	
1328248747	Play__Robot_Vic_Sfx__Fist_Bump	
1333208724	Stop__Robot_Vic_Sfx__Snowglobe_All_Loop_Stop	
1345769624	Play__Robot_Vic_Alexa__Sfx_Sml_Alerts_Notification_01	
1345769626	Play__Robot_Vic_Alexa__Sfx_Sml_Alerts_Notification_03	
1350709836	Play__Robot_Vic_Sfx__Scrn_Curious_Short	
1354807363	Play__Robot_Vic_Sfx__Cube_Search_Ping	
1357335898	Play__Robot_Vic_Sfx__Head_Loop_Play	
1363830477	Play__Robot_Vic_Sfx__Tread_Curious	
1370162399	Play__Robot_Vic_Alexa__Sfx_Sml_Ui_Wakesound	
1371648746	Play__Robot_Vic_SFX__Lift_High_Down_Short_Surprised	
1372621318	Play__Robot_Vic_Sfx__Holiday_Lights_Shake_Short	
1377916814	Play__Robot_Vic_SFX__Head_Up_Long_Happy	
1379596305	Play__Robot_Vic_Sfx__Timer_Countdown	
1380232790	Play__Robot_Vic_Sfx__Scrn_Sad_Long	
1393504716	Play__Robot_Vic_SFX__Head_Up_Short_Curious	
139550890	Play__Robot_Vic_Sfx__Timer_Alarm_Start	
1399502169	Play__Robot_Vic_Sfx__Onboarding_Power_On_Eyes_Open	
1408631403	Play__Robot_Vic_Sfx__Knowledge_Graph_Listening_Loop_Play	
1430272919	Play__Robot_Vic_Sfx__Holiday_Fireworks_Explode	
1432672265	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Happy	
1439924065	Play__Robot_Vic_Alexa__External_Voice_Play	
146478435	Play__Robot_Vic_Sfx__Scrn_Procedural_Shift	
1465647653	Play__Robot_Vic_Sfx__Lift_Down_Long	
1471825745	Play__Dev_Robot__Tone_440Hz_5sec	
1476560429	Play__Robot_Vic_Sfx__Lift_Low_Up_Short	
1485491007	Stop__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Medium_Stop	
1498070908	Play_Robot_Vic_SFX_Head_Up_Long_Excited_Dev	
1498450824	Play__Robot_Vic_Sfx__Lift_Up_Long	
1500480562	Play__Robot_Vic_SFX__Head_Down_Long_Angry	
1504093363	Play__Robot_Vic_Sfx__Lift_Low_Up_Long	
1506863732	Play__Robot_Vic_Sfx__Emote_Sad_Short_Stim	
1517760137	Play__Robot_Vic_Sfx__Wake_Word_Success_No_SFX	
1518602369	Play__Robot_Vic_Sfx__Weather_Thunder	

ID	Trigger	Description
152409978	Play__Robot_Vic_Sfx__Scan_Face_Fail	
1528281890	Play__Robot_Vic_Sfx__Scrn_Surprised	
1531247580	Play__Robot_Vic_Sfx__Distress_Alert	
1532745019	Play__Robot_Vic_Sfx__Wake_Word_On	
1536061703	Play__Dev_Robot__Tone_150_Frames_01	
1545176977	Play__Robot_Vic_Sfx__Power_Off_End	
1550510811	Play__Dev_Robot_Factory__Scan_Loop_Stop_Bell	
1554373331	Play__Robot_Vic_Sfx__Scan_Face_Success	
1559136524	Play__Robot_Vic_Sfx__Timer_Run_Down_Loop_Play	
1579329514	Play__Robot_Vic_Sfx__Emote_Weather_Cloudy	
158400132	Play__Robot_Vic_SFX__Head_Up_Short_Excited	
1585467997	Play__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Slow_Play	
1601827987	Play__Robot_Vic_Sfx__Wake_Word_On_No_Vo	
16087513	Play__Robot_Vic_SFX__Head_Down_Micro_Curious	
1632540473	Play__Robot_Vic_SFX__Lift_High_Down_Short_Effort	
1646082370	Play_Robot_Vic_SFX_Down_Long_Curious_Dev	
166531022	Play__Robot_Vic_Sfx__Lift_Low_Down_Long	
168487306	Play__Robot_Vic_Sfx__Emote_Curious_Long	
1686447722	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Happy	
1697864692	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Angry	
1711301107	Play__Robot_Vic_Sfx__Holiday_Lights_Build	
1712900907	Play__Robot_Vic_Sfx__Pet_Attention_Test_17k	
1717116727	Play__Robot_Vic_SFX__Lift_High_Down_Long_Happy	
1721825381	Play__Robot_Vic_Alexa__Avs_System_Prompt_Error_Cannot_Play_Song	
1723140054	Play__Robot_Vic_Sfx__Shaking_Level_2_Play	
1729678520	Play__Robot_Vic_Sfx__Pet_Attention_Test_16k	
1731537112	Stop__Robot_Vic_Sfx__Concentrate_Loop_Stop	
1735763806	Play__Dev_Robot__Fx_Test_Sequence	
1736370596	Play__Robot_Vic_Sfx__Camera_Focus_3	
1736370597	Play__Robot_Vic_Sfx__Camera_Focus_2	
1736370598	Play__Robot_Vic_Sfx__Camera_Focus_1	
1751049963	Play__Robot_Vic_Sfx__Sleeping_3	
1751049964	Play__Robot_Vic_Sfx__Sleeping_4	
1751049965	Play__Robot_Vic_Sfx__Sleeping_5	
1752486067	Play__Robot_Vic_Sfx__Scrn_Neutral_Short	

ID	Trigger	Description
1755199022	Stop__Robot_Vic_Sfx__Snowglobe_Wind_Loop_Fast_Stop	
1762574059	Play__Robot_Vic_Scene__Mov_RTPC_Reset	
1763233682	Play__Robot_Vic_Sfx__Pet_Attention_Test_14k	
1768349824	Play__Robot_Vic_Alexa__External_Alerts_Play	
1784237387	Play_Robot_Vic_SFX_Head_Up_Short_Curious_Dev_02	
1786707793	Play__Robot_Vic_Sfx__Emote_Greeting_Good_Morning	
1789553612	Play__Robot_Vic_SFX__Lift_High_Down_Long_Neutral	
1795232951	Stop__Robot_Vic_Sfx__Low_Light_Charging_Loop_Stop	
1818400219	Play__Robot_Vic_Alexa__Sfx_Sml_System_Alerts_Melodic_01_Short	
1824844463	Play__Robot_Vic_Alexa__Sfx_Sml_State_Bluetooth_Connected	
1826579815	Play__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Medium_Play	
1827504806	Play__Robot_Vic_SFX__Lift_High_Down_Long_Angry	
1842808354	Play__Robot_Vic_Sfx__Tread_Curious_Long	
1848638287	Play__Robot_Vic_Sfx__Timer_End	
1851080792	Play__Robot_Vic_SFX__Head_Up_Long_Sad	
1868738390	Play_Robot_Vic_SFX_Head_Up_Short_Excited_Dev	
1871298466	Play__Robot_Vic_Sfx__Blackjack_Lose	
1883414229	Play__Robot_Vic_Sfx__Scrn_Neutral_Long	
1893130014	Play__Robot_Vic_SFX__Lift_High_Up_Long_Curious	
1900064481	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_To_Eyes_2	
1900064482	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_To_Eyes_1	
190587959	Play__Robot_Vic_Sfx__How_Old_Fast	
1909472813	Play__Robot_Vic_Sfx__Tread_Angry_Long	
1937738898	Play__Robot_Vic_SFX__Head_Down_Short_Angry	
1946288595	Play__Robot_Vic_SFX__Head_Down_Micro_Frustrated	
1946291652	Play__Robot_Vic_SFX__Lift_High_Down_Short_Neutral	
1959779482	Play__Robot_Vic_SFX__Head_Up_Micro_Frustrated	
1964410187	Play__Robot_Vic_Sfx__Scrn_Angry_Long	
1964565174	Play__Robot_Vic_Sfx__Pet_Attention_Test_18k	
1984864293	Play__Robot_Vic_SFX__Head_Down_Long_Sad	
1987410443	Play__Robot_Vic_Scene__Quiet_On	
1997461118	Play__Robot_Vic_SFX__Lift_High_Down_Short_Angry	
2023173047	Play__Robot_Vic_Scene__Low_Power_Mode_Off	
2025442913	Play_Robot_Vic_SFX_Head_Up_Long_Excited_Dev_02	
2026706803	Play__Robot_Vic_SFX__Head_Down_Micro_Sad	

ID	Trigger	Description
2028807349	Play__Robot_Vic_Sfx__Snowglobe_Wind_Loop_Slow_Play	
203477807	Play__Robot_Vic_Sfx__Scan_Loop_Play	
2054107014	Stop__Robot_Vic_Sfx__Shaking_Level_2_Stop	
205416233	Play__Robot_Vic_Sfx__No_Wifi_Icon_Glitch	
2057988846	Play__Robot_Vic_Sfx__Weather_Sunny	
2071056089	Play__Robot_Vic_Sfx__Purr_Increase_Level	
207364355	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_Eyes_Appear_Single	
2075663789	Play__Robot_Vic_Sfx__Knowledge_Graph_Loop_Play	
2075720120	Stop__Robot_Vic_Sfx__Knowledge_Graph_Searching_Loop_Stop	
2091194049	Play__Robot_Vic_Sfx__Scrn_Procedural_Squint	
2091613738	Play__Robot_Vic_Sfx__Holiday_Lights_Appear	
2099947214	Play__Robot_Vic_Sfx__Emote_Greeting_Goodnight_Sleep	
2112819935	Play__Robot_Vic_Scene__Quiet_Off	
211516390	Play__Dev_Robot__Tone_10_Frames_01	
212200851	Play__Robot_Vic_SFX__Head_Up_Long_Surprised	
2136765628	Play__Robot_Vic_Sfx__Emote_Feedback_Apology	
2172739081	Play__Robot_Vic_Alexa__Avs_System_Prompt_Error_Offline_Not_Registered	
2179926555	Play__Robot_Vic_Sfx__Power_On	
219117156	Play__Robot_Vic_Alexa__Sfx_Sml_System_Alerts_Melodic_01	
219117159	Play__Robot_Vic_Alexa__Sfx_Sml_System_Alerts_Melodic_02	
2198114280	Play__Robot_Vic_Sfx__Timer_Set	
2199944662	Play__Robot_Vic_Sfx__Emote_Greeting_Goodnight	
2208543816	Play__Robot_Vic_Sfx__Scrn_Happy_Short	
2211318753	Play__Robot_Vic_Sfx__Knowledge_Graph_Searching_Loop_End	
2211807329	Play__Robot_Vic_SFX__Lift_High_Down_Long_Effort	
2241028412	Play__Robot_Vic_Sfx__Cant_Do	
2242865123	Play__Robot_Vic_Sfx__Scrn_Holiday_Confetti_Eyes_Disappear	
2263664842	Play_Robot_Vic_SFX_Head_Up_Short_Curious_Dev	
2268001086	Play__Dev_Squawk__No_Wifi	
2268335938	Play__Robot_Vic_Sfx__Emote_Happy_Short_Stim	
2274456998	Play__Robot_Vic_Sfx__Emote_Happy_Long	
2276351468	Play__Robot_Vic_Sfx__Holiday_Lights_Build_Short	
2315630444	Play__Robot_Vic_Sfx__Timer_Beep	
2316978067	Play__Robot_Vic_SFX__Head_Down_Long_Happy	
2320289125	Play__Dev_Squawk__Head_Short_Curious	

ID	Trigger	Description
2345965616	Play__Robot_Vic_Sfx__Eye_Color_Change	
2347545436	Play__Robot_Vic_Alexa__Avs_System_Prompt_Error_Offline_Lost_Connection	
2349381079	Play__Robot_Vic_Sfx__Scrn_Happy	
2363892348	Play__Dev_Robot__Playpen_Freq_Sweep	
2369271905	Play__Robot_Vic_Sfx__Petting_Level_04	
2369271908	Play__Robot_Vic_Sfx__Petting_Level_01	
2369271910	Play__Robot_Vic_Sfx__Petting_Level_03	
2369271911	Play__Robot_Vic_Sfx__Petting_Level_02	
2385293445	Play__Robot_Vic_Sfx__Scrn_Angry_Short	
238773808	Play__Robot_Vic_Scene__Working_On_It_On	
2399841641	Play__Robot_Vic_SFX__Lift_High_Up_Short_Surprised	
2415524415	Play__Dev_Device__Tone_Generator	
2420361307	Stop__Robot_Vic_Sfx__Planning_Loop_Stop	
2429853747	Play__Robot_Vic_Sfx__Lift_High_Up_Short	
2441768920	Play__Robot_Vic_Alexa__Sfx_Sml_System_Alerts_Melodic_02_Short	
2451007033	Play__Robot_Vic_Sfx__Snowglobe_Chime_Shakeoff_End	
2451569527	Play__Robot_Vic_Sfx__Head_Up_Short	
2452862642	Stop__Robot_Vic_Sfx__Wake_Word_Success_Processing_Stop	
2459769158	Play__Robot_Vic_Sfx__Scrn_Angry	
2460233869	Play__Robot_Vic_Sfx__Emote_Feedback_Be_Quiet	
2468739505	Stop__Robot_Vic_Sfx__Working_Loop_Stop	
2479504035	Stop__Robot_Vic_Sfx__Attention_Device_Loop_Stop	
2483759946	Play__Robot_Vic_Sfx__Emote_Happy_Short	
2500637363	Play__Robot_Vic_Sfx__Emote_Weather_Sunny	
2505487592	Stop__Robot_Vic_Sfx__Lift_High_Mood	
2507176587	Play__Robot_Vic_SFX__Head_Down_Micro_Effort	
2510834985	Play__Robot_Vic_Sfx__No_Wifi	
2518881836	Play__Robot_Vic_SFX__Lift_High_Up_Short_Happy	
252118152	Play__Robot_Vic_Sfx__Speaker_Test_03	
252118153	Play__Robot_Vic_Sfx__Speaker_Test_02	
252118154	Play__Robot_Vic_Sfx__Speaker_Test_01	
252118157	Play__Robot_Vic_Sfx__Speaker_Test_06	
252118158	Play__Robot_Vic_Sfx__Speaker_Test_05	
252118159	Play__Robot_Vic_Sfx__Speaker_Test_04	
253187573	Play__Dev_Robot__Freq_Sweep_5sec	

