# Syringenator

Generated by Doxygen 1.8.16

# 1 README

University of Washington

Winter 2019

Syringe Robot

## 1.1 Development Team Vulcan

### Authors

| Name | Email |
|---|---|
| Alex Boyle | boylea4@uw.edu |
| Ammon Dodson | ammon0@uw.edu |
| Alex Marlow | alexmarlow117@gmail.com |
| Jake McKenzie | jake314@uw.edu |
| Brooke Stevenson | brooks04@uw.edu |

### Copyright



## 1.2 Project Pages

- The Github Repo
- Documentation Website

## 1.3 Communication

### 1.3.1 Don't Clobber Other People's Work

Since we're all working in the same space it is important to be courteous. Pretty much this comes down to not overwriting other people's work. If there is some real need to change something that already exists there should be a discussion between everyone involved.

#### 1.3.2 Comment Your Work

Not everything will be obvious to everyone else. Write a paragraph for every non-trivial function. Write a detailed explanation any time you want to get clever with the code. Always put your name or initials on larger comments and blocks of code that you have written. That way it's easy to know who to talk to if there are questions.

## 1.4 Using Git

Git is a command-line tool for managing source code. Github is an on-line service that provides git remotes. A git remote is a remote copy of a git repository. Multiple people work in the same repository through the use of a single remote. The trick is to manage version conflicts intelligently.

Each team member should periodically merge master into their own branch to ensure that we are synced up. The master branch should only ever have merge commits and working code. I will try to enforce this with Github so that we don't make a mess. –ABD

#### 1.4.1 Work in Your Own Branch

Each team member should create their own branch to work in. You may make as many branches as you like, just make sure you have one. You can create branches on the command line with:

```
$ git branch <branch-name>
```

To switch to your branch do:

```
$ git checkout <branch-name>
```

#### 1.4.2 Commit Your Work

Commits are a permanent record of your work. They should be as small and purpose-driven as possible. Think: "can I write a couple lines that explains what I did?" To check for uncommitted changes, or check your status in general do:

```
$ git status
On branch ammon
Your branch is up-to-date with 'github/ammon'.  <- this is the remote
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout - <file>..." to discard changes in working directory)
    modified:   README.md
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    docs/autotoc_md6.html
    latex/autotoc_md6.tex
no changes added to commit (use "git add" and/or "git commit -a")
```

You make a commit in two steps: first you stage the changed files that you want to include in this next commit.

```
$ git add <filename> <anotherfile>
```

Once you have staged a bunch of changes you can check your status again:

```
$ git status
On branch ammon
Your branch is up-to-date with 'github/ammon'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
    modified:   README.md
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout - <file>..." to discard changes in working directory)
    modified:   Makefile
    deleted:    refman.pdf
```

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    docs/autotoc_md7.html
    latex/autotoc_md7.tex
```

Once you are satisfied with what is currently staged you finish the commit by doing:
`$ git commit`

Git will automatically open a text editor where you can describe what the changes are. Make this a meaningful message since it will be the only thing that distinguishes this commit from hundreds of others.

You can also do:
`$ git commit -m "<commit message>"`

(-m is shorthand for –messages command which tells other collaborators (and your future self) the nature of the change you just made. –Jake

#### 1.4.3 Merge All the Latest Changes

The magic of git is being able to merge conflicting changes. Before you share your changes (pushing), you must pull the latest changes and merge them with yours. First pull the master branch:
`$ git pull origin master`

You will need to enter your password and git will tell you if there have been any changes. Git will attempt to merge the master branch into yours. If there are any conflicts it will tell you. Git will rewrite your files to include both versions of the conflicting code. To see which files are in conflict do:
`$ git status`

You have to open those files, find, and fix the conflicting versions. Once you think you are done, rebuild and test all the code. Look for any new errors and fix them. Once you are satisfied that the merge has been completed successfully add and commit your changes as usual.

#### 1.4.4 Push Your Branch

Pushing your work to the remote allows everyone else to see it. You should merge master before pushing. To push do:
`$ git push origin <your-branch>`

#### 1.4.5 Make a Pull Request

The master branch is where we integrate all the changes everyone is making. This is done through "pull requests". A pull request is a way for everyone to see and comment on new code. It will also allow us to only make merge commits to the master branch. If we work this way the master branch will always be clean and there will be less errors, lost work, and wasted time.

#### 1.4.6 What not to do

- **Don't commit directly to master**. I've tried to setup Github to make this difficult or impossible, but in any case that it isn't protected properly nobody should be trying this anyway.
- **Don't –force** Read your error messages, they are usually very helpful. The force tag overwrites history and can easily erase work already done. If git complains there is a reason for it.

# 2 Calibration

## 2.1 Coordinate Systems

This robot, of necessity uses multiple sets of coordinates.

### 2.1.1 Image Cartesian

This coordinate system is used to locate pixels and distance measurements in the images generated from the camera. It consists of a positive integer tuple horizontal and vertical. Its axes are at right angles, and its origin is in the upper left corner of the image. Its values are always positive and its units are pixels.

We may also consider the camera's depth value as the third member of the image coordinates. Its units should be meters.

### 2.1.2 Floor Cartesian

This coordinate system is used to locate targets around the robot. It consists of a signed integer tuple fore-aft and port-starboard. Positive values are forward and starboard. Its axes are at right angles and its origin is at the front edge of the robot. Its units of length are millimeters.

### 2.1.3 Arm Cylindrical

This coordinate system is used to locate targets around the xArm. It consists of an unsigned integer tuple azimuth and range. Its origin is at the level of the floor and directly below the xArm axis of rotation. Its units are those convenient for the use of the arm, and its range of values is recorded in constants.in

# 3 Installations

## 3.1 librealsense

from github

### 3.1.1 Downloads

Update the system
```
sudo apt update
```

get the kernel headers so that we can compile new things
```
sudo apt install raspberrypi-kernel-headers
```

make sure that raspberrypi-kernel and raspberrypi-bootloader are at the latest versions

install git and other build tools
```
sudo apt install git build-essential -y
```

get the latest librealsense
```
git clone -depth 1 https://github.com/IntelRealSense/librealsense.git
```

Install Intel Realsense permission scripts located in librealsense source directory:
```
sudo cp config/99-realsense-libusb.rules /etc/udev/rules.d/
sudo udevadm control -reload-rules && udevadm trigger
```

get the source for the current kernel make sure version numbers match apt-cache
```
wget https://github.com/raspberrypi/linux/archive/raspberrypi-kernel_1.20161215-1.tar.gz
```

extract it
```
tar -xzf raspberrypi-kernel_1.20161215-1.tar.gz
```

#### 3.1.2 Kernel source patching

```
`LINUX_BRANCH=$(uname -r)
# Construct branch name from distribution codename {xenial,bionic,..} and kernel version
ubuntu_codename=`. /etc/os-release; echo ${UBUNTU_CODENAME/*, /}`
if [ -z "${ubuntu_codename}" ];
then
    # Trusty Tahr shall use xenial code base
    ubuntu_codename="xenial"
    retpoline_retrofit=1
fi
kernel_branch=$(choose_kernel_branch ${LINUX_BRANCH} ${ubuntu_codename})
kernel_name="ubuntu-${ubuntu_codename}-$kernel_branch"`
```

#### 3.1.3 Kernel Configuration

Load the kernel configuration module

```
sudo modprobe configs
```

get a copy of the current kernel configuration

```
cp /proc/config.gz ./
```

decompress it

```
gunzip config.gz
```

put the configuration in the source tree

```
mv config linux-raspberrypi-kernel_1.20161215-1/.config
```

In the kernel directory update the config

```
make silentoldconfig
```

#### 3.1.4 Build librealsense

```
mkdir build && cd build
```

The default build is set to produce the core shared object and unit-tests binaries in Debug mode.

