EAN-Landing Aid

PN: EAN-Landing-Aid

9/16/2020

Contact:

Web: sightlineapplications.com

Sales: sales@sightlineapplications.com

Support: support@sightlineapplications.com

Phone: +1 (541) 716-5137

Export Controls

Exports of SightLine products are governed by the US Department of Commerce, Export Administration Regulations (EAR); classification is ECCN 4A994. The export summary sheet located on the support/documentation page of our website outlines customers responsibilities and applicable rules. SightLine Applications takes export controls seriously and works to stay compliant with all export rules.

Copyright and Use Agreement

Before loading, downloading, installing, upgrading or using any Licensed Product of SightLine Applications, Inc., users must read and agree to the license terms and conditions outlined in the End User License Agreement.

All data, specifications, and information contained in this publication are based on information that we believe is reliable at the time of printing. SightLine Applications, Inc. reserves the right to make changes without prior notice.

Alerts

The following notifications are used throughout the document to help identify important safety and setup information to the user:

⚠ **CAUTION:** Alerts to a potential hazard that may result in personal injury, or an unsafe practice that causes damage to the equipment if not avoided.

ⓘ **IMPORTANT:** Identifies crucial information that is important to setup and configuration procedures.

📄 *Used to emphasize points or reminds the user of something. Supplementary information that aids in the use or understanding of the equipment or subject that is not critical to system use.*

Contents

List of Figures

List of Tables

1 Overview

This document describes how to configure the SightLine Applications landing aid detection capability.

1.1 Additional Support Documentation

Additional Engineering Application Notes (EANs) can be found on the Documentation page of the SightLine Applications website.

The Panel Plus User Guide provides a complete overview of settings and dialog windows located in the Help menu of the Panel Plus application.

The Interface Command and Control (IDD) describes the native communications protocol used by the SightLine Applications product line. The IDD is also available as a PDF download on the Documentation page under Software Support Documentation.

1.2 SightLine Software Requirements

ⓘ **IMPORTANT:** The Panel Plus software version should match the firmware version running on the board. Firmware and Panel Plus software versions are available on the Software Download page.

1.3 Application Bit Requirements

The functions described in this EAN require Application Bits (app bits) purchased from SightLine. App bits are enabled with a license file provided by SightLine at initial unit purchase or during a license upgrade process. License files use a hardware ID that is applicable to a specific hardware serial number. For questions and upgrade support contact Sales.

Table 1: Application Bits Requirement Table

Function	Initial Software Release	Required Application Bit(s) v7 License
Landing Aid	2.23.xx	Precision Landing 0x0000 2000

2 Setup and Configuration

Before using the landing aid pattern in the field, print the SightLine landing aid pattern and setup a bench configuration to help understand the setup and configuration parameters. The landing aid pattern zip file can be downloaded from the SightLine website.

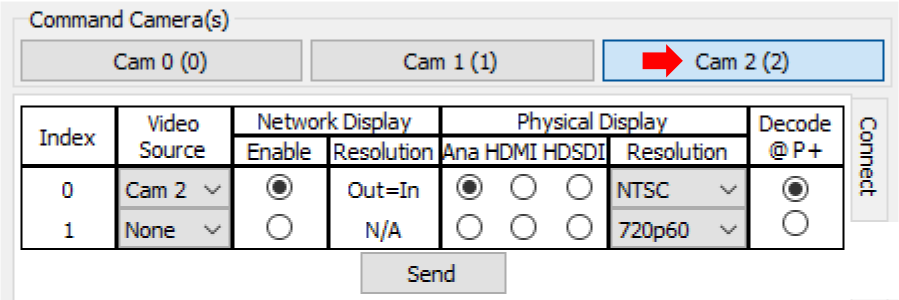
ⓘ **IMPORTANT:** Printing to an 8½ in x 11 in paper will result in a target that matches the default size for the landing aid detection tool. The landing aid pattern can be scaled to fit most landing scenarios. When scaling the landing aid pattern, it is important to maintain the aspect ratio of the pattern and the white space within the red border.