ID	Trigger	Description
2539447680	Play__Dev_Robot__External_Source	
2545453695	Play__Robot_Vic_SFX__Lift_High_Up_Long_Angry	
2548052428	Play__Robot_Vic_Sfx__Lift_High_Down_Long	
2558890771	Play__Robot_Vic_Sfx__Scrn_Curious	
2568320945	Play__Robot_Vic_SFX__Lift_High_Down_Long_Sad	
258785237	Play__Robot_Vic_Sfx__Holiday_Fireworks_Shoot	
2602622212	Play__Robot_Vic_Sfx__Power_On_Short	
2606889820	Play__Robot_Vic_Sfx__Holiday_Lights_Slide	
2607206181	Play__Robot_Vic_Sfx__Emote_Weather_Windy	
2609931602	Play__Robot_Vic_Sfx__Timer_Alarm_Build	
2616099714	Play__Robot_Vic_SFX__Lift_High_Down_Long_Surprised	
2634648776	Play__Robot_Vic_SFX__Lift_High_Up_Short_Effort	
2640434952	Play__Robot_Vic_SFX__Lift_High_Up_Short_Curious	
2645885938	Play__Robot_Vic_Sfx__Volume_Level_4	
2645885939	Play__Robot_Vic_Sfx__Volume_Level_5	
2645885940	Play__Robot_Vic_Sfx__Volume_Level_2	
2645885941	Play__Robot_Vic_Sfx__Volume_Level_3	
2645885943	Play__Robot_Vic_Sfx__Volume_Level_1	
2655837071	Play__Robot_Vic_Sfx__Power_Off	
2655981186	Play_Robot_Vic_SFX_Head_Down_Short_Curious_Dev_02	
2666575538	Play_Robot_Vic_SFX_Up_Long_Neutral_Dev_01	
2675710110	Stop__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Fast_Stop	
2676091470	Play__Robot_Vic_Scene__Working_On_It_Off	
2687281750	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Neutral	
2688639192	Play__Robot_Vic_Alexa__External_Alerts_Pause	
2692631652	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_Blink	
2699760977	Play__Robot_Vic_Sfx__Scrn_Sad	
2711176996	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Surprised	
2720120229	Play_Robot_Vic_SFX_Down_Long_Neutral_Dev	
2725306379	Play__Robot_Vic_Alexa__Sfx_Sml_Ui_Endpointing_Touch	
2727115439	Play__Robot_Vic_SFX__Head_Down_Long_Excited	
2727833730	Play__Robot_Vic_SFX__Head_Up_Long_Frustrated	
2749641932	Play__Robot_Vic__External_SDK_Playback_01	
2749641935	Play__Robot_Vic__External_SDK_Playback_02	
2751727815	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Effort	

ID	Trigger	Description
2756642540	Play__Robot_Vic_Sfx__Weather_Windy	
2774572774	Play__Robot_Vic__External_Voice_Text	
2774632011	Play_Robot_Vic_SFX_Up_Long_Curious_Dev_01	
278213126	Play__Robot_Vic_Sfx__Emote_Curious_Short	
2821049893	Play__Robot_Vic_Alexa__External_Notifications_Pause	
2834027966	Play__Robot_Vic_SFX__Head_Up_Long_Excited	
2834616264	Play__Dev_Robot__Tone_30_Frames_01	
2839960576	Play__Dev_Robot__Angry_Muttering	
284488955	Play__Robot_Vic_Sfx__Planning_Loop_Play	
286038569	Play__Robot_Vic_Sfx__Wake_Word_Success	
2869910227	Play__Robot_Vic_Alexa__External_Alerts_Resume	
2871689124	Play__Robot_Vic_Sfx__Tread_Surprised	
2894417895	Play__Robot_Vic_Sfx__Scrn_Power_On_Crooked_Eye_Droop	
2895150215	Play__Robot_Vic_Sfx__No_Wifi_Low_Warning	
2904355467	Play__Robot_Vic_SFX__Head_Up_Micro_Angry	
2908727346	Play__Robot_Vic_Sfx__Charger_Search_Ping	
2918178017	Play__Robot_Vic_SFX__Head_Up_Long_Neutral	
2920632256	Stop__Robot_Vic_Sfx__Timer_Run_Down_Loop_Stop	
293598009	Play__Robot_Vic_Sfx__Blackjack_Win	
2939047873	Play__Robot_Vic_SFX__Head_Up_Short_Angry	
2942381837	Play__Dev_Robot__Curious_Muttering	
2943031975	Play_Robot_Vic_SFX_Head_Up_Short_Excited_Dev_02	
2943586044	Play__Robot_Vic_Sfx__Alexa_Display_On	
2955397640	Play__Robot_Vic_Sfx__Scrn_Happy_Long	
2955541383	Play__Robot_Vic_Sfx__Scan_One_Shot	
2956401514	Play__Robot_Vic_SFX__Lift_High_Up_Long_Excited	
2960318302	Play__Robot_Vic_Alexa__Avs_System_Prompt_Error_Offline_Not_Connected_To_Internet	
2989671047	Play__Robot_Vic_Sfx__Behavior_Playback_Audio	
2992999317	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Neutral	
300132915	Play__Robot_Vic_Sfx__Weather_Rain	
3007251021	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Curious	
3013819119	Play__Robot_Vic_SFX__Lift_High_Down_Short_Happy	
3043844820	Play__Robot_Vic_Sfx__Camera_Flash	
3044908163	Play__Robot_Vic_Sfx__Knowledge_Graph_Listening_Loop_Start	
3051150798	Play__Robot_Vic_Sfx__How_Old_Slow	

ID	Trigger	Description
3056211831	Play__Robot_Vic_Sfx__Emote_Feedback_Shut_Up	
307430215	Play__Robot_Vic_Alexa__Sfx_Sml_Ui_Endpointing	
3093506817	Play__Robot_Vic_Sfx__Emote_Greeting_Goodbye_Sad	
3093985234	Play__Robot_Vic_Sfx__Lift_Low_Down_Short	
3096506751	Play__Robot_Vic_Alexa__External_Notifications_Play	
3132507500	Play__Robot_Vic_Sfx__Scrn_Neutral	
3149295678	Play__Robot_Vic_Alexa__External_Voice_Resume	
3161791584	Play_Robot_Vic_SFX_Down_Long_Curious_Dev_01	
3168489024	Play__Robot_Vic_Sfx__Emote_Sad_Long	
3168753800	Play__Robot_Vic__External_Voice_Message	
3169780311	Stop__Robot_Vic_Sfx__Camera_Charge_Stop	
3178694131	Play__Robot_Vic_SFX__Head_Down_Short_Happy	
3179862666	Play__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Fast_Play	
3194465711	Play__Robot_Vic_Sfx__Scrn_Holiday_Confetti_Eyes_Appear	
3217099864	Play__Robot_Vic_Sfx__Weather_Snow	
3223693447	Play__Robot_Vic_Sfx__Scrn_Surprised_Long	
3229565588	Stop__Robot_Vic_Sfx__How_Old_Loop_Stop	
3240385917	Play__Robot_Vic_SFX__Head_Down_Long_Effort	
3260182727	Play_Robot_Vic_SFX_Up_Long_Curious_Dev	
3271727348	Stop__Robot_Vic_Sfx__Head_Mood	
3277683584	Play__Robot_Vic_SFX__Head_Down_Micro_Surprised	
3279354518	Play__Robot_Vic_SFX__Head_Down_Long_Surprised	
3291976308	Play__Robot_Vic_Sfx__Gazing_Scan	
32945214	Play__Robot_Vic_Sfx__Emote_Happy_Long_Stim	
3298247409	Stop__Robot_Vic__External_SDK_Playback_02	
3298247410	Stop__Robot_Vic__External_SDK_Playback_01	
3300673888	Play__Robot_Vic_Sfx__Emote_Sad_Short	
3309105517	Play__Robot_Vic_SFX__Head_Down_Micro_Excited	
3317590423	Play__Robot_Vic_SFX__Lift_High_Up_Long_Surprised	
3323464084	Play__Robot_Vic_Sfx__Emote_Sad_Long_Stim	
3327136896	Play__Robot_Vic_Sfx__Low_Light_Charging_Start	
3327244935	Play__Robot_Vic_Sfx__Timer_Alarm	
3329286691	Stop__Robot_Vic_Sfx__Scan_Loop_Stop	
333319843	Play__Robot_Vic_SFX__Head_Up_Long_Angry	
3369051132	Play__Robot_Vic_Sfx__Tread_Loop_Play	

ID	Trigger	Description
3376462589	Play__Robot_Vic_SFX__Head_Down_Micro_Happy	
3379703224	Play__Dev_Robot_Factory__Tone_1k_5sec	
3396586077	Play__Robot_Vic_Sfx__Emote_Feedback_Good_Robot	
3404666410	Play__Robot_Vic_Sfx__Look_At_Device	
340818175	Play__Robot_Vic_SFX__Lift_High_Up_Short_Neutral	
3428971105	Play__Robot_Vic_Sfx__Blackjack_Swipe	
3430886582	Play__Robot_Vic_Sfx__Concentrate_Success	
3462246304	Play__Dev_Device__External_Source	
3463435173	Play_Robot_Vic_SFX_Head_Down_Short_Curious_Dev	
3478768817	Play__Robot_Vic_Alexa__External_Media_Pause	
3488492107	Play__Dev_Robot__Tone_1760Hz_5sec	
3504133917	Play__Robot_Vic_Sfx__Tread_Surprised_Long	
3515369910	Play__Robot_Vic_SFX__Head_Up_Micro_Effort	
3515689336	Play__Robot_Vic_Sfx__Holiday_Confetti_Build	
3522081323	Play__Robot_Vic_SFX__Lift_High_Down_Short_Curious	
3553396295	Play__Robot_Vic_Sfx__Low_Light_Charging_Loop_Play	
3557132089	Stop__Robot_Vic_Sfx__Shaking_Level_3_Stop	
3557511739	Stop__Robot_Vic_Sfx__Emote_Stop	
3560735309	Play_Robot_Vic_SFX_Head_Up_Short_Neutral_Dev	
3566487636	Play__Robot_Vic_Sfx__Lift_High_Down_Short	
3575715931	Play__Robot_Vic_Scene__Anim_Abort	
3588927098	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Excited	
3594885818	Play__Robot_Vic_Sfx__Scrn_Sad_Short	
3606332945	Play__Robot_Vic_Sfx__Emote_Feedback_Shut_Up_Eye_Roll	
3616541293	Play__Robot_Vic_Sfx__Power_On_Crooked	
3624594237	Stop__Robot_Vic_Sfx__Purr_Loop_Stop	
3633273702	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Frustrated	
364591870	Play__Robot_Vic_Sfx__Power_On_Mismatched_Eyes	
364928220	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Sad	
3654126394	Play__Robot_Vic_SFX__Lift_High_Up_Long_Happy	
3670008418	Play__Robot_Vic_Sfx__Lift_High_Weather_Shiver_Loop	
3679685607	Play__Robot_Vic_Sfx__Weather_Windy_Eye_Flyoff	
3688286452	Play__Robot_Vic_Sfx__Blink	
3691452503	Play__Robot_Vic_Sfx__Blackjack_Deal	
3705436708	Play__Robot_Vic_Sfx__Attention_Device_Phone	