```
cmake ../
```

- -DCMAKE_BUILD_TYPE=Release to build with optimizations.
- -DBUILD_EXAMPLES=true Builds librealsense along with the demos and tutorials
- -DBUILD_GRAPHICAL_EXAMPLES=false For systems without OpenGL or X11 build only textual examples

Command used:
```
cmake -DCMAKE_BUILD_TYPE=Release -DBUILD_PYTHON_BINDINGS=bool:true
       -DPYTHON_EXECUTABLE=/home/big/Desktop/Syringenator/pyVirtEnv/syringenator/bin/python ../
```

cmake returns:
```
- Info: REALSENSE_VERSION_STRING=2.18.0
- Setting Unix configurations
- Checking internet connection...
- Internet connection identified, enabling BUILD_WITH_TM2
- Found PythonInterp: /home/big/Desktop/Syringenator/pyVirtEnv/syringenator/bin/python (found version "2.7.12")
- Found PythonLibs: /usr/lib/arm-linux-gnueabihf/libpython2.7.so
- pybind11 v2.2.1
- Performing Test HAS_FLTO
- Performing Test HAS_FLTO - Success
- LTO enabled
- Could NOT find Vulkan (missing: VULKAN_LIBRARY VULKAN_INCLUDE_DIR)
- Using X11 for window creation
- Building with TM2
- ------------------------------------------------
- T265 Product versions:
- - HOST 0.19.3.1505 (Default from versions.cmake)
- - Remote FW 0.0.18.4577 (Default from versions.cmake)
- - Remote CENTRAL APP 2.0.19.271 (Default from versions.cmake)
- - Remote CENTRAL BL 1.0.1.112 (Default from versions.cmake)
- ------------------------------------------------------------------------------------------------------------
- Downloading FW 0.0.18.4577 from
       'http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/target/0.0.18.4577/target-0.0.18.4577.mvcmd'
- Converting FW version 0.0.18.4577 from target.mvcmd to
       /home/big/Desktop/librealsense/third-party/libtm/libtm/src/fw.h
- Downloading Central App 2.0.19.271 from
       'http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/app/2.0.19.271/central_app-2.0.19.271.bin'
- Converting Central App version 2.0.19.271 from central_app.bin to
       /home/big/Desktop/librealsense/third-party/libtm/libtm/src/CentralAppFw.h
- Downloading Central BL 1.0.1.112 from
       'http://realsense-hw-public.s3.amazonaws.com/Releases/TM2/FW/bl/1.0.1.112/central_bl-1.0.1.112.bin'
- Converting Central BL version 1.0.1.112 from central_bl.bin to
       /home/big/Desktop/librealsense/third-party/libtm/libtm/src/CentralBlFw.h
- ------------------------------------------------------------------------------------------------------------
- Building libtm project on , LIBTM version [0.19.3.1505], API version [10.0], branch [master], FW
       [0.0.18.4577], Central APP [2.0.19.271], Central BL [1.0.1.112]
- Creating version file /home/big/Desktop/librealsense/third-party/libtm/libtm/src/Version.h
- Building project tm as STATIC library lib
- ------------------------------------------------------------------------------------------------------------
- Building all projects of libtm_samples
- Building project libtm_util
- ------------------------------------------------
- CMake Done
- ------------------------------------------------
- Configuring done
- Generating done
- Build files have been written to: /home/big/Desktop/librealsense/build
```

Recompile and install librealsense binaries:
```
sudo make uninstall && make clean && make && sudo make install
```

## 3.2 OpenCV

We used this `tutorial` with some modifications.

### 3.2.1 Dependencies

The tutorial's atlas installation is insufficient resulting in:
```
- Could NOT find Atlas (missing: Atlas_CLAPACK_INCLUDE_DIR)
```

Refering to `issue #10442` I did:
```
sudo apt install liblapacke-dev
```

#### 3.2.2 Python Virtual Environment

I wanted to include the python virtual environment in the git repo so that it can be used by anyone. I am not sure if this is the prefered way to share virtual environments. We also won't lose it if the pi has to be rebuilt. so the .bashrc script reads:

```
export WORKON_HOME=$HOME/Desktop/Syringenator/src/pi/pyVirtEnv
source /usr/local/bin/virtualenvwrapper.sh
```

#### 3.2.3 cmake

the cmake step then needs to be modified to acommodate:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D INSTALL_C_EXAMPLES=OFF \
    -D OPENCV_EXTRA_MODULES_PATH= /Desktop/opencv_contrib-4.0.1/modules \
    -D PYTHON_EXECUTABLE= /Desktop/Syringenator/pyVirtEnv/syringenator/bin/python \
    -D BUILD_EXAMPLES=ON \
    -D WITH_OPENMP=ON ..
```