The following steps reference the Panel Plus software.

1. Connect to the board using the Panel Plus application. See the corresponding OEM startup guide for connection instructions.

2. Confirm the firmware version is 2.22 or greater and the landing feature license is enabled. This can be verified on the *Connect* tab of Panel Plus.

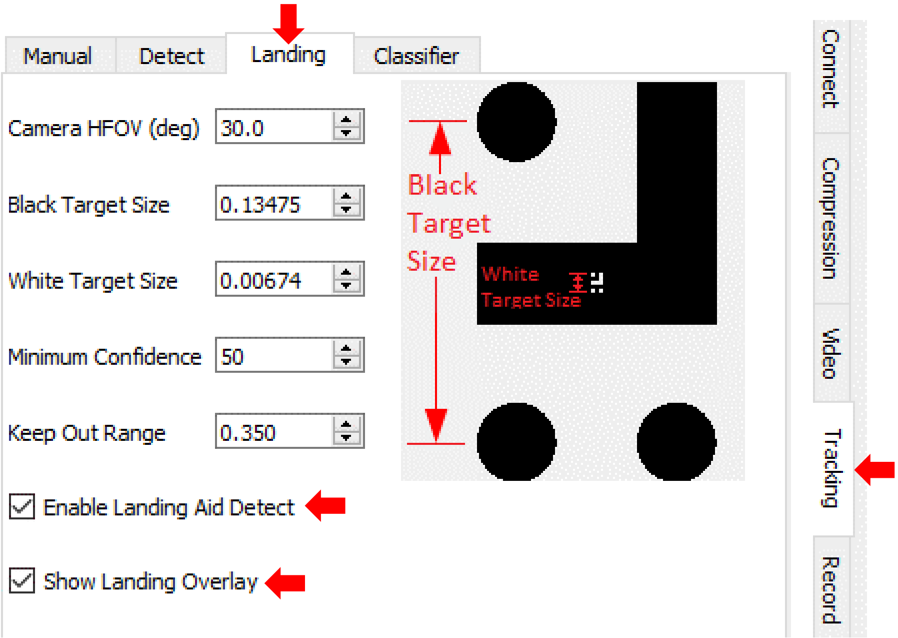
3. In *Panel Plus* set the commanded camera to the camera that will be used for landing. In this example for the 3000-OEM, *Cam 2(2)* is used for landing.



Command Camera(s)

Index	Video Source	Network Display		Physical Display				Decode @P+
		Enable	Resolution	Ana	HDMI	HDSDI	Resolution	
0	Cam 2	●	Out=In	●	○	○	NTSC	●
1	None	○	N/A	○	○	○	720p60	○