ID	Trigger	Description
3711429789	Play__Robot_Vic_Sfx__Scan_Face_Loop_Play	
3712102539	Stop__Robot_Vic_Sfx__Knowledge_Graph_Listening_Loop_Stop	
3714792276	Play__Robot_Vic_Sfx__Holiday_Fireworks_Start	
3717200567	Play__Robot_Vic_Sfx__Snore	
3743401560	Play__Robot_Vic_SFX__Lift_High_Up_Short_Excited	
3765223799	Play__Robot_Vic_SFX__Head_Down_Short_Curious	
3767391969	Play__Robot_Vic_Sfx__Weather_Cloudy	
3773492765	Play__Robot_Vic_SFX__Head_Down_Long_Frustrated	
3780596321	Play__Dev_Squawk__Head_Long_Curious	
3797626084	Play__Robot_Vic_Sfx__Blackjack_Spread	
3802306023	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Sad	
3825278913	Play__Robot_Vic_Sfx__Head_Up_Long	
3831159749	Play__Robot_Vic_Sfx__Working_Loop_Play	
3841377270	Play__Robot_Vic_SFX__Head_Down_Short_Surprised	
384504930	Play__Robot_Vic_Sfx__Onboarding_Power_On_Initialize	
3875852678	Play__Robot_Vic_SFX__Head_Up_Short_Sad	
3894118537	Play__Robot_Vic_Sfx__Tread_Happy	
389997670	Play__Robot_Vic_SFX__Head_Up_Micro_Excited	
3912070771	Play__Dev_Squawk__Blackjack_Lose	
3914244997	Play__Robot_Vic_SFX__Head_Down_Short_Sad	
3920361320	Play__Robot_Vic_Sfx__Lift_High_Petting_Level_04	
3920361325	Play__Robot_Vic_Sfx__Lift_High_Petting_Level_01	
3920361326	Play__Robot_Vic_Sfx__Lift_High_Petting_Level_02	
3920361327	Play__Robot_Vic_Sfx__Lift_High_Petting_Level_03	
392476676	Play__Dev_Robot_Factory__Tone_1k_1sec	
3927586993	Stop__Robot_Vic_Sfx__Knowledge_Graph_Loop_Stop	
3950866384	Play__Robot_Vic_Sfx__Emote_Cant_Do_That_2	
3950866385	Play__Robot_Vic_Sfx__Emote_Cant_Do_That_3	
3950866387	Play__Robot_Vic_Sfx__Emote_Cant_Do_That_1	
3950866390	Play__Robot_Vic_Sfx__Emote_Cant_Do_That_4	
3953422061	Play__Robot_Vic_Sfx__Purr_Loop_Play	
3954155308	Play__Robot_Vic_SFX__Lift_High_Up_Long_Sad	
3965268362	Play__Robot_Vic_Sfx__Holiday_Lights_Shake_Long	
3965747163	Play__Robot_Vic_Sfx__Lift_Loop_Play	
3968555817	Play__Robot_Vic_SFX__Lift_High_Down_Long_Frustrated	

ID	Trigger	Description
3970067883	Play__Robot_Vic_SFX__Head_Up_Micro_Surprised	
3980968688	Play__Dev_Robot__Pink_1sec	
3981041882	Play__Robot_Vic_Sfx__Blackjack_Getin	
3995516895	Play__Robot_Vic_Sfx__No_Wifi_Icon	
3995789686	Play__Robot_Vic_SFX__Head_Up_Micro_Happy	
4005853328	Play__Robot_Vic_SFX__Head_Down_Short_Neutral	
4014518887	Play__Robot_Vic_Sfx__Tread_Sad	
4031777881	Play__Robot_Vic_Sfx__Scrn_Surprised_Short	
4035154116	Play__Robot_Vic_Alexa__Sfx_Sml_Utility_500ms_Blank	
4042659783	Play__Robot_Vic_Sfx__Weather_Cold	
4046953533	Play_Robot_Vic_SFX_Head_Up_Short_Sad_Dev_02	
4063827073	Play__Dev_Squawk__Camera_Flash	
4068444155	Play__Dev_Robot__Tone_1k_1sec	
4070524900	Play__Robot_Vic_Sfx__Emote_Weather_Thunder	
4074272165	Play__Robot_Vic_Sfx__Dsp_Loop_Play	
4080696361	Play__Robot_Vic_Sfx__Concentrate_Fail	
4117194074	Play__Robot_Vic_Sfx__Snowglobe_Chime_Shakeoff_Start	
4128432838	Play__Dev_Robot__Freq_Sweep_20sec	
4129494295	Play__Robot_Vic_SFX__Head_Down_Long_Curious	
4157094450	Play__Robot_Vic_SFX__Head_Up_Micro_Curious	
41581170	Play__Robot_Vic_Sfx__Purr_Single	
4159310964	Play__Robot_Vic_Sfx__Tread_Angry	
4190885404	Play__Robot_Vic_Sfx__Emote_Greeting_Goodbye	
4193980929	Stop__Robot_Vic_Sfx__Snowglobe_Chime_Loop_Slow_Stop	
4206946690	Play__Robot_Vic_Sfx__Emote_Curious_Long_Stim	
4211032222	Play__Dev_Squawk__Wake_Word_On	
4217238598	Play__Robot_Vic_Sfx__Camera_Focus	
4218051108	Play__Robot_Vic_Sfx__Emote_Weather_Rain	
4223875502	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Curious	
4233613545	Play__Robot_Vic_SFX__Head_Up_Micro_Neutral	
4236291785	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Excited	
4246099063	Play__Robot_Vic_SFX__Lift_High_Down_Micro_Frustrated	
4257411731	Play__Robot_Vic_Sfx__Emote_Feedback_Love	
4266515628	Play__Robot_Vic_Alexa__External_Notifications_Resume	
4271694941	Play__Robot_Vic_SFX__Head_Down_Short_Effort	

ID	Trigger	Description
4274302454	Play__Robot_Vic_SFX__Lift_High_Up_Long_Frustrated	
4289674931	Play__Dev_Robot__White_5sec	
434655980	Play__Robot_Vic_Sfx__Weather_Stars	
435573870	Play__Robot_Vic_Sfx__Holiday_Lights_Move	
436264283	Play__Robot_Vic_Sfx__Low_Light_Charging_End	
45686810	Play__Robot_Vic_SFX__Lift_High_Up_Long_Effort	
482240837	Play__Robot_Vic_Sfx__Holiday_Lights_Shake_Medium	
485658282	Play__Robot_Vic_SFX__Head_Up_Long_Curious	
497523539	Play__Robot_Vic_Sfx__Shaking_Slowmo	
505667781	Play__Robot_Vic_Sfx__Shaking_Level_3_Play	
509799084	Play__Robot_Vic_SFX__Lift_High_Up_Short_Frustrated	
514935507	Play__Robot_Vic_SFX__Lift_High_Down_Long_Excited	
531078287	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Angry	
540775366	Play__Dev_Squawk__Emote_Happy	
544144366	Play__Robot_Vic_SFX__Head_Up_Long_Effort	
555760346	Stop__Robot_Vic_Sfx__Snowglobe_Global_Loop_Stop	
567951592	Play_Robot_Vic_SFX_Head_Up_Short_Sad_Dev	
571987810	Play__Robot_Vic_SFX__Head_Down_Micro_Neutral	
596052867	Play__Robot_Vic_Scene__Low_Power_Mode_On	
621662666	Play__Dev_Squawk__Tread_Happy	
624164417	Play__Robot_Vic_Sfx__No_Wifi_Icon_End	
627442460	Play__Dev_Robot__Mozart	
635575819	Play__Robot_Vic_SFX__Lift_High_Down_Short_Excited	
649069951	Play__Robot_Vic_Sfx__Holiday_Confetti_Explode	
652872314	Play__Robot_Vic_Sfx__Alexa_Display_Off	
661891843	Play__Robot_Vic_Sfx__Lift_Down_Short	
670694969	Stop__Robot_Vic_Sfx__Scan_Face_Loop_Stop	
67389845	Play__Robot_Vic_Sfx__Scrn_Procedural_Blink	
686615416	Play__Robot_Vic_SFX__Head_Up_Short_Happy	
711426914	Play__Robot_Vic_Sfx__Snowglobe_Wind_Loop_Fast_Play	
732391128	Play__Robot_Vic_Sfx__Knowledge_Graph_Listening_Loop_End	
736173087	Stop__Robot_Vic_Sfx__Lift_Loop_Stop	
739422973	Play__Robot_Vic_SFX__Head_Down_Short_Frustrated	
761202130	Play__Robot_Vic_Sfx__Knowledge_Graph_Searching_Loop_Start	
76586799	Play__Robot_Vic_Sfx__Wake_Word_Off	

ID	Trigger	Description
789391628	Play_Robot_Vic_SFX_Up_Long_Neutral_Dev	
791376499	Play__Robot_Vic_SFX__Lift_High_Down_Long_Curious	
797290054	Stop__Robot_Vic_Sfx__Head_Loop_Stop	
815245339	Play__Robot_Vic_Alexa__Sfx_Sml_State_Bluetooth_Disconnected	
816586909	Stop__Robot_Vic_Sfx__Dsp_Loop_Stop	
827015036	Play__Robot_Vic_Sfx__Holiday_Lights_Disappear	
83693475	Play__Robot_Vic_SFX__Head_Up_Short_Neutral	
837073000	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_Eyes_To_Lights_1	
854495775	Play__Robot_Vic_Alexa__Sfx_Sml_State_Privacy_Mode_On	
857094875	Play__Robot_Vic_Alexa__External_Media_Play	
883462549	Play__Robot_Vic_SFX__Lift_High_Up_Short_Angry	
902182529	Play__Robot_Vic_Sfx__Head_Petting_Level_01	
902182530	Play__Robot_Vic_Sfx__Head_Petting_Level_02	
902182531	Play__Robot_Vic_Sfx__Head_Petting_Level_03	
902182532	Play__Robot_Vic_Sfx__Head_Petting_Level_04	
919702861	Play__Robot_Vic_Sfx__Scrn_Holiday_Lights_Disappear	
9207464	Play__Robot_Vic_Sfx__Scrn_Power_On_Eye_Fix	
923666381	Play__Robot_Vic_Sfx__Emote_Feedback_Bad_Robot	
941039741	Play__Robot_Vic_SFX__Head_Up_Short_Surprised	
948923880	Play__Robot_Vic_SFX__Head_Down_Micro_Angry	
94934096	Play__Robot_Vic_Alexa__External_Media_Resume	
954391655	Play__Robot_Vic_SFX__Lift_High_Up_Micro_Surprised	
969212508	Play__Dev_Squawk__Fist_Bump	
96988472	Play__Robot_Vic_Sfx__Knowledge_Graph_Searching_Loop_Play	
978953316	Play__Robot_Vic_Sfx__How_Old_Loop_Play	
988519260	Play__Dev_Robot__Pink_5sec	
990175872	Stop__Robot_Vic_Sfx__Tread_Loop_Stop	
992178953	Play__Robot_Vic_Sfx__Lift_High_Dancing_Small	
995249895	Play__Robot_Vic_Sfx__Shaking_Level_1_Play	

13.20 Sound Parameters

Vector's sound engine is designed to give a more life-like quality to his sounds. It does this by varying the sounds it plays when an audio event is received by the audio engine. Instead of playing a single, fixed sound file, a sequence of semi-randomized sounds is played. Some events have many possible sequences that could be played. The audio engine selects the sequence based primarily on Vector's current level of stimulation.

13.20.1 Input parameters (inputs from the animation system and engine)

Audio parameters are settable values passed to the audio engine, guiding it in how it selects and plays the sounds. (These are called "game parameters" in Wwise parlance.) Vector mainly uses these to adjust the sounds based on his current mood and activity. Like the action, these parameters are on a per object (within the audio engine) basis.

The sound engine is connected to the high-level state of the other major robot sub-systems, so that the sounds could reflect what Vector's body is doing, his mood, emotional state, level of stimulation, and even to respond to the environment. Although many parameters are provisioned in the soundbanks, only few are used: they were too hard to get right.

The parameters linked to the emotion state & mood:

Parameters	Id	Description
Robot_Vic_Confident	2193267925	This captures the emotion dimension "confidence"
Robot_Vic_Happy	1391855411	This captures the emotion dimension "happy"
Robot_Vic_Held_Trust	1255095877	This captures the emotion dimension "trust"
Robot_Vic_Purr_Level	3113529605	
Robot_Vic_Social	240199822	This captures the emotion dimension "social"
Robot_Vic_Stimulation	651061178	"This stimulation variable is a distillation of all possible environmental affects Vector experiences. For example, 'hey Vector' or touch automatically triggers a 1.0, getting stuck drop the stim to about 0.5, etc. .. We adapted the audio system to gradually lower in probability and volume as the stimulation level lowered. The goal being that active Vector users get a more lively sounding robot, and Vectors left on but not being interacted with wouldn't be so chirpy." (Ben Gabaldon)

The parameters linked to the body's posture and movement:

Parameters	Id	Description
Robot_Vic_Head_Accelerate	2691629051	How fast the head is changing speed.
Robot_Vic_Head_Position	1549905781	This captures the heads position: where it is looking.
Robot_Vic_Head_Speed	4244799613	This is linked to the speed that the head is moving at, and the direction it is moving in.
Robot_Vic_Lift_Accelerate	1685856300	How fast the lift is changing speed.
Robot_Vic_Lift_Position	4014791842	This is the position of the lift.
Robot_Vic_Lift_Speed	2317283412	This is linked to the speed that the lift is moving at, and the direction it is moving in.
Robot_Vic_Tread_Accelerate	496587279	How fast the treads are changing speed.
Robot_Vic_Tread_Speed	724346585	This is linked to the speed that the treads are moving, and the direction they are moving in.
Robot_Vic_Tread_Spin_Speed	3313708352	This is linked to the speed that the robot is spinning, and the direction it is spinning in.