cmake reports:

```
- Looking for ccache - not found
- FP16 is not supported by C++ compiler
- Found ZLIB: /usr/lib/arm-linux-gnueabihf/libz.so (found suitable version "1.2.8", minimum required is "1.2.3")

- Found ZLIB: /usr/lib/arm-linux-gnueabihf/libz.so (found version "1.2.8")
- Checking for module 'gstreamer-base-1.0'
-   No package 'gstreamer-base-1.0' found
- Checking for module 'gstreamer-video-1.0'
-   No package 'gstreamer-video-1.0' found
- Checking for module 'gstreamer-app-1.0'
-   No package 'gstreamer-app-1.0' found
- Checking for module 'gstreamer-riff-1.0'
-   No package 'gstreamer-riff-1.0' found
- Checking for module 'gstreamer-pbutils-1.0'
-   No package 'gstreamer-pbutils-1.0' found
- Checking for module 'gstreamer-base-0.10'
-   No package 'gstreamer-base-0.10' found
- Checking for module 'gstreamer-video-0.10'
-   No package 'gstreamer-video-0.10' found
- Checking for module 'gstreamer-app-0.10'
-   No package 'gstreamer-app-0.10' found
- Checking for module 'gstreamer-riff-0.10'
-   No package 'gstreamer-riff-0.10' found
- Checking for module 'gstreamer-pbutils-0.10'
-   No package 'gstreamer-pbutils-0.10' found
- Checking for module 'libdc1394-2'
-   No package 'libdc1394-2' found
- Checking for module 'libdc1394'
-   No package 'libdc1394' found
- Looking for linux/videodev2.h
- Looking for linux/videodev2.h - found
- Looking for sys/videoio.h
- Looking for sys/videoio.h - not found
- Checking for module 'libavresample'
-   No package 'libavresample' found
- LAPACK(Atlas): LAPACK_LIBRARIES: /usr/lib/liblapack.so;/usr/lib/libcblas.so;/usr/lib/libatlas.so
- LAPACK(Atlas): Support is enabled.
- Could NOT find JNI (missing: JAVA_INCLUDE_PATH JAVA_INCLUDE_PATH2 JAVA_AWT_INCLUDE_PATH)
- Could NOT find Pylint (missing: PYLINT_EXECUTABLE)
- Could NOT find Flake8 (missing: FLAKE8_EXECUTABLE)
- VTK is not found. Please set -DVTK_DIR in CMake to VTK build directory, or to VTK install subdirectory with
       VTKConfig.cmake file
- OpenCV Python: during development append to PYTHONPATH: /home/big/Desktop/opencv-4.0.1/build/python_loader
- Caffe:   NO
- Protobuf:   NO
- Glog:   NO
- freetype2:   YES
- harfbuzz:    YES
- Could NOT find HDF5 (missing: HDF5_LIBRARIES HDF5_INCLUDE_DIRS) (found version "")
- Module opencv_ovis disabled because OGRE3D was not found
- No preference for use of exported gflags CMake configuration set, and no hints for include/library directories
       provided. Defaulting to preferring an installed/exported gflags CMake configuration if available.
```

```
- Failed to find installed gflags CMake configuration, searching for gflags build directories exported with
      CMake.
- Failed to find gflags - Failed to find an installed/exported CMake configuration for gflags, will perform
      search for installed gflags components.
- Failed to find gflags - Could not find gflags include directory, set GFLAGS_INCLUDE_DIR to directory
      containing gflags/gflags.h
- Failed to find glog - Could not find glog include directory, set GLOG_INCLUDE_DIR to directory containing
      glog/logging.h
- Module opencv_sfm disabled because the following dependencies are not found: Eigen Glog/Gflags
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.sse2.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.sse3.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.ssse3.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.sse4_1.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.sse4_2.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.avx.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.fp16.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin128.avx2.cpp
- Excluding from source files list: <BUILD>/modules/core/test/test_intrin256.avx2.cpp
- Excluding from source files list: modules/imgproc/src/corner.avx.cpp
- Excluding from source files list: modules/imgproc/src/filter.avx2.cpp
- Excluding from source files list: modules/imgproc/src/imgwarp.avx2.cpp
- Excluding from source files list: modules/imgproc/src/imgwarp.sse4_1.cpp
- Excluding from source files list: modules/imgproc/src/resize.avx2.cpp
- Excluding from source files list: modules/imgproc/src/resize.sse4_1.cpp
- Excluding from source files list: <BUILD>/modules/dnn/layers/layers_common.avx.cpp
- Excluding from source files list: <BUILD>/modules/dnn/layers/layers_common.avx2.cpp
- Excluding from source files list: <BUILD>/modules/dnn/layers/layers_common.avx512_skx.cpp
- freetype2:   YES
- harfbuzz:    YES
- Excluding from source files list: modules/features2d/src/fast.avx2.cpp
- Checking for modules 'tesseract;lept'
-   No package 'tesseract' found
-   No package 'lept' found
- Tesseract:   NO
- Excluding from source files list: modules/calib3d/src/undistort.avx2.cpp
- OpenCL samples are skipped: OpenCL SDK is required
-
- General configuration for OpenCV 4.0.1 =====================================
-   Version control:               unknown
-
-   Extra modules:
-     Location (extra):            /home/big/Desktop/opencv_contrib-4.0.1/modules
-     Version control (extra):     unknown
-
-   Platform:
-     Timestamp:                   2019-02-14T22:20:14Z
-     Host:                        Linux 4.4.38-v7+ armv7l
-     CMake:                       3.13.3
-     CMake generator:             Unix Makefiles
-     CMake build tool:            /usr/bin/make
-     Configuration:               RELEASE
-
-   CPU/HW features:
-     Baseline:
-       requested:                 DETECT
-       disabled:                  VFPV3 NEON
-
-   C/C++:
-     Built as dynamic libs?:      YES
-     C++ Compiler:                /usr/bin/c++  (ver 5.5.0)
-     C++ flags (Release):         -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor
       -Werror=address -Werror=sequence-point -Wformat -Werror=format-security -Wmissing-declarations -Wundef
       -Winit-self -Wpointer-arith -Wshadow -Wsign-promo -Wuninitialized -Winit-self -Wno-narrowing
       -Wno-delete-non-virtual-dtor -Wno-comment -fdiagnostics-show-option -pthread -fomit-frame-pointer
       -ffunction-sections -fdata-sections  -mfp16-format=ieee -fvisibility=hidden -fvisibility-inlines-hidden
       -fopenmp -O3 -DNDEBUG  -DNDEBUG
-     C++ flags (Debug):           -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor
       -Werror=address -Werror=sequence-point -Wformat -Werror=format-security -Wmissing-declarations -Wundef
       -Winit-self -Wpointer-arith -Wshadow -Wsign-promo -Wuninitialized -Winit-self -Wno-narrowing
       -Wno-delete-non-virtual-dtor -Wno-comment -fdiagnostics-show-option -pthread -fomit-frame-pointer
       -ffunction-sections -fdata-sections  -mfp16-format=ieee -fvisibility=hidden -fvisibility-inlines-hidden
       -fopenmp -g  -O0 -DDEBUG -D_DEBUG
-     C Compiler:                  /usr/bin/cc
-     C flags (Release):           -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor
       -Werror=address -Werror=sequence-point -Wformat -Werror=format-security -Wmissing-declarations
       -Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized
       -Winit-self -Wno-narrowing -Wno-comment -fdiagnostics-show-option -pthread -fomit-frame-pointer
       -ffunction-sections -fdata-sections  -mfp16-format=ieee -fvisibility=hidden -fopenmp -O3 -DNDEBUG
       -DNDEBUG
-     C flags (Debug):             -fsigned-char -W -Wall -Werror=return-type -Werror=non-virtual-dtor
       -Werror=address -Werror=sequence-point -Wformat -Werror=format-security -Wmissing-declarations
```

```
       -Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized
       -Winit-self -Wno-narrowing -Wno-comment -fdiagnostics-show-option -pthread -fomit-frame-pointer
       -ffunction-sections -fdata-sections  -mfp16-format=ieee -fvisibility=hidden -fopenmp -g  -O0 -DDEBUG
       -D_DEBUG
-      Linker flags (Release):
-      Linker flags (Debug):
-      ccache:                   NO
-      Precompiled headers:      YES
-      Extra dependencies:       dl m pthread rt
-      3rdparty dependencies:
-
-    OpenCV modules:
-      To be built:              aruco bgsegm bioinspired calib3d ccalib core datasets dnn dnn_objdetect dpm
       face features2d flann freetype fuzzy gapi hfs highgui img_hash imgcodecs imgproc java_bindings_generator
       line_descriptor ml objdetect optflow phase_unwrapping photo plot python2 python_bindings_generator reg
       rgbd saliency shape stereo stitching structured_light superres surface_matching text tracking ts video
       videoio videostab xfeatures2d ximgproc xobjdetect xphoto
-      Disabled:                 world
-      Disabled by dependency:   -
-      Unavailable:              cnn_3dobj cudaarithm cudabgsegm cudacodec cudafeatures2d cudafilters
       cudaimgproc cudalegacy cudaobjdetect cudaoptflow cudastereo cudawarping cudev cvv hdf java js matlab ovis
       python3 sfm viz
-      Applications:             tests perf_tests examples apps
-      Documentation:            NO
-      Non-free algorithms:      NO
-
-    GUI:
-      GTK+:                     YES (ver 3.18.9)
-        GThread :               YES (ver 2.48.2)
-        GtkGlExt:               NO
-      VTK support:              NO
-
-    Media I/O:
-      ZLib:                     /usr/lib/arm-linux-gnueabihf/libz.so (ver 1.2.8)
-      JPEG:                     /usr/lib/arm-linux-gnueabihf/libjpeg.so (ver 80)
-      WEBP:                     build (ver encoder: 0x020e)
-      PNG:                      /usr/lib/arm-linux-gnueabihf/libpng.so (ver 1.2.54)
-      TIFF:                     /usr/lib/arm-linux-gnueabihf/libtiff.so (ver 42 / 4.0.6)
-      JPEG 2000:                /usr/lib/arm-linux-gnueabihf/libjasper.so (ver 1.900.1)
-      OpenEXR:                  build (ver 1.7.1)
-      HDR:                      YES
-      SUNRASTER:                YES
-      PXM:                      YES
-      PFM:                      YES
-
-    Video I/O:
-      DC1394:                   NO
-      FFMPEG:                   YES
-        avcodec:                YES (ver 57.64.100)
-        avformat:               YES (ver 57.56.100)
-        avutil:                 YES (ver 55.34.100)
-        swscale:                YES (ver 4.2.100)
-        avresample:             NO
-      GStreamer:                NO
-      v4l/v4l2:                 linux/videodev2.h
-
-    Parallel framework:         OpenMP
-
-    Trace:                      YES (built-in)
-
-    Other third-party libraries:
-      Lapack:                   YES (/usr/lib/liblapack.so /usr/lib/libcblas.so /usr/lib/libatlas.so)
-      Eigen:                    NO
-      Custom HAL:               YES (carotene (ver 0.0.1))
-      Protobuf:                 build (3.5.1)
-
-    OpenCL:                     YES (no extra features)
-      Include path:             /home/big/Desktop/opencv-4.0.1/3rdparty/include/opencl/1.2
-      Link libraries:           Dynamic load
-
-    Python 2:
-      Interpreter:              /home/big/Desktop/Syringenator/src/pi/pyVirtEnv/syringenator/bin/python (ver
       2.7.12)
-      Libraries:                /usr/lib/arm-linux-gnueabihf/libpython2.7.so (ver 2.7.12)
-      numpy:
       /home/big/Desktop/Syringenator/src/pi/pyVirtEnv/syringenator/lib/python2.7/site-packages/numpy/core/include
       (ver 1.16.1)
-      install path:             lib/python2.7/site-packages/cv2/python-2.7
-
-    Python (for build):         /home/big/Desktop/Syringenator/src/pi/pyVirtEnv/syringenator/bin/python
-
```

```
-   Java:
-     ant:                         NO
-     JNI:                         NO
-     Java wrappers:               NO
-     Java tests:                  NO
-
-   Install to:                    /usr/local
- -----------------------------------------------
-
- Configuring done
- Generating done
- Build files have been written to: /home/big/Desktop/opencv-4.0.1/build
```