4. From the *Tracking* tab » *Landing* tab, check *Enable Landing Aid Detect* and *Show Landing Overlay*.



Camera HFOV (deg) 30.0

Black Target Size 0.13475

White Target Size 0.00674

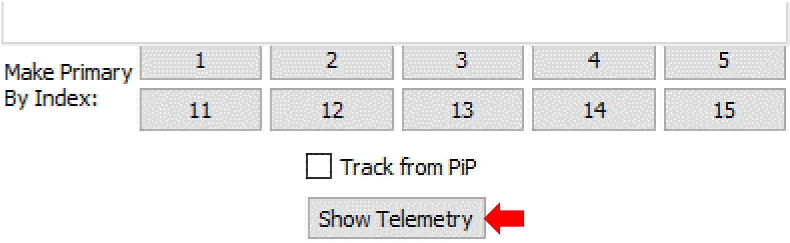
Minimum Confidence 50

Keep Out Range 0.350

☑ Enable Landing Aid Detect

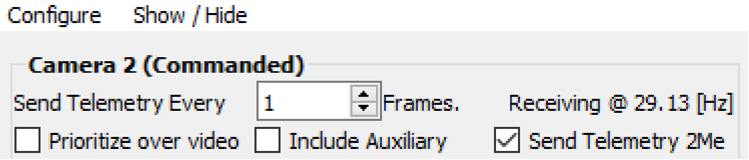
☑ Show Landing Overlay

5. Click *Show Telemetry*.



□ Track from PiP

6. In the *Telemetry* window, select the telemetry for the command camera.

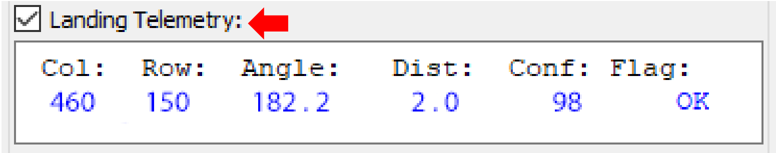


Camera 2 (Commanded)

Send Telemetry Every 1 Frames. Receiving @ 29.13 [Hz]

□ Prioritize over video □ Include Auxiliary ☑ Send Telemetry 2Me

7. Check the *Landing Telemetry* box to show the coordinates.



☑ Landing Telemetry:

```
Col:   Row:   Angle:   Dist:   Conf:  Flag:
460    150    182.2    2.0     98      OK
```

8. Verify that the cross overlay is centered on the landing aid pattern as shown in Figure 1.
9. Verify that the coordinates and score (confidence) show up in the *Landing Telemetry* section.

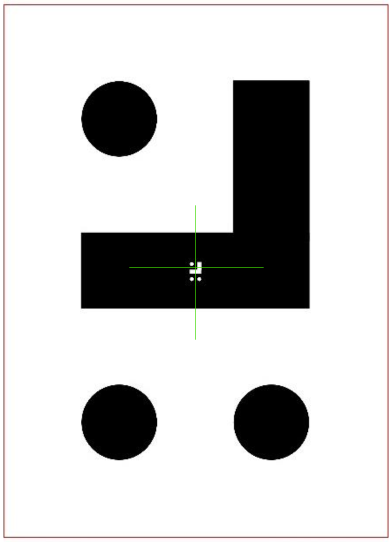


Figure 1: Centered Cross Overlay

2.1 Target Size Setup

When setting up the landing pattern, it is important to pick units for the target size that will not overflow the distance value returned in the telemetry. For example, if the target size is specified in meters, the maximum distance possible to report is 65536 meters. If the units are specified in cm, the maximum distance is 65536 cm = 655.36 meters.

Any units of measurement can be used in the black and white landing pattern target size fields. Whatever units are used, the result distance will be in those units. For example, if the black target size on the pattern is 0.13475 meters. The distance to target result will be in meters.

2.1.1 Determine Physical Target Size

Black Target Size: Measure the vertical distance between the centers of the two black dots of the printed landing pattern as shown in Figure 2.

White Target Size: 5% of the black target size height.

The landing aid target pattern can also be reverse polarity: a black background with a large white pattern and a small black target inside. For example, if using a thermal emissive/reflective target reverse polarity can be helpful. When using a reverse polarity target the white target size can be larger than the black target size. Reverse polarity landing patterns can be found in the landing aid pattern zip file.

When scaling the landing aid pattern, it is important to maintain the aspect ratio of the pattern and the white space within the red border.