Parameters	Id	Description
Dev_Squawk_Volume	847280258	
Dev_Tone_Freq	2324555758	
Event_Volume	3530059848	
External_Shape	1292394150	
External_Speed	732885492	
Robot_Alexa_Volume_Alerts	522470314	
Robot_Alexa_Volume_Master	3289463297	
Robot_Vic_Environment_Ambient_Volume	344494516	
Robot_Vic_How_Old_Speed	4000308525	
Robot_Vic_Meter_Bus_SFX	1248968661	
Robot_Vic_Meter_Bus_TTS	913563323	
Robot_Vic_Meter_Bus_VO	3946120957	
Robot_Vic_Planning	954718164	
Robot_Vic_Screen_Shift_Interpolation_Time	1775375659	
Robot_Vic_Sdk_Volume	995643170	
Robot_Vic_Shaking	2064796514	
Robot_Vic_Timer_Countdown	165240122	
Robot_Vic_Volume_Animation	1008373634	
Robot_Vic_Volume_Behavior	359550870	
Robot_Vic_Volume_Master	1036978186	This is linked to the volume preference setting, controlled by the mobile application (or SDK).
Robot_Vic_Volume_Procedural	346758509	
Ss_Air_Fear	1351367891	
Ss_Air_Freefall	3002758120	
Ss_Air_Fury	1029930033	
Ss_Air_Month	2648548617	
Ss_Air_Presence	3847924954	
Ss_Air_RPM	822163944	
Ss_Air_Size	3074696722	
Ss_Air_Storm	3715662592	
Ss_Air_Timeofday	3203397129	
Ss_Air_Turbulence	4160247818	

13.20.2 Derived sound settings

Many events trigger a sound which is made of a sequence of smaller sounds. In some cases there are several possible sequences, selected by a switch setting.

Those inputs above are smashed and mashed down into a switch setting, that the sound banks used to select from different sound alternatives. A switch, for those familiar with software, is a variable that has an enumeration type. Except that, in Vector's sound banks, the enumeration is unique in each case.

Parameter	Switch States	Ids	Description
Robot_Vic_Head_Speed	Robot_Vic_Head_Direction_Down	3234804922	
	Robot_Vic_Head_Direction_Up	123512153	
Robot_Vic_Lift_Speed	Robot_Vic_Lift_Direction_Down	3238791869	
	Robot_Vic_Lift_Direction_Up	807633034	
Robot_Vic_Screen_Shift_Interpolation_Time	Shift_Short	3156794696	
	Shift_Long	3265274760	
Robot_Vic_Stimulation	Stim_01	446181668	
	Stim_02	446181671	
	Stim_03	446181670	
	Stim_04	446181665	
Robot_Vic_Tread_Spin_Speed	Robot_Vic_Tread_Drive	4283612440	
	Robot_Vic_Tread_Spin	2961118536	
Robot_Vic_Tread_Speed	Robot_Vic_Tread_Backward	3258446947	
	Robot_Vic_Tread_Forward	1541107887	

13.20.3 Change history synopsis

Date	Change
2021-2-26	Created
2021-7-3	Added ids

13.21 Start up behavior

Summary: The initial startup behavior that kicks off normal operation. (These are self-maintenance behaviors)

When Vector's application starts, it looks up a top level state to kick off the initial behavior -- has the robot been tested at the factory? Has the owner gone thru on boarding? And so on.

This behavior is the root of the behavior tree that Vector will use. There are 7 of these broad, top level states:

- PR demo
- Factory test (e.g. the playpen tests)
- Acoustic testing
- On-boarding
- Post on-boarding
- Normal
- Developer

Note: when Vector exits the Customer Care screens, it resumes operation by re-running the top level behavior.

13.21.1 Mapping to the initial behavior

These top-level states are mapped to initial behavior using `victor_behavior_config.json`

State	Starting Behavior
factory	
acoustic testing	AcousticTestMode
onboardingBehavior	Onboarding
postOnboardingBehavior	ModeSelector
normalBaseBehavior	InitNormalOperation
devBaseBehavior	DevBaseBehavior
PR demo	InitPRDemo
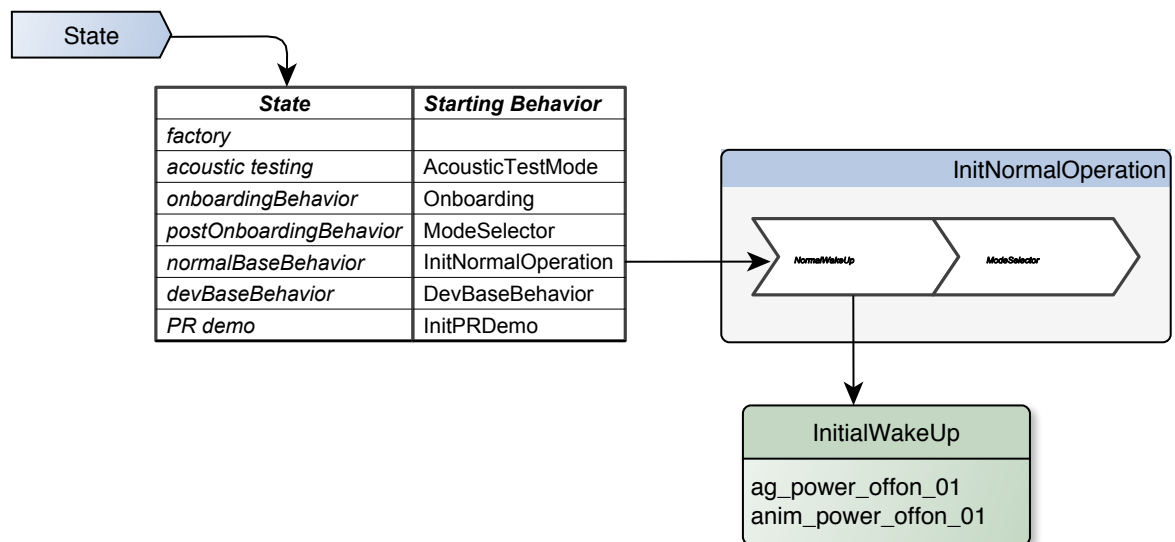
Figure: The start-up behavior tree

(In a few cases the mapping is hardcoded in the software.)

In normal operation, this is the *InitNormalOperation* behavior. The behavior file is located at:

```
behaviors/victorBehaviorTree/initNormalOperation.json
```

13.21.2 Walk thru of the InitNormalOperation behavior

When the *InitNormalOperation* (class *DispatcherStrictPriorityWithCooldown*) behavior first starts, it does a one-off run of a *NormalWakeUp* behavior. This behavior is not run when *InitNormalOperation* is started again later. (For instance, it is run again when the Customer Care screens are exited.) This one-shot execution is acheived by using settings its cooldown timer to a period that is infinitely long.

The *NormalWakeUp* (class *AnimSequence*) behavior checks to see that it is not night time, and not a maintenance reboot. It isn't, then it triggers the *InitialWakeUp* animation group. The animation affect Vectors eyes, head angle, backpack lights, and sounds. (There are not any other movements). This behavior file is located at:

```
behaviors/victorBehaviorTree/normalWakeUp.json
```

After this it defers to *ModeSelector* behavior for the top level, prioritized behavior dispatch.

13.21.3 Other variations

There are three other animation variations on Vector that are currently not used:

- `anim_power_offon_02`
- `anim_power_offon_03`
- `anim_power_offon_04`

13.21.4 Change history synopsis

Date	Change
2020-11-29	Created, setup format
2020-11-30	Added file references
2020-12-1	Moved some intro material to behavior tree

14. Tools

14.1 Animation tool

14.1.1 Maya

Anki used Maya to animate Cozmo and Vector. The tool used a plugin to emit the movements, as JSON using a format that the animation engine could read with the flatbuffers library. (See *How to convert animation bin files to JSON* for a bit more on converting between JSON text and the binary format.)

The animations tools had UIs with at least the following two screens:
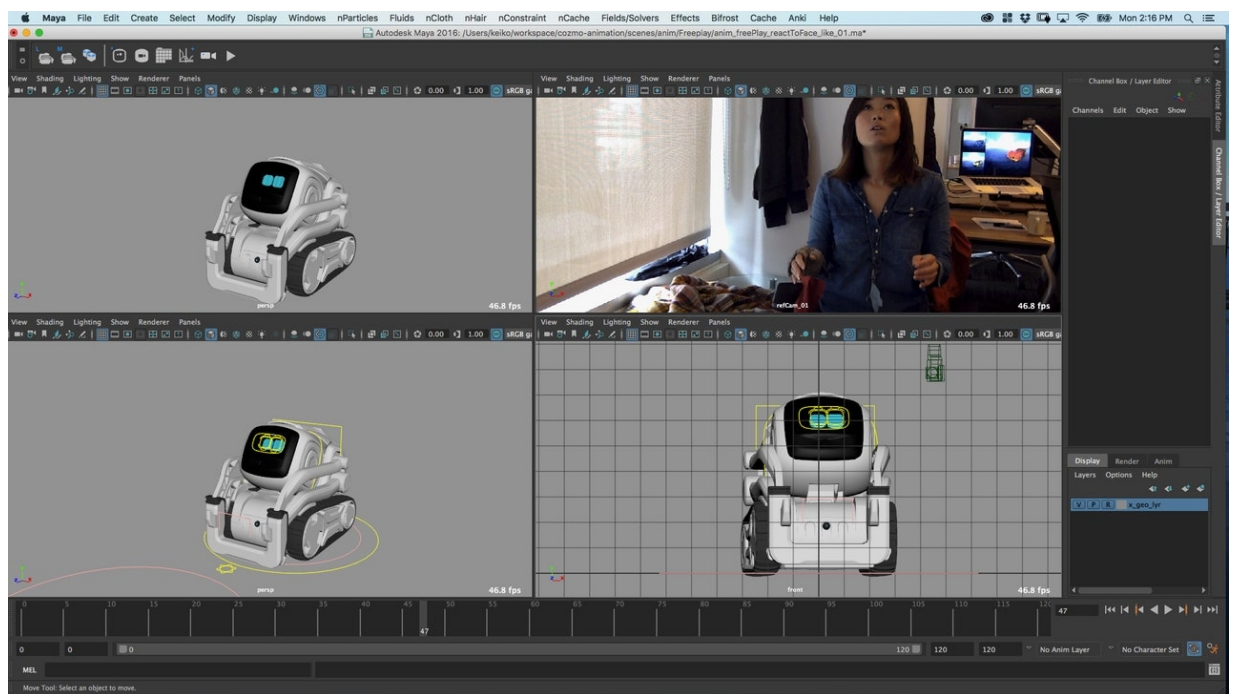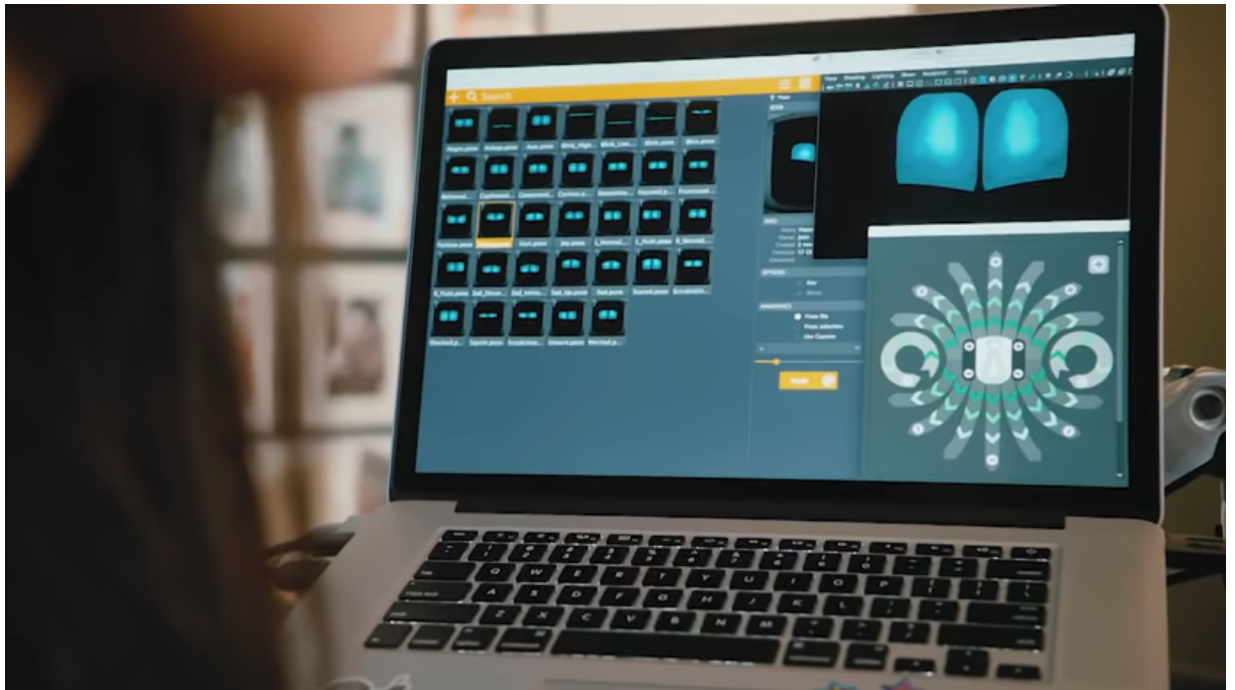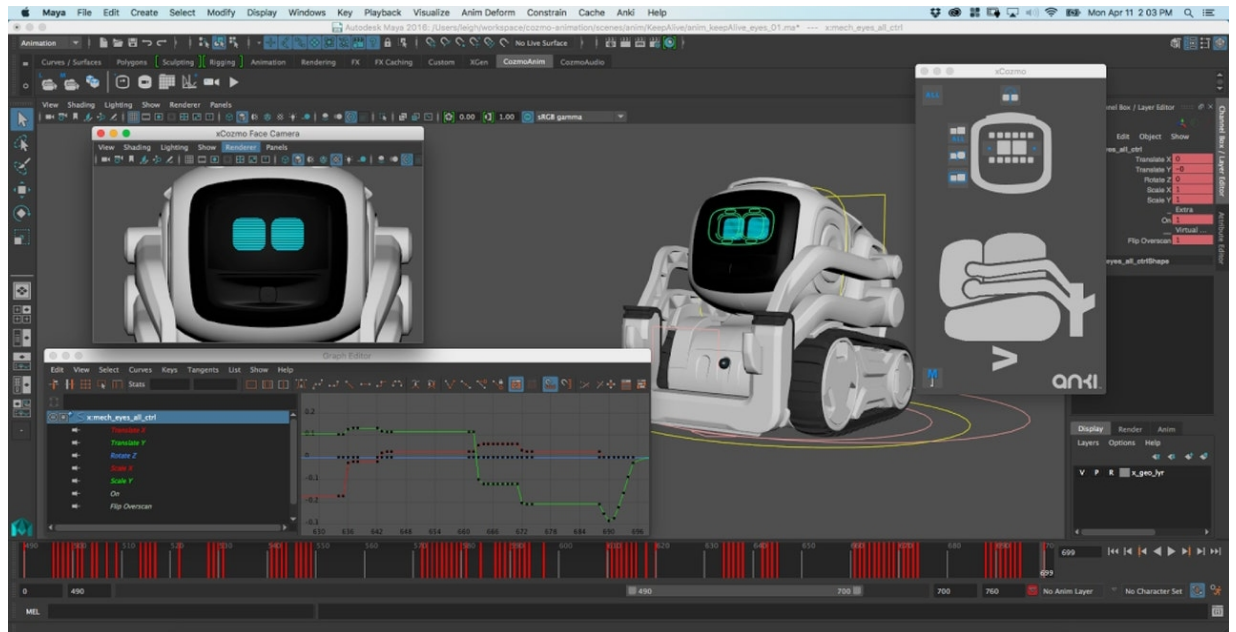
VIRTUAL VECTOR IN MAYA PERFORMING AN ANIMATION