# 4 Training

This file contains refrences for totorials and programs used in training and running Yolov3-tiny.

## 4.1 Refrences

- OpenCV
- Yolo
- Training Yolo

# 5 Todo List

**File Syringenator.py**

how do we initialize the robot run? a button press? –ABD

**Member Syringenator.returnToLine ()**

do we need to check that we actually returned? how do we recover if dead reckoning fails? –ABD

# 6 Module Index

## 6.1 Modules

Here is a list of all modules:



# 7 Namespace Index

## 7.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

# 8 Class Index

## 8.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:



# 9 File Index

## 9.1 File List

Here is a list of all documented files with brief descriptions:

# 10 Module Documentation

## 10.1 pyimagesearch Code

This code was taken and refactored from `pyimagesearch`

### Functions

- def Syringenator.rescale (detection)

  *Rescale the bounding box coordinates scale the bounding box coordinates back relative to the size of the image, keeping in mind that YOLO actually returns the center (x, y)-coordinates of the bounding box followed by the boxes' width and height.*
- def Syringenator.extractTargets (dataIn)

  *process the OpenCV output to generate actionable targets*

### 10.1.1 Detailed Description

This code was taken and refactored from `pyimagesearch`

### 10.1.2 Function Documentation

#### 10.1.2.1 extractTargets()

```
def Syringenator.extractTargets (
              dataIn )
```

process the OpenCV output to generate actionable targets

**Parameters**

| | |
|---|---|
| *dataIn* | the data from OpenCV |

#### 10.1.2.2 rescale()

```
def Syringenator.rescale (
              detection )
```

Rescale the bounding box coordinates scale the bounding box coordinates back relative to the size of the image, keeping in mind that YOLO actually returns the center (x, y)-coordinates of the bounding box followed by the boxes' width and height.

#### Returns

(centerX, centerY, width, height) for the target

# 11 Namespace Documentation

## 11.1 Syringenator Namespace Reference

The top-level Pi program.

#### Classes

- class Camera

  *A class to wrap the Camera.*
- class Log

  *A class to wrap the logging functions.*
- class NeuralNet

  *A class to wrap the DNN.*
- class Target

  *A class to contain everything we know about an aquired target.*

#### Functions

- def rescale (detection)

  *Rescale the bounding box coordinates scale the bounding box coordinates back relative to the size of the image, keeping in mind that YOLO actually returns the center (x, y)-coordinates of the bounding box followed by the boxes' width and height.*
- def extractTargets (dataIn)

  *process the OpenCV output to generate actionable targets*
- def imageCart2floorCart (t)

  *Derive floor position from image data.*
- def pixelRadius (t)

  *Determine how far a target is outside of the pickup radius.*
- def floorCart2armCylinder (x, y)

  *Derive cylindrical coordinates, centered on the arm from cartesian coordinates centered on the camera.*
- def floorCart2steer (x, y)

  *Generate a steering azimuth from floor cartesian.*
- def scan (cam, net)

  *A routine to take a picture and report back the closest target The Computer vision routine must be able to handle multiple targets in the image.*
- def canBePicked (t)

  *A routine to determine if the target is in position to be picked up.*
- def approach (t)

  *Move the robot closer to the given target.*
- def avoid ()

*avoid an obstacle*

- def pickUp (t)

  *Attempt to pickup and dispose the target.*

- def returnToLine ()

  *signl the arduino to return to the line.*

- def lineFollow ()

  *Follow the line.*

### Variables

- bool DEBUG_CAPTURE = False

  *Display an image once it's been captured.*

- bool DEBUG_AQUISITION = False

  *Draw centers and bounding boxes on the image.*

- bool DEBUG_APPROACH = False

  *Report on turn and forward values during approach.*

- bool DEBUG_TRANSFORM = False

  *Report azimuth and range values during pickUp.*

- bool DEBUG_ORIENTATION = False

  *display processed image for hand orientation*

- bool DEBUG_TIMING = False

  *report on yolo timing*

- bool CAL_X = False

  *Interrupt the normal loop and take X calibration photos.*

- bool CAL_Y = False

  *Interrupt the normal loop and take Y calibration photos.*

- bool DISABLE_WHEELS = False

  *Disable any wheel commands.*

- bool DISABLE_LINE_FOLLOW = False

  *robot only picks and returns to start position*

- int **FRAME_RATE** = 30
- int **IMG_WIDTH** = 640
- int **IMG_HEIGHT** = 480
- float **FOV_X** = 63.4
- float **FOV_Y** = 40.4
- int **CONFIDENCE** = .5
- int **NMS_THRESHOLD** = .1
- int **XPIX2LEN** = 10/6.48
- int **YPIX2LEN** = 10/6.0
- bool onTheLine = True

  *boolean indicating whether we are on the line*

- bool obstacle = False

  *boolean indicating that we have detected an obstacle*

- bool **inWindow** = False
- target = None

  *The currently aquired target.*

- int pickUpCount = 0

*the number of times a pickup has been attempted*

- int PICKUP_LIMIT = 2

  *the maximum number of times to attempt a pickup*
- **log** = Log()
- **camera** = Camera()
- **neuralNet** = NeuralNet()
- **comPort** = SySerial.ComPort()
- **status** = comPort.status()
- int **SCALE** = 4
- **image** = camera.capture()
- **crop**
- **bigger** = cv2.resize(src=crop, dsize=(0,0), fx=SCALE, fy=SCALE)
- tuple **biggerDim**
- tuple **centerX** = (IMG_WIDTH/2-constants.PICKUP_X_MIN+(constants.CAL_CAM_X_OFFSET))∗SCALE
- int **CROP_Y_MIN** = 400

#### 11.1.1 Detailed Description

The top-level Pi program.