REAL WORLD VECTOR PERFORMING THE SAME ANIMATION

More Info

- Maya Tutorial (Beginner Video 2016)
- AutoDesk Maya Tutorial Links

14.2 Eye Animation tools

The eye parameters in the animation files are very complex. FBS animation file in turn took different eye patterns, and interpolated movements between the points.

14.2.1 Individual frames

The animator's probably only rarely set up the eye in detail. Instead, they had tools to simplify the process. They were pre loaded with the eye settings for many common looks that took into account:

- Brow

- Gaze direction (left to right, up to down)

- Cheeks

- How open, closed, the eyes are such as eye squint (per eye?), or sleepiness

- Head tilt

- Squash and stretch -- at least, how close and far apart the eyes are

We saw some of the animators tools in magazines photos, and they give clues to the controls that they worked with.   Below are screen grabs of the eye animation tools for Cozmo and Vector:

These tools likely translate the button settings into one of a couple of dozen "canned" eye patterns (pre-programmed eye parameters).

Cozmo's eye configuration tool

Figure: Cozmo's eye configuration tool

Vector's eye configuration tool

Figure: Vector's eye configuration tool

We can clearly see the controls to where Vector should look, some of his relative eye shape, and so on.

14.2.2 What about movement, and playing with movement

On Cozmo and Vector, the animation system moves its working version of each the parameters to transition from frame to frame over the given time period.

On the desktop, the animation tools likely do this as well -- looping movements between the eye frames to checkout the eye motion.

There probably was a style guide or informal rules on the kinds of frames to use (and when) for eye shape and movement to reflect:

- Internal state like emotion, mood, stress,

- Blinking, and breathing.

- Sleepiness and interest

14.2.3 JoystickNSliders techniques

Last spring Digital Dream Labs animator Molly Wright made an interesting video using a JoystickNSliders to animate a Cozmo/ Vector fact. This tool probably reflects how these could have worked.

- Making Cozmo and Vector an UwU Face #joysticksnsliders (May 22, 2020)

This one is interesting in presenting a bit more of how to think about the eyes. It appears to have simplified the eye controls four axis:

1. Worry vs Curiosity.

2. Mad vs happy

3. A blink rate, from slow to fast

4. A gaze direction of left to right

The tool works by setting up the eye parameters for each end of the slider, and perhaps the middle. These form the template for how the eyes look. Then, as the sliders move, it performs an interpolation between each of these points.

Some, like the blinking, may be a bit of timer and controls a few other intermediate steps.

14.3 Playpen

Playpen is a test station used calibration Vector and Cozmo's camera, and perhaps other sensors.

> Once Cozmo is fully assembled, he's placed in the Playpen to take his "final test". He does a lot of things in there, but one thing he does is an eye test. He drives around from target to target, making sure he can count all the dots, they're all in focus, and they're all where he expects them to be (literally, his head is on straight!).discord

14.3.1 Cozmo's playpen

This is what one looks like for Cozmo:

FULL SYSTEM TEST + SENSOR CALIBRATION
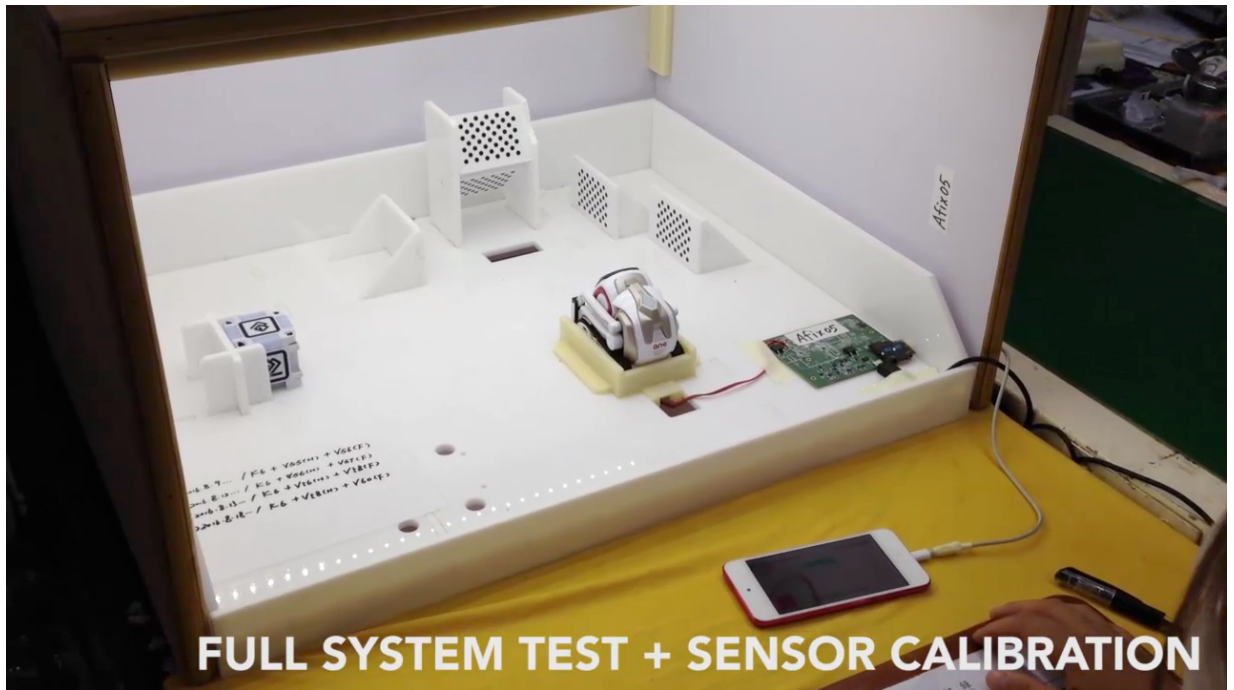
And for Vector

And this is what Cozmo sees:
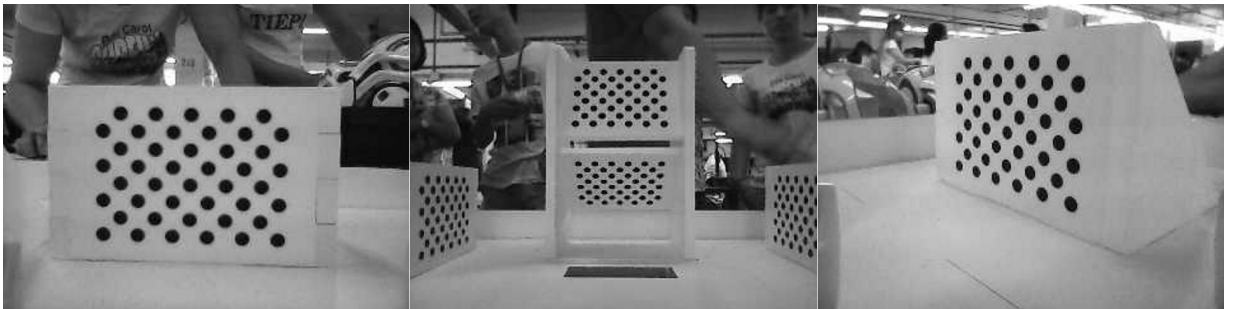
A direct shot of a calibration image that Vector sees:

14.3.2 Creating a new one for Vector?

TODO / TBD: We don't know how to make a replacement one yet. Or all the steps in issuing commands to Vector.

14.4 Webots

Anki used Webots to test Cozmo and Vector's emotion model on the desktop before downloading. And perhaps the behavior tree.

Cozmo's Mood Manager can be visualized in Webots, a software program designed for the development and simulation of robots. Labeled by the small, multi-color words (top right-hand corner of the photo), data points change over time based on Cozmo's mood.(Interview with Sr Sound Designer Ben Gabaldon)
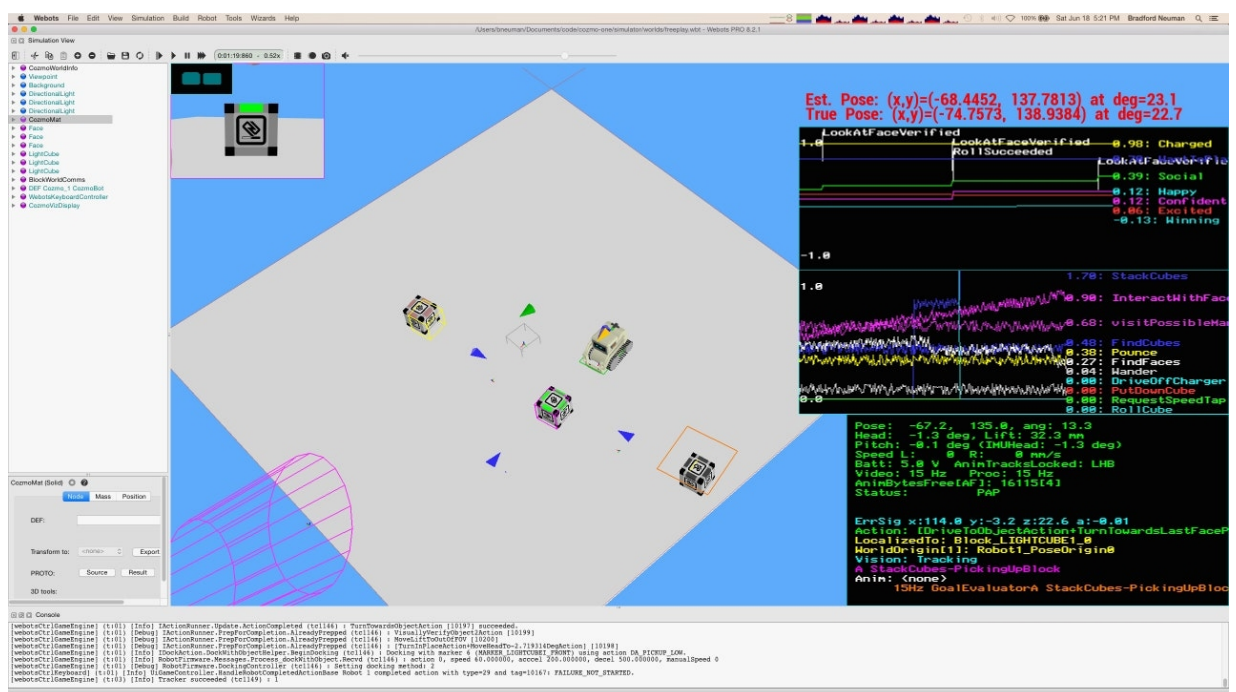
15. Troubleshooting

15.1 Backpack Lights

VECTOR LED SUMMARY

LEDs	Animation	What it means
●	The small circular light glows a steady green.	Vector is on.
	The rectangular lights glow green and climb up his Back button.	**Charging:** Vector is getting more energy. When he's done, his rectangular lights will stop glowing.
	The rectangular lights glow light blue and climb up his Back button 1 then 2 then 3.	**Booting:** Vector is starting up. When he's done, these rectangular lights will stop glowing light blue.
	The rectangular lights glow solid blue.	**Speaking to Vector:** After you've said "Hey Vector", Vector is ready to hear what you have to say.
	The rectangular lights will pulse orange continuously.	**Connection.** Vector can't connect to Wi-Fi. Connect with the Vector app to figure out what's happening.
	The rectangular red light on the bottom of his backpack will pulse..	**Low battery:** Place Vector on his charger.