#### 11.1.2 Function Documentation

##### 11.1.2.1 approach()

```
def Syringenator.approach (
              t )
```

Move the robot closer to the given target.

The moveCloser() routine attempts to aproach the target by relatively small increments. Because the move routines may be interrupted by the obstacle avoidance ISRs and the risk of jambing the wheels etc. we cannot expect to be able to approch successfully on the first try. Hence moveCloser() should only move a relatively short distance before exiting to allow another loop through the scan cycle.

Should we spend effort trying to avoid running over decoys here?

This routine is likely where we will have the most issues. –ABD

**Parameters**

| | |
|---|---|
| *t* | a Target object containing the location of the target to be approched |

**Returns**

None

#### 11.1.2.2 avoid()

```
def Syringenator.avoid ( )
```

avoid an obstacle

**Returns**

None

#### 11.1.2.3 canBePicked()

```
def Syringenator.canBePicked (
              t )
```

A routine to determine if the target is in position to be picked up.

Calculates whether the center of the target bounding box is in the pickup area.

**Returns**

a boolean

#### 11.1.2.4 floorCart2armCylinder()

```
def Syringenator.floorCart2armCylinder (
              x,
              y )
```

Derive cylindrical coordinates, centered on the arm from cartesian coordinates centered on the camera.

**Parameters**

| | |
|---|---|
| *x* | the x-value of the point of interest on the floor |
| *y* | the y-value of the point of interest on the floor |

**Returns**

a tuple (Azimuth, Range)

#### 11.1.2.5 floorCart2steer()

```
def Syringenator.floorCart2steer (
              x,
              y )
```

Generate a steering azimuth from floor cartesian.

**Parameters**

| | |
|---|---|
| *(x,y)* | a floor cartesian tuple |

**Returns**

an steering angle in degrees.

#### 11.1.2.6 imageCart2floorCart()

```
def Syringenator.imageCart2floorCart (
              t )
```

Derive floor position from image data.

**Parameters**

| | |
|---|---|
| *t* | a Target object |

**Returns**

a tuple (x, y) the coordinates on the floor in mm

#### 11.1.2.7 lineFollow()

```
def Syringenator.lineFollow ( )
```

Follow the line.

this routine simply signals the arduino to execute its lineFollow() routine

**Returns**

None

#### 11.1.2.8 pickUp()

```
def Syringenator.pickUp (
              t )
```

Attempt to pickup and dispose the target.

This routine must determine orientation of the target. If this is not done by some OpenCV magic we can attempt it here using the raw image data and the bounding box.

Divide the longer dimension of the bounding box by some constant divisor. Scan along each of those raster lines twice. On the first pass calculate an average brightness (RGB values can be summed). The second pass will pick out points of greatest brightness. Find the centers of clustered bright pixeles. We now have a set of points in cartesian space. Have Jake find the slope of the line of best fit.

The center can be estimated as the center of the bounding box, or the center of the points, the mean of both, etc.

Once the values for x, y, and m have been determined they will have to pass through a calibration transform to determine the arm a, r, o values. –ABD

**Parameters**

| | |
|---|---|
| *t* | a Target object containing the raw bitmap data |

**Returns**

None

#### 11.1.2.9 pixelRadius()

```
def Syringenator.pixelRadius (
              t )
```

Determine how far a target is outside of the pickup radius.

**Parameters**

| | |
|---|---|
| *t* | a Target object |

**Returns**

a distance in some unit

#### 11.1.2.10 returnToLine()

```
def Syringenator.returnToLine ( )
```

signl the arduino to return to the line.

**Todo** do we need to check that we actually returned? how do we recover if dead reckoning fails? –ABD

We disscussed the possibility of a timer on lineFollow(), that if the line has not been detected recently then we know we are off track and must recoves somehow.

**Returns**

None

#### 11.1.2.11 scan()

```
def Syringenator.scan (
              cam,
              net )
```

A routine to take a picture and report back the closest target The Computer vision routine must be able to handle multiple targets in the image.

It would be best if all targets are reported. Then this routine will determine the closest one to pursue. –ABD

**Parameters**

| | |
|---|---|
| *cam* | a Camera object to get pictures from |
| *net* | a NeuralNet object to process the pictures |

**Returns**

a Target obljject

### 11.1.3 Variable Documentation

#### 11.1.3.1 biggerDim

```
tuple Syringenator.biggerDim
```

**Initial value:**

```
1 =  (
2          (constants.PICKUP_X_MAX-constants.PICKUP_X_MIN)*SCALE,
3          (constants.PICKUP_Y_MAX-constants.PICKUP_Y_MIN)*SCALE
4      )
```

#### 11.1.3.2 crop

```
Syringenator.crop
```

**Initial value:**

```
1 =  image[
2          constants.PICKUP_Y_MIN:constants.PICKUP_Y_MAX,
3          constants.PICKUP_X_MIN:constants.PICKUP_X_MAX
4      ]
```

# 12 Class Documentation

## 12.1 Syringenator.Camera Class Reference

A class to wrap the Camera.

### Public Member Functions

- def **__init__** (self)
- def capture (self)

  *Capture an image and return it as a multidimentional matrix.*

### Public Attributes

- **pipeline**

### 12.1.1 Detailed Description

A class to wrap the Camera.

The documentation for this class was generated from the following file:

- src/pi/Syringenator.py

## 12.2 Syringenator.Log Class Reference

A class to wrap the logging functions.

### Public Member Functions

- def **__init__** (self)
- def record (self, datatype, ∗args)

  *Record system events for later analysis.*

### Public Attributes

- **file**

### 12.2.1 Detailed Description

A class to wrap the logging functions.

### 12.2.2 Member Function Documentation

#### 12.2.2.1 record()

```
def Syringenator.Log.record (
              self,
              datatype,
             * args )
```

Record system events for later analysis.

### Returns

None

The documentation for this class was generated from the following file:

- src/pi/Syringenator.py

## 12.3 Syringenator.NeuralNet Class Reference

A class to wrap the DNN.

#### Public Member Functions

- def **__init__** (self)
- def detect (self, img)

  *process an image*

#### Public Attributes

- **nn**
- **ln**

#### Static Public Attributes

- int **NETREZ** = 320
- string **WEIGHTSPATH** = "nn/yolov3-tiny-obj_37000.weights"
- string **CONFIGPATH** = "nn/yolov3-tiny-obj.cfg"

### 12.3.1 Detailed Description

A class to wrap the DNN.

The documentation for this class was generated from the following file:

- src/pi/Syringenator.py

## 12.4 roboMove Struct Reference

#### Public Attributes

- bool **typeMove**
- int **ticks**
- byte **direction**

The documentation for this struct was generated from the following file:

- src/Arduino/libraries/Syringenator/Syringenator.hpp

## 12.5 Syringenator.Target Class Reference

A class to contain everything we know about an aquired target.

#### Public Member Functions

- def **__init__** (self, box, score, center)
- def **setImg** (self, img)
- def distance (self)

  *Get the taxicab distance to the target.*
- def **getBox** (self)

#### Public Attributes

- **confidence**
- **centerX**
- **centerY**
- **box**
- **image**

### 12.5.1 Detailed Description

A class to contain everything we know about an aquired target.