Things that this could do:

- Diagram of the backpack lights

- Show the FAC lights

- Changes to the backpack lights in the custom software

See also DDL.

15.2 Purple circle light

- For the first few seconds at boot, this is normal and should get fixed later on in the body board boot process. However, if your Vector is stuck on it and he shows an error code (801, 898, 899), there could be an issue. First try to reboot by holding the button for 5-6 seconds. If that doesn't work, leave him and let the battery die. This will probably take a few hours. After the battery dies, turn him back on. If he still shows a purple light after being turned back on, then there is a hardware issue on the bodyboard.

15.3 Escape pod setup troubleshooting

This is a page for troubleshooting the EscapePod software set up. See also

- Pi equipment for information about the Raspberry Pi and charger that you will need

- Pi issues for information related to troubleshooting the Pi hardware

- Pi connectivity to troubleshoot the connection between your Pi, network and computer. This includes Wifi, and mDNS troubleshooting.

15.3.1 License code was not sent to Escape pod

Symptom: After the hot-word, the wifi/no cloud animation will play. There isn;t be a cycling white lights on the backpack. This happens repeatedly.

This is indicates that the license code for this bot wasn't added to the Escape Pod. (I know I've forgotten that once or twice.)

Follow the steps below to add a license.

15.3.2 Unable to add the license

When going to the escape pod {the URL https://escapepod.local:8443/} you should see a screen like:
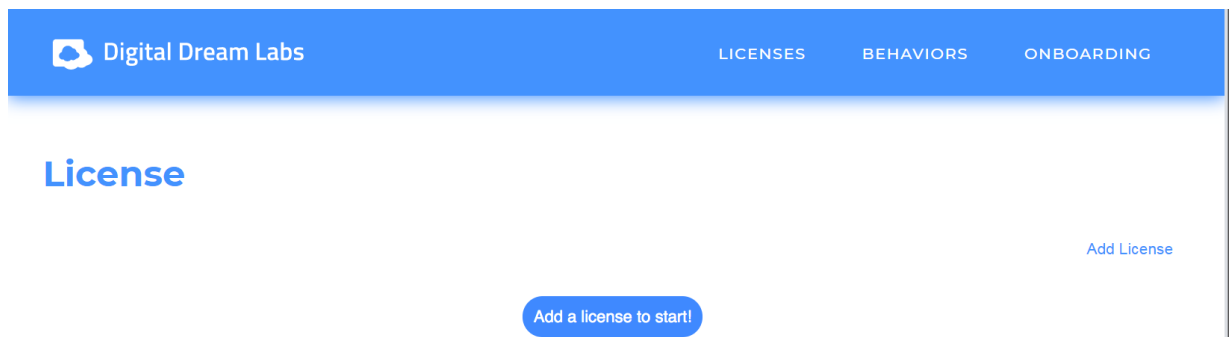
Figure: You should see a blue add license button

If you see the blue-tone Digital Dream Labs screen, but do not see the "Add license to start" button there is a problem. This is the button that you should see:
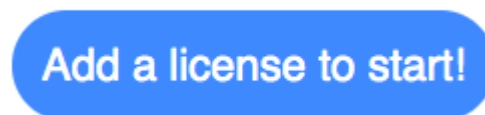
Figure: Add license button

For instance, if you see swirling spinner -- it can be subtle -- like this:

License

Add License
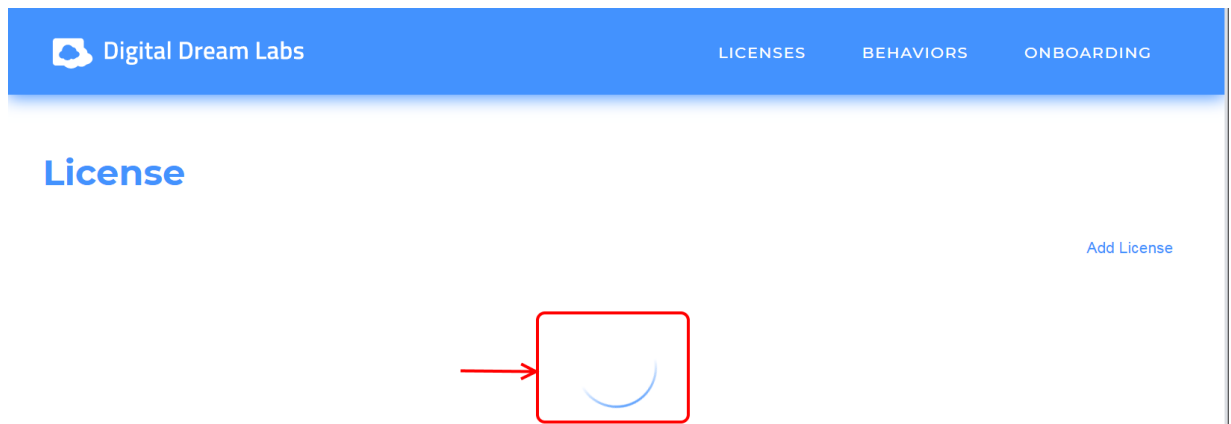
Figure: Swirling blue waiter

this indicates a network naming issue.

Open the browsers error console. Look for messages like the following:

```
    Failed to load resource: net::ERR_NAME_NOT_RESOLVED
    .local:8085/v1/license/add:1 Failed to load resource: net::ERR_NAME_NOT_RESOLVED
    DevTools failed to load SourceMap: Could not load content for https://escapepod:8443/react-router-dom.js.map:
 HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
```

If you have this problem, go to the following pages:

- Computer setup for information about the software to install on your computer, and other adjustments to make mDNs work.

- Pi connectivity to troubleshoot the mDNS configuration on your network and computer.

Alternatively, a person (me!) might have not noticed the swirly's significance, and clicked on "Add License". That will bring up a screen, where one can enter the license. After submitting, the page will appear unresponsive, then provide an error like:

```
 ***** The entered license is not valid. Make sure you typed it in correctly and try again.**
```

These might occur if the url used "escapepod" or "escapepod.lan" or an IP address to access the escapepod. The "escapepod.local" name is not resolving on the computer and it was hidden by the other names work. If you have this problem, go to the following pages:

- Computer setup for information about the software to install on your computer, and other adjustments to make mDNs work.

- Pi connectivity to troubleshoot the mDNS configuration on your network and computer.

(In my case this was fixed by correcting the OpenWRT router's default local domain.)

15.4 FAC (Factory) Mode

Vector has a "FAC" mode, used in the factory to test and calibrate the robot. When in FAC mode, the display has a red background, with either the letters "FAC" displayed:

Or one to two digits displayed. These appear to be calibration errors (makes sense since we don't have a playpen to calibrate them with).

And his backpack lights have an unusual color pattern – red, green, and blue:

This mode is never intended to be seen outside of the factory, so little is known. Only a couple of units have been found in this mode; one after it had been intentionally damaged, and its calibration & EMR data were corrupted or inaccessible. In all likelihood, the software checks its EMR to see if it has been released; if not, it enters the FAC mode at whatever the "next" stage is according to the EMR. At that point Vector expects to be placed into manufacturing test fixtures, such as the playpen.

If you see a normal Vector in this mode on a place like eBay, it is recommended you don't buy him. The software he is running is still 0.9.0 recovery just in a different mode and there aren't any dev things open.

15.5 Hardware error codes

If something has gone wrong, an error will appear on Vector's face. These error happen if Vector's hardware is bad, but some of them could also be software.

801

- Rampost was unable to communicate with the body board at boot. This will show up before an 898 or 899 error. If the board shows just a purple light and won't turn off, you need to wait for his battery to die. After that, turn him back on. If there is still a purple light, there could be a hardware issue with the body board.

870-895

- Body board has a specific hardware fault. For all of these, try a reboot. If that doesn't work, your Vector probably needs some fixin'.

870

- The front right microphone is not working correctly.

871

- The front left microphone is not working correctly.

872

- The back right microphone is not working correctly.

873

- The back left microhpone is not working correctly.

// I'm not sure if 870-873 actually show up as I have seen boards get through rampost without the backpack board connected.

890

- The front right cliff sensor is not working correctly.

891

- The front left cliff sensor is not working correctly.

892

- The back right cliff sensor is not working correctly.

893

- The back left cliff sensor is not working correctly.

894

- The front ToF sensor is not working correctly.

895

- The touch sensor is not working correctly.

896(?)-897

- Seems to be Whiskey specific. It seems to only show in their dev recovery and they work fine in normal firmware, so this doesn't seem to be a worry.

898

- There was an error when trying to communicate with the body board. If the board shows just a purple light and won't turn off, you need to wait for his battery to die. After that, turn him back on. If there is still a purple light, there could be a hardware issue with the body board.

899

- The firmware was unable to find the body. If the board shows just a purple light and won't turn off, you need to wait for his battery to die. After that, turn him back on. If there is still a purple light, there could be a hardware issue with the body board.

950

- This error will only occur on a Whiskey. The software is unable to open the extra ToF sensors. It is possible that one or both of the sensors are broken.
- This can be fixed with an EMR/OEM swap. Instructions soon

960

- IMU hardware failed.

970

- The Wi-Fi hardware failed.

980-981

- An error occured when trying to communicate with the camera. If he is stuck on this error, go to recovery and clear user data. If he is still stuck, the camera may not be soldered on well or it could be broken.

990

- Vic-anim is unable to open the display for writing. This is something you will probably never see.

15.6 Pi Network Troubleshooting

This is a page for troubleshooting the EscapePod software set up. See also

- Pi equipment for information about the Raspberry Pi and charger that you will need

- Pi issues for information related to troubleshooting the Pi hardware

- Troubleshooting the EscapePod software setup, especially adding licenses.

15.6.1 Symptoms of Network and Wifi Connectivity issues

Symptoms: The Raspberry Pi is connected to the network, but sometimes is unavailable; it appears to drop off. Fixes:

- Use wired ethernet if possible

- If using Wifi:

  - use only 2.5GHz -- remove the your 5GHz Wifi access points from the list.

  - Check that the WiFi configuration file syntax is correct

  - Don't forget to reboot twice

  - If using HDMI, reduce the resolution

  - Update the firmware

- Check the Router DHCP lease time; give a static/fixed IP address if possible

The following will help diagnose specific problems.

First Steps: The Ping test

First, double check that the Raspberry Pi is powered on, it's red LED is on.

Can you ping the EscapePod:

Try

```
ping escapepod.local
```

If you are able to ping using this step, the EscapePod is on the network, and mDNS is working. Go to Step TBD

Try

```
ping escapepod.lan
```

Try

```
ping escapepod.box
```

Try

```
ping escapepod
```

If you are able to ping using either of these two steps, the EscapePod is on the network, but there is a problem with mDNS. Go to step mDNS issues.

If you know the address for the escapepod, trying pinging it. If you can ping it, the issue is with mDNS. Go to step mDNS issues.

Otherwise your LAN or WiFi connection is not working. If you are using a ethernet LAN cable, check that it is fully plugged into the Pi and your router. If you are using WiFi, go to the next secton.i

15.6.2 Wifi Issues

Troubleshooting the Wifi. Note: this section does not apply if you are using an ethernet cable between your router and the escape pod.

If possible, it is recommended to use an ethernet cable -- at least until the EscapePod has been setup on your network and any other possible issues are resolved.

Did not reboot enough times after setting up the network config (ie config file not transferred to main linux)

Unplug the power from the raspberry pi. Plug the power in and give it a few minutes. Unplug the power from the raspberry pi *again*. Plug the power in and give it a few minutes. Repeat the ping test above.

Wifi config file typo/syntax error

It is possible that there is a typo in the wifi networking configuration file. Take the SD card out, and put it into your computer. Open the configuration file and double check that it looks right:

- Does it have the right spacing?
- Does the wifi Access Point name match the name of your Wifi? Any typos?
- Is the password the right one for your Wifi? Any typos?

Use 2.5GHz instead of 5GHz Wifi

The Raspberry Pi 3 doesn't work with 5GHz

The Raspberry Pi 4 is "internet reported" to be flaky with 5GHz, and periodically drop off.

HDMI Interference

If you are having network connectivity issues -- the WiFi works for a while then drops out -- it may be the HDMI. High resolution mode HDMI generates enough noise to wreck WiFi on Pi 4.

Fixes:

- Don't use HDMI so high of resolution
- Don't use 5Ghz
- Try updatign the firmware

UPDATING THE RASPBERRY PI FIRMWARE.

A firmware update can fix some of the HDMI interference. Here is how I updated it. As a first step I installed "rpi-eeprom-update" to list the current installed firmware, but it may have updated it on me (it was that or when I did an ubuntu update).