### 12.5.2 Member Function Documentation

#### 12.5.2.1 distance()

```
def Syringenator.Target.distance (
              self )
```

Get the taxicab distance to the target.

#### Returns

an integer representing distance

The documentation for this class was generated from the following file:

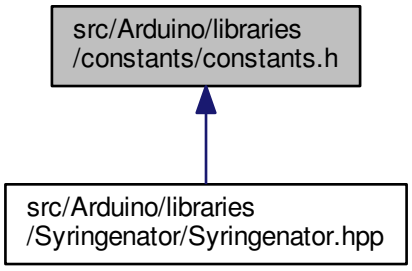
- src/pi/Syringenator.py

# 13 File Documentation

## 13.1 src/Arduino/libraries/constants/constants.h File Reference

Constants shared across the whole system.

This graph shows which files directly or indirectly include this file:




### Macros

- #define ARM_AZIMUTH_MIN 0

  *The minimum azimuth byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define ARM_AZIMUTH_MAX 180

  *The maximum azimuth byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define ARM_RANGE_MIN 12

  *The minimum range byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define ARM_RANGE_MAX 19

  *The maximum range byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define ARM_ORIENT_MIN 0

  *The minimum orientation byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define ARM_ORIENT_MAX 180

  *The maximum orientation byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- #define PICKUP_X_MIN 270

  *The minimum target center x-value that allows a pickup.*
- #define PICKUP_X_MAX 395

  *The maximum target center x-value that allows a pickup.*
- #define PICKUP_Y_MIN 420

  *The minimum target center y-value that allows a pickup.*
- #define PICKUP_Y_MAX 470

  *The maximum target center y-value that allows a pickup.*
- #define PICKUP_RADIUS 110

*The pixel value for the arm radius.*

- #define PICKUP_ARM_OFFSET 50

  *The arms offset from the bottom of the image [px].*
- #define FWD_MAX_TICKS 140

  *The maximum count of forward ticks used in moveCloser()*
- #define FWD_MIN_TICKS 2

  *The minimum count of forward ticks.*
- #define ROT_MAX_TICKS 90

  *The maximum absolute value of rotation ticks used in moveCloser()*
- #define ROT_MIN_TICKS 5

  *The minimum count of rotational ticks.*
- #define CAL_ROT_FACTOR 0.15

  *Calibration factor used in rotation calculation.*
- #define CAL_FWD_FACTOR 0.20

  *Calibration factor used in forward calculation.*
- #define CAL_ARM_OFFSET 95

  *Offset of the arm axis from the bottom of the image [mm].*
- #define CAL_CAM_HEIGHT 984

  *Height of the camera from the floor [mm].*
- #define CAL_CAM_ANGLE 1.2117

  *Angle of the camera from the horizon [radians].*
- #define CAL_CAM_AXIS 1140

  *distance to the floor on the camera's center axis*
- #define CAL_CAM_X_OFFSET -15

  *x-offset between camera and body*
- #define ARDUINO_NULL 0

  *A place holder for troubleshooting etc.*
- #define ARDUINO_STATUS_ACK 65

  *If the arduino needs to acknowledge something.*
- #define ARDUINO_STATUS_READY 82

  *If the arduino needs to indicate it is ready.*
- #define ARDUINO_STATUS_PICK_FAIL 3

  *Report that the pick failed.*
- #define ARDUINO_STATUS_PICK_SUCCESS 4

  *Report that the pick succeded.*
- #define ARDUINO_STATUS_ARM_FAULT 5

  *Report a general arm failure.*
- #define ARDUINO_STATUS_OBSTACLE 6

  *Report an obstacle detected.*
- #define ARDUINO_STATUS_NACK 78

  *Report the command not understood.*
- #define ARDUINO_LINE_FOLLOW 12

  *serial command the arduino to follow the line*
- #define ARDUINO_AVOID 13

  *serial command the arduino to avoid an obstacle*
- #define ARDUINO_RETURN 14

  *serial command the arduino to return to the line*

- #define ARDUINO_FWD 15
  *serial command the arduino forward*
- #define ARDUINO_RIGHT 16
  *serial command the arduino to rotate right*
- #define ARDUINO_LEFT 17
  *serial command the arduino to rotate left*
- #define ARDUINO_ARM_PARK 20
  *serial command the arduino to call the park action sequence*
- #define ARDUINO_ARM_DISPOSE 21
  *serial command the arduino to call the dispose action sequence*
- #define ARDUINO_ARM_PICKUP 22
  *serial command the arduino to attempt a pick, followed by three bytes: azimuth, range, and orientation*
- #define LINE_FOLLOW_TIME 100
  *timer ticks to follow the line*
- #define SERIAL_BAUD 9600
  *baudrate for serial communication between Arduino and Pi*
- #define PORT_MOTOR_FWD None
  *Arduino pin for port motor forward.*
- #define PORT_MOTOR_REV None
  *Arduino pin for port motor reverse.*
- #define STBD_MOTOR_FWD None
  *Arduino pin for starboard motor forward.*
- #define STBD_MOTOR_REV None
  *Arduino pin for starboard motor reverse.*
- #define PORT_LINE_SENSE None
  *Arduino pin for the port line sensor.*
- #define STBD_LINE_SENSE None
  *Arduino pin for the starboard line sensor.*
- #define PORT_FWD_OBSTACLE None
  *Arduino pin for the port forward obstacle sensor.*
- #define PORT_AFT_OBSTACLE None
  *Arduino pin for the port aft obstacle sensor.*
- #define STBD_FWD_OBSTACLE None
  *Arduino pin for the starboard forward obstacle sensor.*
- #define STBD_AFT_OBSTACLE None
  *Arduino pin for the starboard aft obstacle sensor.*
- #define ARM_CONTROL None
  *Arduino pin for communication with the xArm.*

### 13.1.1 Detailed Description

Constants shared across the whole system.

Includes constants used by both the arduino sketch and the the python script. The format of `constants.in` is three whitespace sparated columns:

[NAME] [value] [comments]

Any changes must be made in `constants.in` and followed by running:

```
make constants
```

–ABD

### Copyright



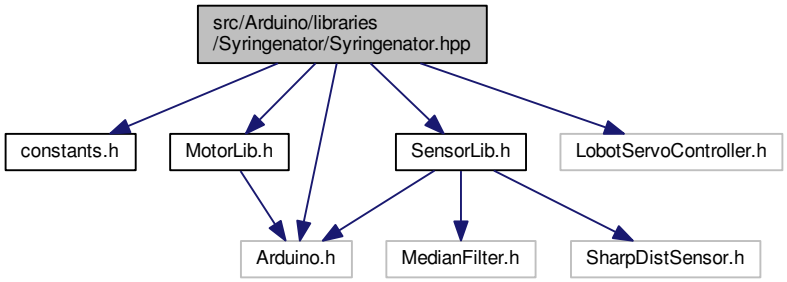
This file has been autogenerated, CHANGES MADE HERE WILL NOT PERSIST

## 13.2 src/Arduino/libraries/Syringenator/Syringenator.hpp File Reference

Arduino controller code –ABD.

```
#include "constants.h"
#include "MotorLib.h"
#include "SensorLib.h"
#include "Arduino.h"
#include <LobotServoController.h>
```

Include dependency graph for Syringenator.hpp:




### Classes

- struct roboMove

### Macros

- #define **LOGSIZE** 256
- #define **FULL90ROT_RIGHT** 875
- #define **FULL90ROT_LEFT** 835