The following installation steps are from: https://askubuntu.com/questions/1253070/raspberry-pi-4-firmware-upgrade-eeprom-over-ubuntu-20-04

```
curl -O http://ports.ubuntu.com/pool/universe/r/raspberrypi-userland/
libraspberrypi0_0~20200520+git2fe4ca3-0ubuntu2_arm64.deb
sudo apt install ./libraspberrypi0_0~20200520+git2fe4ca3-0ubuntu2_arm64.deb
curl -O http://ports.ubuntu.com/pool/universe/r/raspberrypi-userland/libraspberrypi-
bin_0~20200520+git2fe4ca3-0ubuntu2_arm64.deb
sudo apt install ./libraspberrypi-bin_0~20200520+git2fe4ca3-0ubuntu2_arm64.deb
sudo add-apt-repository ppa:waveform/eeprom
sudo apt update
apt list --upgradable
```

After doing that (I had to do it several times to complete all of the myriads of updates) this is what I saw:

```
BCM2711 detected
VL805 firmware in bootloader EEPROM
*** UPDATE AVAILABLE ***
BOOTLOADER: update available
CURRENT: Thu Mar 19 14:27:25 UTC 2020 (1584628045)
 LATEST: Thu Sep  3 12:11:43 UTC 2020 (1599135103)
 FW DIR: /lib/firmware/raspberrypi/bootloader/critical
VL805: up-to-date
**CURRENT: 000137ad**
** LATEST: 000137ad**
```

A day later (with the unexplained firmware update):

```
sudo rpi-eeprom-update

BCM2711 detected
VL805 firmware in bootloader EEPROM
BOOTLOADER: up-to-date
CURRENT: Thu Sep  3 12:11:43 UTC 2020 (1599135103)
 LATEST: Thu Sep  3 12:11:43 UTC 2020 (1599135103)
 FW DIR: /lib/firmware/raspberrypi/bootloader/critical
VL805: up-to-date
**CURRENT: 000138a1**
** LATEST: 000138a1**
```

One of those tools I installed gives another way to display the version numbers (it's what the tool uses):

```
vcgencmd bootloader_version | grep timestamp

sudo lspci -d 1106:3483 -xxx | awk '/^50:/ { print "VL805 FW version: " $5 $4 $3 $2" }
```

The first displays the bootloader version and the second for the VL805 firmware. This can useful for identifying if an update is relevant before applying one.

15.6.3 mDNS issues

The network name for the escape pod should appear as "escapepod.local" But it may also appears as "escapepod" or "escapepod.lan" to your computer. In that case you won't be to use the web interface without problems.

There are a few different causes that could be making this happen. First, check that your computer has the right software installed and settings changes made. See

- Computer setup for information about the software to install on your computer, and other adjustments to make mDNs work.

Next, if you're still having trouble, your router may be contributing. There are too many to know ahead of time their configurations, but we know how to tweak OpenWRT.

Router-based issues

One possible source of problems is the router configuration. Here is what to look for to see if you have an OpenWRT-based (or Dnsmasq-based) router. (This section may not apply if you have a different kind of router; there are too many to know ahead of time their configurations.)

First, go to your network router, and choose the "Network" menu, and then select "DHCP and DNS" router:
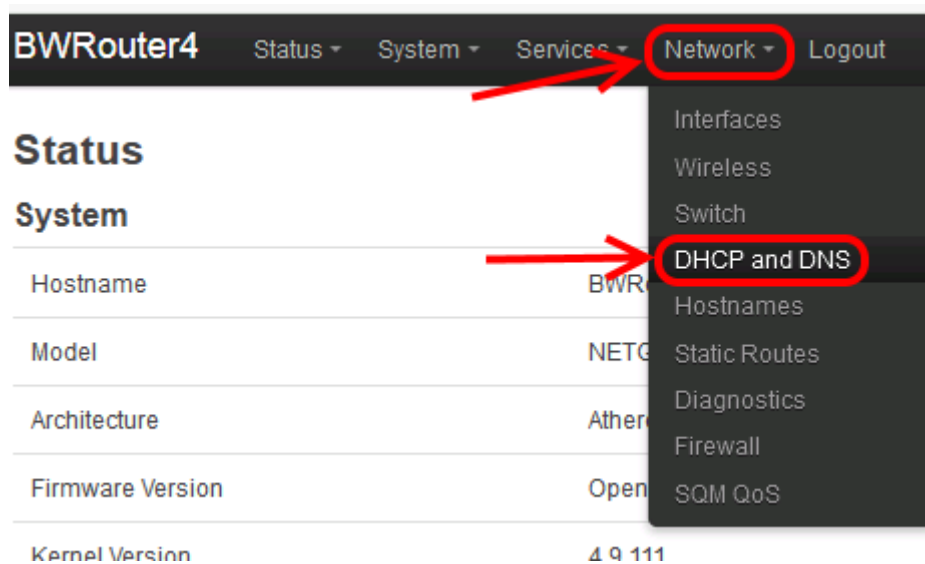
Figure: openWRT

This will open a settings page like the following:

Figure: openWRT local domain settting

The key thing to look for here is the "Local domain" In my case the local domain was set "lan" (possibly by default, or a choice I made long ago). For some german routers (e.g. from Telekom) may be set to "box". This setting explains why the name "escapepod.lan" worked. To fix the problem change the local domain to "local".

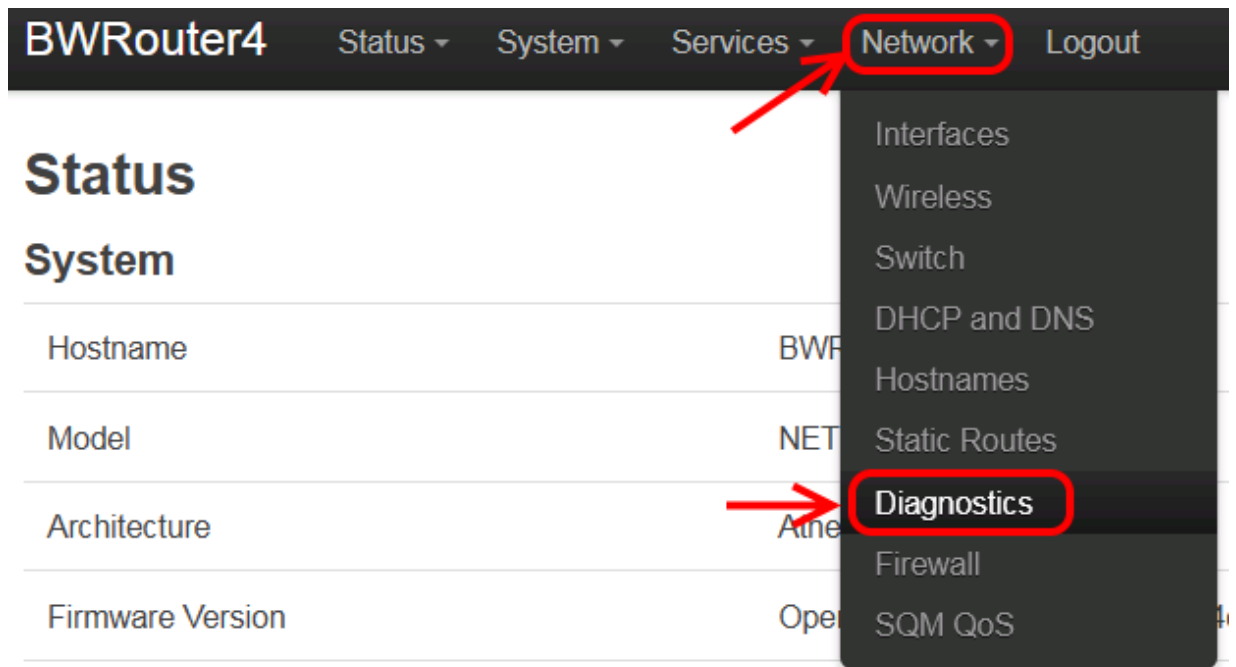Once the change has been made, test it on the router. Open the "Network" menu and select "Diagnostics":

Figure: openWRT diagnostics menu item

You will get a ping tool on the router:
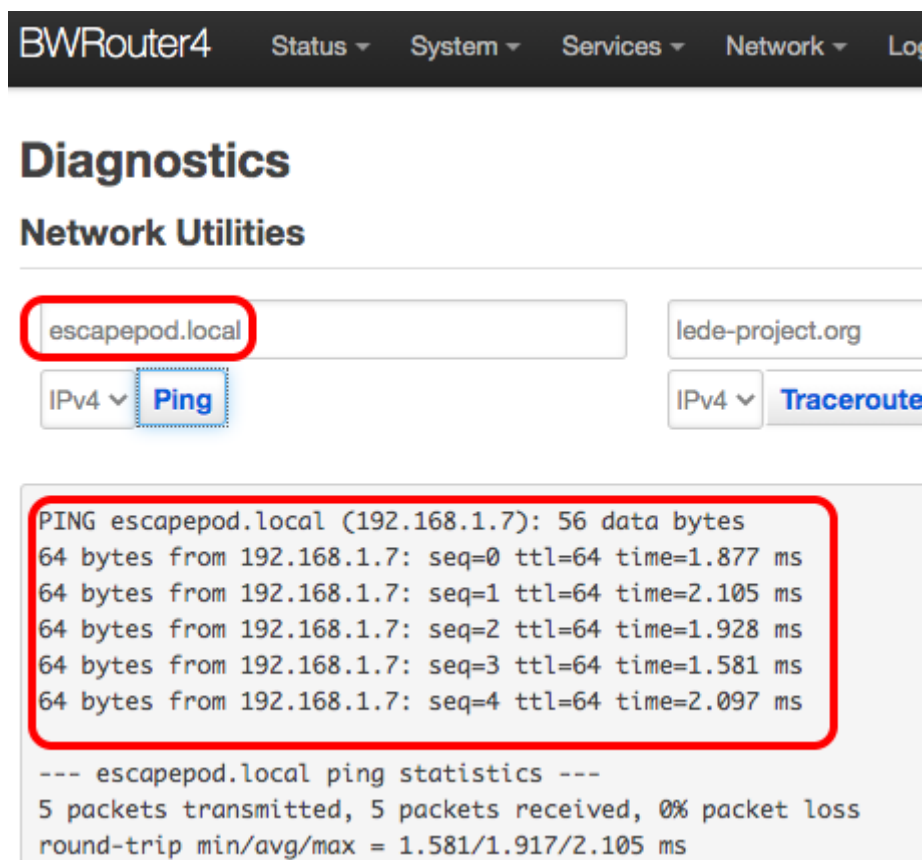
Figure: openWRT ping tool

Enter "escapepod.local" in the field, click ping. You should see successful pings from the escape pod. If not, there is another problem.

IPV6 SHORT LEASE TIMES CAUSING MDNS UPDATE ISSUES

With my OpenWRT (dnsmasq) router, the IPv6 lease time defaults to 40 hours (and it can't be changed). This may be a cause why the Vector is unable to find the EscapePod after awhile. The EscapePod has to renew it's addressing information and Vector is unable to reach it during the renewal.

The following worked with success for me:

- Use fixed IP addresses on the router

- Configured EscapePod to only use 2.5GHz WiFi access points

15.6.4 SSH related issues

SSH can not reach the Pi at all

First, follow the steps earlier to be sure that you can ping.

1. Can you ping? See section First Steps: The Ping test to perform the basic connectivity tests.

2. Check to see if yu can ssh to the IP address.

3. Does a message appear on the raspberry Pi console? That may indicate how far the connection made it; it may also reveal a power issue. See Pi issues for diagnosing power issues.

4. Can you log in on the console with a USB kbd and monitor? (user name ubuntu, password ubuntu)

SSH/Putty says it is "out of date: can't exchange keys"

If putty displays a message that says it can't exchange keys, this indicates you (well, I) have a very old version of putty. For example:
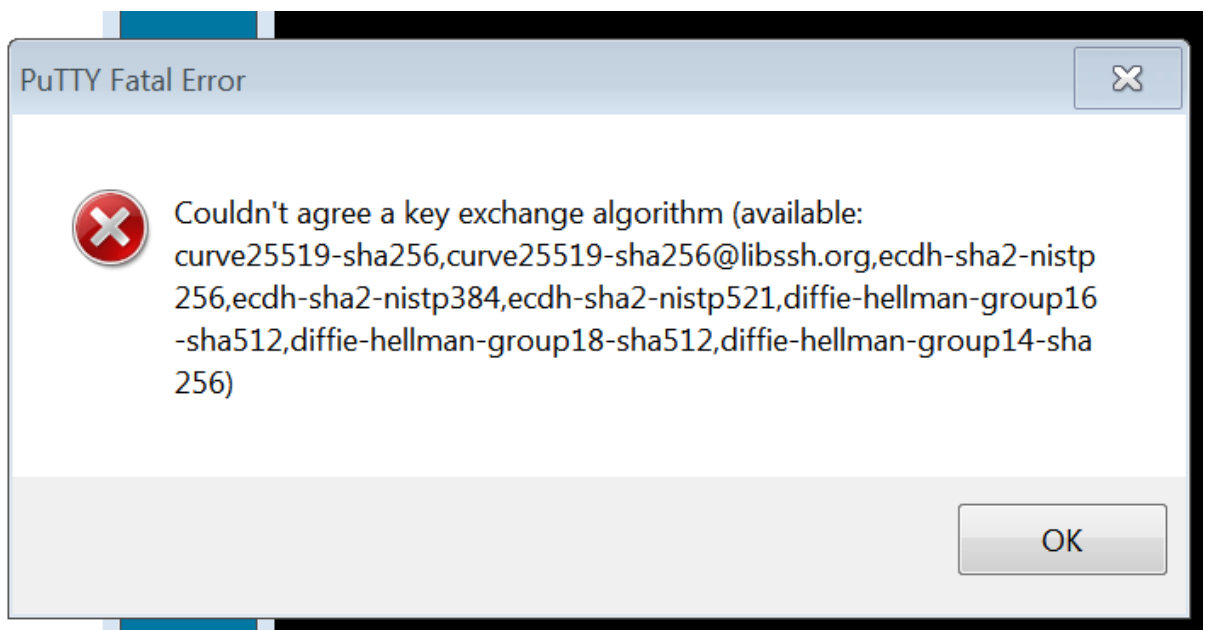
Figure: Putty out of date error message

The solution is to update ssh / putty.