### Functions

- int setupSensor_ISR ()

  *A function to respond to a line detector being triggered.*
- **ISR** (TIMER3_COMPA_vect)
- void **stopSensor_ISR** ()
- void **startSensor_ISR** ()
- void serialCommunication_ISR (void)

  *A function to handle incomming communication from the pi.*
- int **isDoneCommand** (int type_command)
- bool **isReady** (void)
- void moveRotate (byte direction, byte angle, bool mode=0)

  *Rotate the robot around central axis rotate by running both motors at the same speed in opposite directions.*
- void moveStraight (byte ticks, byte direction=1, bool mode=0)

  *Move the robot forward or reverse.*
- int **deadReckoning** (void)
- int logMove (int type, int ticks, byte direction=0)

  *Routine to follow the guide-line for some fixed interval.*
- void **stopLineFollow** (void)
- void **moveLineFollow** (void)
- void readLines ()

  *Routine to follow the guide-line for some fixed interval.*
- void armPark (void)

  *Move the arm to its parking position.*
- void armDispose (void)

  *Routine to dispose of a syringe once it has been picked.*
- void grabObject (byte angle, byte radius, byte handAngle)

  *Routine to attempt target pickup.*

### Variables

- volatile bool **done_with_command**

#### 13.2.1 Detailed Description

Arduino controller code –ABD.

### Copyright



#### 13.2.2 Function Documentation

#### 13.2.2.1 armPark()

```
void armPark (
             void  )
```

Move the arm to its parking position.

The parking position needs to leave a clear view of the pickup area, but also should move the center of gravity as far forward as possible to reduce drive wheel slippage.

#### 13.2.2.2 grabObject()

```
void grabObject (
             byte angle,
             byte radius,
             byte handAngle )
```

Routine to attempt target pickup.

This routine should attempt to close the claw completely and detect if an object as actually been grabbed. parameters should be bytes because they will have to be transmitted over serial from the pi. Ranges on these values TBD as convenient for the arm software, but must be recorded in the system constants file. –ABD

**Parameters**

| | |
|---|---|
| *azimuth* | arm azimuth value |
| *range* | distance to the target |
| *orientation* | rotation of the target |

**Returns**

true on successful pick, false otherwise.

#### 13.2.2.3 logMove()

```
int logMove (
             int type,
             int ticks,
             byte direction = 0 )
```

Routine to follow the guide-line for some fixed interval.

This function assumes that we are already over the line

#### 13.2.2.4 moveRotate()

```
void moveRotate (
             byte direction,
             byte angle,
             bool mode = 0 )
```

Rotate the robot around central axis rotate by running both motors at the same speed in opposite directions.

**Parameters**

| | |
|---|---|
| *ticks* | sign indicates direction of rotation: positive is rotation to the right. magnitude indicates the number of encoder ticks on each motor. |

#### 13.2.2.5 moveStraight()

```
void moveStraight (
            byte ticks,
            byte direction = 1,
            bool mode = 0 )
```

Move the robot forward or reverse.

**Parameters**

| | |
|---|---|
| *ticks* | number of encoder ticks to move. Sign indicates direction: positive is forward. |
| *mode* | 0 = line follow, 1 = object avoidance |
| *ticks* | number of encoder ticks to move. Sign indicates direction: positive is forward. |

#### 13.2.2.6 readLines()

```
void readLines ( )
```

Routine to follow the guide-line for some fixed interval.

This function assumes that we are already over the line

#### 13.2.2.7 setupSensor_ISR()

```
int setupSensor_ISR ( )
```

A function to respond to a line detector being triggered.

The line detectors are mounted forward and inboard of the wheels. This function needs to reorient the robot to clear the sensor, but also to prevent the line from being hit again.

The simplest way to do this is to rotate the opposite wheel forward until the sensor clears. Because the sensor is forward of the wheel it will rotate away from the line as the opposite wheel moves forward. This should work as long as the curvature of the line is not too great.

This may need to be two routines, one for each sensor –ABDA function to respond to a detected obstacle while under locamotion. There may be two cases to handle: whether we are line following, or aproaching. If we are line following we need to ensure that we don't lose the line while avoiding the obstacle.

This may need to be multiple routines, one for each sensor –ABD

## 13.3 src/pi/constants.py File Reference

Constants shared across the whole system.

### Variables

- int constants.ARM_AZIMUTH_MIN = 0

  *The minimum azimuth byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.ARM_AZIMUTH_MAX = 180

  *The maximum azimuth byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.ARM_RANGE_MIN = 12

  *The minimum range byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.ARM_RANGE_MAX = 19

  *The maximum range byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.ARM_ORIENT_MIN = 0

  *The minimum orientation byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.ARM_ORIENT_MAX = 180

  *The maximum orientation byte value that can be passed to the arduino with ARDUINO_ARM_PICKUP.*
- int constants.PICKUP_X_MIN = 270

  *The minimum target center x-value that allows a pickup.*
- int constants.PICKUP_X_MAX = 395

  *The maximum target center x-value that allows a pickup.*
- int constants.PICKUP_Y_MIN = 420

  *The minimum target center y-value that allows a pickup.*
- int constants.PICKUP_Y_MAX = 470

  *The maximum target center y-value that allows a pickup.*
- int constants.PICKUP_RADIUS = 110

  *The pixel value for the arm radius.*
- int constants.PICKUP_ARM_OFFSET = 50

  *The arms offset from the bottom of the image [px].*
- int constants.FWD_MAX_TICKS = 140

  *The maximum count of forward ticks used in moveCloser()*
- int constants.FWD_MIN_TICKS = 2

  *The minimum count of forward ticks.*
- int constants.ROT_MAX_TICKS = 90

  *The maximum absolute value of rotation ticks used in moveCloser()*
- int constants.ROT_MIN_TICKS = 5

  *The minimum count of rotational ticks.*
- float constants.CAL_ROT_FACTOR = 0.15

  *Calibration factor used in rotation calculation.*
- float constants.CAL_FWD_FACTOR = 0.20

  *Calibration factor used in forward calculation.*
- int constants.CAL_ARM_OFFSET = 95

  *Offset of the arm axis from the bottom of the image [mm].*
- int constants.CAL_CAM_HEIGHT = 984

  *Height of the camera from the floor [mm].*

- float constants.CAL_CAM_ANGLE = 1.2117

  *Angle of the camera from the horizon [radians].*
- int constants.CAL_CAM_AXIS = 1140

  *distance to the floor on the camera's center axis*
- int constants.CAL_CAM_X_OFFSET = -15

  *x-offset between camera and body*
- int constants.ARDUINO_NULL = 0

  *A place holder for troubleshooting etc.*
- int constants.ARDUINO_STATUS_ACK = 65

  *If the arduino needs to acknowledge something.*
- int constants.ARDUINO_STATUS_READY = 82

  *If the arduino needs to indicate it is ready.*
- int constants.ARDUINO_STATUS_PICK_FAIL = 3

  *Report that the pick failed.*
- int constants.ARDUINO_STATUS_PICK_SUCCESS = 4

  *Report that the pick succeded.*
- int constants.ARDUINO_STATUS_ARM_FAULT = 5

  *Report a general arm failure.*
- int constants.ARDUINO_STATUS_OBSTACLE = 6

  *Report an obstacle detected.*
- int constants.ARDUINO_STATUS_NACK = 78

  *Report the command not understood.*
- int constants.ARDUINO_LINE_FOLLOW = 12

  *serial command the arduino to follow the line*
- int constants.ARDUINO_AVOID = 13

  *serial command the arduino to avoid an obstacle*
- int constants.ARDUINO_RETURN = 14

  *serial command the arduino to return to the line*
- int constants.ARDUINO_FWD = 15

  *serial command the arduino forward*
- int constants.ARDUINO_RIGHT = 16

  *serial command the arduino to rotate right*
- int constants.ARDUINO_LEFT = 17

  *serial command the arduino to rotate left*
- int constants.ARDUINO_ARM_PARK = 20

  *serial command the arduino to call the park action sequence*
- int constants.ARDUINO_ARM_DISPOSE = 21

  *serial command the arduino to call the dispose action sequence*
- int constants.ARDUINO_ARM_PICKUP = 22

  *serial command the arduino to attempt a pick, followed by three bytes: azimuth, range, and orientation*
- int constants.LINE_FOLLOW_TIME = 100

  *timer ticks to follow the line*
- int constants.SERIAL_BAUD = 9600

  *baudrate for serial communication between Arduino and Pi*
- constants.PORT_MOTOR_FWD = None

  *Arduino pin for port motor forward.*
- constants.PORT_MOTOR_REV = None

*Arduino pin for port motor reverse.*

- constants.STBD_MOTOR_FWD = None

  *Arduino pin for starboard motor forward.*
- constants.STBD_MOTOR_REV = None

  *Arduino pin for starboard motor reverse.*
- constants.PORT_LINE_SENSE = None

  *Arduino pin for the port line sensor.*
- constants.STBD_LINE_SENSE = None

  *Arduino pin for the starboard line sensor.*
- constants.PORT_FWD_OBSTACLE = None

  *Arduino pin for the port forward obstacle sensor.*
- constants.PORT_AFT_OBSTACLE = None

  *Arduino pin for the port aft obstacle sensor.*
- constants.STBD_FWD_OBSTACLE = None

  *Arduino pin for the starboard forward obstacle sensor.*
- constants.STBD_AFT_OBSTACLE = None

  *Arduino pin for the starboard aft obstacle sensor.*
- constants.ARM_CONTROL = None

  *Arduino pin for communication with the xArm.*

#### 13.3.1 Detailed Description

Constants shared across the whole system.

Includes constants used by both the arduino sketch and the the python script. The format of `constants.in` is three whitespace sparated columns:

[NAME] [value] [comments]

Any changes must be made in `constants.in` and followed by running:

```
make constants
```

–ABD

**Copyright**

Copyright © 2019 by the authors. All rights reserved.

This file has been autogenerated, CHANGES MADE HERE WILL NOT PERSIST

### 13.4 src/pi/Syringenator.py File Reference

This is the main control script.

### Classes

- class Syringenator.Target

  *A class to contain everything we know about an aquired target.*
- class Syringenator.NeuralNet

  *A class to wrap the DNN.*
- class Syringenator.Camera

  *A class to wrap the Camera.*
- class Syringenator.Log

  *A class to wrap the logging functions.*

### Namespaces

- Syringenator

  *The top-level Pi program.*

### Functions

- def Syringenator.rescale (detection)

  *Rescale the bounding box coordinates scale the bounding box coordinates back relative to the size of the image, keeping in mind that YOLO actually returns the center (x, y)-coordinates of the bounding box followed by the boxes' width and height.*
- def Syringenator.extractTargets (dataIn)

  *process the OpenCV output to generate actionable targets*
- def Syringenator.imageCart2floorCart (t)

  *Derive floor position from image data.*
- def Syringenator.pixelRadius (t)

  *Determine how far a target is outside of the pickup radius.*
- def Syringenator.floorCart2armCylinder (x, y)

  *Derive cylindrical coordinates, centered on the arm from cartesian coordinates centered on the camera.*
- def Syringenator.floorCart2steer (x, y)

  *Generate a steering azimuth from floor cartesian.*
- def Syringenator.scan (cam, net)

  *A routine to take a picture and report back the closest target The Computer vision routine must be able to handle multiple targets in the image.*
- def Syringenator.canBePicked (t)

  *A routine to determine if the target is in position to be picked up.*
- def Syringenator.approach (t)

  *Move the robot closer to the given target.*
- def Syringenator.avoid ()

  *avoid an obstacle*
- def Syringenator.pickUp (t)

  *Attempt to pickup and dispose the target.*
- def Syringenator.returnToLine ()

  *signl the arduino to return to the line.*
- def Syringenator.lineFollow ()

  *Follow the line.*

### Variables

- bool Syringenator.DEBUG_CAPTURE = False

  *Display an image once it's been captured.*
- bool Syringenator.DEBUG_AQUISITION = False

  *Draw centers and bounding boxes on the image.*
- bool Syringenator.DEBUG_APPROACH = False

  *Report on turn and forward values during approach.*
- bool Syringenator.DEBUG_TRANSFORM = False

  *Report azimuth and range values during pickUp.*
- bool Syringenator.DEBUG_ORIENTATION = False

  *display processed image for hand orientation*
- bool Syringenator.DEBUG_TIMING = False

  *report on yolo timing*
- bool Syringenator.CAL_X = False

  *Interrupt the normal loop and take X calibration photos.*
- bool Syringenator.CAL_Y = False

  *Interrupt the normal loop and take Y calibration photos.*
- bool Syringenator.DISABLE_WHEELS = False

  *Disable any wheel commands.*
- bool Syringenator.DISABLE_LINE_FOLLOW = False

  *robot only picks and returns to start position*
- int **Syringenator.FRAME_RATE** = 30
- int **Syringenator.IMG_WIDTH** = 640
- int **Syringenator.IMG_HEIGHT** = 480
- float **Syringenator.FOV_X** = 63.4
- float **Syringenator.FOV_Y** = 40.4
- int **Syringenator.CONFIDENCE** = .5
- int **Syringenator.NMS_THRESHOLD** = .1
- int **Syringenator.XPIX2LEN** = 10/6.48
- int **Syringenator.YPIX2LEN** = 10/6.0
- bool Syringenator.onTheLine = True

  *boolean indicating whether we are on the line*
- bool Syringenator.obstacle = False

  *boolean indicating that we have detected an obstacle*
- bool **Syringenator.inWindow** = False
- Syringenator.target = None

  *The currently aquired target.*
- int Syringenator.pickUpCount = 0

  *the number of times a pickup has been attempted*
- int Syringenator.PICKUP_LIMIT = 2

  *the maximum number of times to attempt a pickup*
- **Syringenator.log** = Log()
- **Syringenator.camera** = Camera()
- **Syringenator.neuralNet** = NeuralNet()
- **Syringenator.comPort** = SySerial.ComPort()
- **Syringenator.status** = comPort.status()
- int **Syringenator.SCALE** = 4

- **Syringenator.image** = camera.capture()
- **Syringenator.crop**
- **Syringenator.bigger** = cv2.resize(src=crop, dsize=(0,0), fx=SCALE, fy=SCALE)
- tuple **Syringenator.biggerDim**
- tuple **Syringenator.centerX** = (IMG_WIDTH/2-constants.PICKUP_X_MIN+(constants.CAL_CAM_X_OFFSET))∗S↩ CALE
- int **Syringenator.CROP_Y_MIN** = 400

### 13.4.1 Detailed Description

This is the main control script.

It will run on the Raspberry Pi and direct all robot operations.

By convention each arduino command routine checks for arduino ready before starting, and logs arduino status on exit.

**Todo** how do we initialize the robot run? a button press? –ABD

#### Copyright

# Index