15.7 Pi general troubleshooting

This is a page for troubleshooting the Pi hardware. See also:

- Pi equipment for information about the Raspberry Pi and charger that you will need

- Pi connectivity to troubleshoot the connection between your Pi, network and computer. This includes Wifi, and mDNS troubleshooting.

- Troubleshooting the EscapePod software setup, especially adding licenses.

15.7.1 Checking the basics

Some potential problems (root causes) to look for:

- The software image is not on the SD card

- The SD card is not in Pi

- The Pi is not powering on

Connect the Raspberry Pi to a hdmi monitor and power it on.

- Does it show a color gradient in a square when first poered on?

  - Yes: this a good sign. This is what should happen

  - No -> there is a problem with the power, the SD card, (or maybe the HDMI cable & monitor) Check that these are properly connected. Do you have the right SD card? Is it flashed?

- Does the Pi show a bunch of text as it boots?

  - Yes: this a good sign. This is what should happen

  - No -> there is a problem with the power, the SD card, (or maybe the HDMI cable & monitor) Check that these are properly connected. Do you have the right SD card? Is it flashed?

15.7.2 Pi Power supply issues

These happen when the power supply doesn't provide enough power, or the cable isn't good enough.

Some potential problems (root causes) to look for:

- Connect the Raspberry Pi to a HDMI monitor and power it on. Watch for a "lightning bolt" in the corner. If it regularly appears, the power supply or cable is the issue.

- Look for text that pops up on the display console with words "under-voltage".

- A common symptom is that the Raspberry Pi may stop responding; or a SSH connection may suddenly disconnect.

If you see those, you need to change to your power supply for the Raspberry Pi. See Pi equipment for a list of chargers.

What else to look for

In /var/log/sys log any lines like the following:

```
Dec 11 11:43:37 escapepod kernel: [  994.885094] rpi_firmware_get_throttled: 7 callbacks suppressed
Dec 11 11:43:37 escapepod kernel: [  994.885103] Voltage normalised (0x00000000)
```

```
Dec 11 11:44:03 escapepod kernel: [ 1021.092573] Under-voltage detected! (0x00050005)
Dec 11 11:44:07 escapepod kernel: [ 1025.124419] Voltage normalised (0x00000000)
```

In /var/log/kern.log any lines like the following:

```
Dec 11 11:22:58 escapepod kernel: [31302.855548] rpi_firmware_get_throttled: 5 callbacks suppressed
Dec 11 11:22:58 escapepod kernel: [31302.855559] Under-voltage detected! (0x00050005)
Dec 11 11:23:02 escapepod kernel: [31306.887532] Voltage normalised (0x00000000)
```

In /var/log/dmesg any lines like the following:

```
[   21.181843] kernel: Under-voltage detected! (0x00050005)
```

The following text that get displayed on the monitor

```
brcmfmac: brcmf_sdio_htclk: HT Avail request error:
```

If you see those, you need to change to your power supply for the Raspberry Pi. See Pi equipment for a list of chargers.

15.8 Software error codes

If something has gone wrong, an error will appear on Vector's face. The ones on this page are (usually) software.

914-915

- There was an issue with vic-engine. Vector should restart to normal operation on his own.

913

- There was an issue with vic-switchboard. This could happen if there was an error in BLE communication or if you entered too long of a string into the SDK. He should restart fine.

916-917

- There was an issue with vic-robot. He should restart just fine.

800

- There was an issue with vic-anim. He should restart just fine. If he doesn't, restart into recovery and clear user data.

850-852

- There was an issue with the cloud and/or serial number. A clear user data may help.

920-921

- There was an issue with vic-gateway or vic-gateway-cert. If he is stuck on this, you may need to clear user data.

923

- Vic-cloud has crashed. He should restart fine. If you got an 801-899 error before, this may be the server's fault.

16. Vector enhancement proposals

16.1 Vector Enhancement Proposals

Memos, cheekily named for Python's memo system. At the moment, I see these as proposals for changes to the software and files on a Vector. (Proposals for changes to the site or documentation, or build tools, etc should go elsewhere.)

This would be relevant for changes, esp substantial changes, that you might like many people to adopt.

File Format:

- I'm going to try to use markdown most often, but
- PDF/HTML export for normal human readers

Common elements, to make it easier to read and management them:

The first part is the markdown front matter: it begins and ends with --- and the lines inside contain YAML. This lets other tools extract the basics.

```
---
title: VEP123 - The name of the VEP (only a few words)
summary: An ptional description of the proposal, if the title is too short
authors:
    - Author Name
date: 2022-07-10
---
```

The title starts with "VEP" and a unique (serial) number. It is followed by a brief description or topic of the proposal. The other fields are self explanatory, and helps track the info

Other outline, organization:

- Description of the changes
- Some Design decisions
- Documentation
- Cavaets
- Status
- References
- Change history synopsis (this is for people)

16.2 VEP1 - Update-engine changes

Summary: Update-engine changes to make for unsigned, incremental updates; and to reduce the number of partitions modified.

Authors: Randall Maas

16.2.1 Description of the changes

Motivation: Building a new, experimental development release is possible but inconvenient with an stock update-engine:

1. The OTA update is very "heavy weight" -- it needs to update the boot and system file-system partitions (with a 200+MB file!) just to change a couple of files.

2. Creating an update an OTA file for OSKR bots and modified development bots is possible, but it too difficult for most people.

3. There is no way to blend changes from updates.

This enhancement changes the following to the update engine:

• Making the signing check of the manifest, update files optional

• Making the encryption of the OTA optional

• Allows replacing individual partitions, esp just the system file system

• Allows using a package manager or other tool to update the contents of the file system.

    • A package manager has been created to help with the process, and handle issues like file permissions. See VEP2 - Package management for modules on Vector

16.2.2 Some Design decisions

Using the `tar` utility is not used, and we found a small tool modify for out purposes instead. The busybox `tar` doesn't preserve permissions, which led to unexpected results and problems when updating executables.

16.2.3 Documentation

The `update-engine` looks the same to the rest of the system as before. (The OTA format, rules are described in the TRM...) It now accepts URLs with a `.vpkg` extension. If these are seen, the There is no at this time.

16.2.4 Cavaets

• this probably doesn't disable delta updates properly, so if a delta update were to be issued, we'd need a way to make sure it doesn't mess up the FS.

16.2.5 Status

The update

• I believe that Wire has used an earlier modified version

16.2.6 References

- The format and rules of the OTA files are in Chapter 32 of the Technical Reference Manual

- A reference implementation of these changes is on github in a gist

16.2.7 Change history synopsis

Date	Change
2020-8-30	Created
2020-12-6	Updated formatting, updated based on changes to 1.7.1, links to code style. Note: Changes in 1.7.1 made it possible to create -- especially encrypt and sign -- the files, a feature needed by the update-engine. Earlier versions of this proposal were mandated because of the inability to sign.

16.3 VEP2 - Package management for modules on Vector

Summary: Supports installing and uninstalling packages/modules on Vector

Authors: Randall Maas

16.3.1 Description of the changes

Motivation: We needed a way to package changes to a few files on an already deployed system, to ensure that the permissions are correct on the files (usually executable), and some restrictions/protections from screwing up system files. (Ie, don't force it to be unbootable)

This is a package manager that does those, and adds in a few extras:

- It allows modifying parts of a file, usually the version identifier of the system, so we know what we're working with.

- Lists the installed packages

- Can uninstall packages

- Can set the premissions for the files.

16.3.2 Some Design decisions

- The tool had to be small, and not hard to deploy

- The tool can't be compiled (we don't know how)

- It had to be based on tools already on Vector: python 2.7, and busybox based utils. Busybox supplies the shell, and tar... except tar doesn't support preserving permissions.

- It is preferred to separate out the package manager from the update-engine as much as possible, to make int more understandable and support testing.

- The package manager needs to protect the /mnt /dev directories; any access to these could irreversible destroy the bot. Since the packages are intended to be used by lots of people who won't (or won't be able to) examine the package for negative consequences we need to limit these unusual risks. If changes are needed to these the owner should ssh in and consciously make changes.

- It can't run arbitrary commands from the package file during installation; the packager can be run as root and this could irreversible destroy the bot. This has the same rationale as above.

It just installs the packages

This tool doesn't do everything that the other managers do:

- It doesn't check dependencies

- It doesn't download files

- It has minimal the pre-flight, post-flight scripts that are run.

The package download is handled by either the update-engine (and its line of control), or by scp command. The lack of dependency check is a benefit, as its hard to maintain, and it is rarely used correctly: maintainers tend to choose a dependency of "the latest version" (as of when the package was), negating its use.

16.3.3 Documentation

The documentation of the tool is included as part of it's tgz. This is a quick overview.

When a package is installed it creates another package taking a snapshot of those files already there. When the package is uninstalled this 2nd package is used to replace the newer files with the older ones. It doesn't delete any files that were added since or by the first package, so some extra stuff can accumulate, but that is far safer.

Creating a package. To create a package, lets call it demo, requires setting up the files sytem with the files, and the package manifest. The manifest says, among other things:

1. The package name, version, and other helpful paperwork info.

2. where to get the files from locally, and where they should be placed into filesystem deployed when deployed on a Vector.

3. The path to any files that should be modifed, and how. This is used to change the reported version string.

4. The permissions to set the files to

I've attached a really simple demo to demonstrate. To create a package unzip them, and then:

```
./vector-pkg.py create —pkg=demo
```

That will create .vpkg file — a gzip'd tar file with a specific layout. From here everything has to be on a Vector.

Installation. To install the vpkg:

```
./vector-pkg.py install —pkg=demo-1.vpkg
```

Uninstall. To uninstall the vpkg later

```
./vector-pkg.py uninstall —pkg=demo-1
```

Adding a restart step after installation

If `vector-pkg` is by called the modified `update-engine` it can tell it to restart the application or reboot the operating system after the package has installed. This is done using a `restart_type=` key in the `[META]` section.

There are four different values to say how to restart after applying the package. To simply restart Vector's application:

```
[META]
restart_type=restart
```

To restart Vector's application, but silently -- that is, not play the *InitialWakeUp* animation:

```
[META]
restart_type=maintenance-restart
```

To reboot the operating system after the package has installed:

```
[META]
restart_type=reboot
```

The following will reboot the operating system using a "maintenance reboot" so that the *InitialWakeUp* animation is not played:

```
[META]
restart_type=maintenance-reboot
```

16.3.4 Cavaets

16.3.5 Status

Not tested by others yet. Once the bugs are shaken out, the update-engine can be tweaked. When the update-engine gets a URL with ".vpkg" (such as from the BLE app) at the end, it downloads it, and then pass it to the package manager.

16.3.6 References

The files are on github https://github.com/randym32/Anki.Vector.PackageInstaller

16.3.7 Change history synopsis

Date	Change
2020-8-30	Created
2020-12-3	Added how to restart after installing a package
2020-12-5	Removed unsafe features, changed how to restart after installing a package

16.4 VEP3 - Developer configurations for robot

Summary: Change robot configurations for

This is a stub proposal for a VPKG with many common configuration setttings for a Developer-tinkering bots.

- configure many the servers to use to main production server

- configure servers to use local servers (logging)

- Customize: CPU, Heat, Display settings

Not sure if these can be downloaded via Bluetooth LE

- server configuration

- local preferences

16.4.1 References

16.4.2 Change history synopsis

16.5 VEP4 - Logging

Summary: Changes to the logging scripts, configuration to local servers. This is to modify the servers to use for the logging, crash dumps, and similar. Events/logs will no longer be sent to AWS, or backtrace.io.

Authors: Randall Maas

This a draft proposal (to be filled in) on how to modify Vector config files and scripts to send logging and crash dumps to a server of our choosing.

- Logging

- Trace information

- Server

- Settings

- DAS optin/optout

Replace `/anki/bin/vic-log-upload`

- moving aside, `/anki/bin/vic-log-upload` since it just does AWS, S3://

- put in something that can contact our local server

Files:

- server_config.json

- log uploader

16.5.1 References

See How change where Vector sends the logs for example changes to the configuration files.

16.5.2 Change history synopsis

Date	Change
2020-11-17	Created