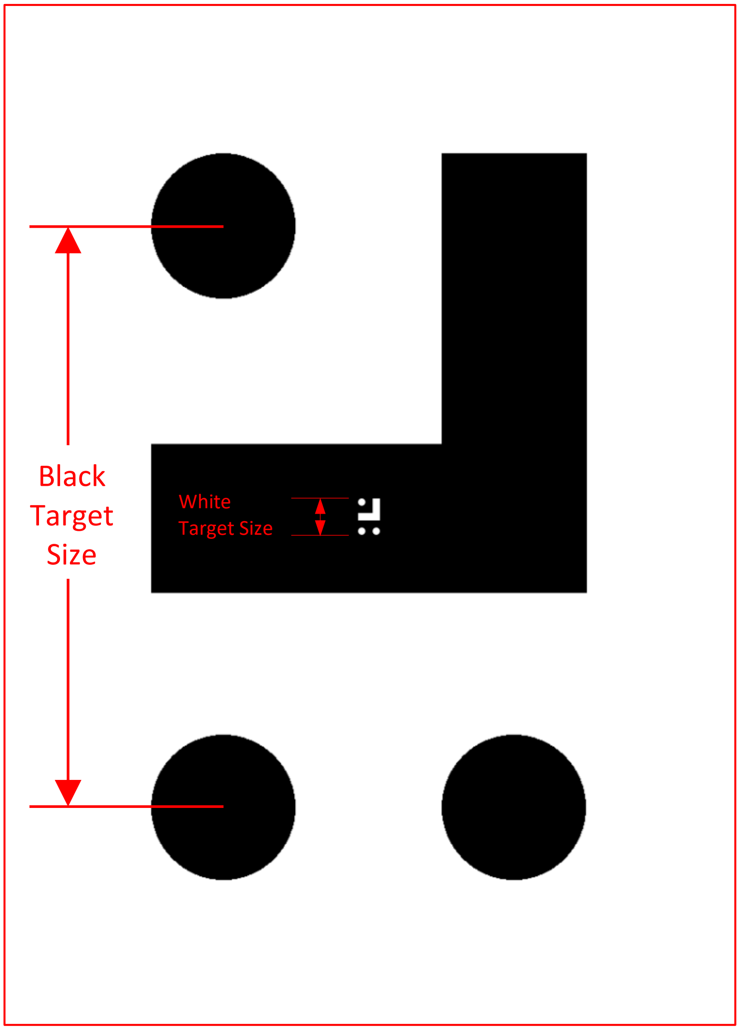


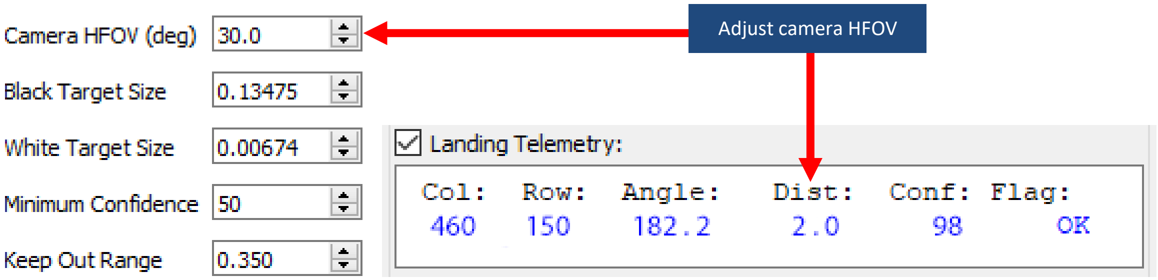
Figure 2: Landing Pattern Target Size

2.2 Camera Horizontal Field of View (HFOV)

The camera HFOV can be set using one of the following methods:

Method A): Calculate the camera HFOV from known camera and lens parameters.

Method B): Calibrate by measuring the distance between the camera and the target. Adjust the camera HFOV until the distance shown in the Telemetry dialog matches the measured distance between the camera and target.



Adjust camera HFOV

Camera HFOV (deg)	30.0
Black Target Size	0.13475
White Target Size	0.00674
Minimum Confidence	50
Keep Out Range	0.350

☑ Landing Telemetry:

```
Col:   Row:   Angle:   Dist:   Conf:  Flag:
460    150    182.2    2.0      98     OK
```

Figure 3: Setting Camera HFOV

2.3 Minimum Confidence

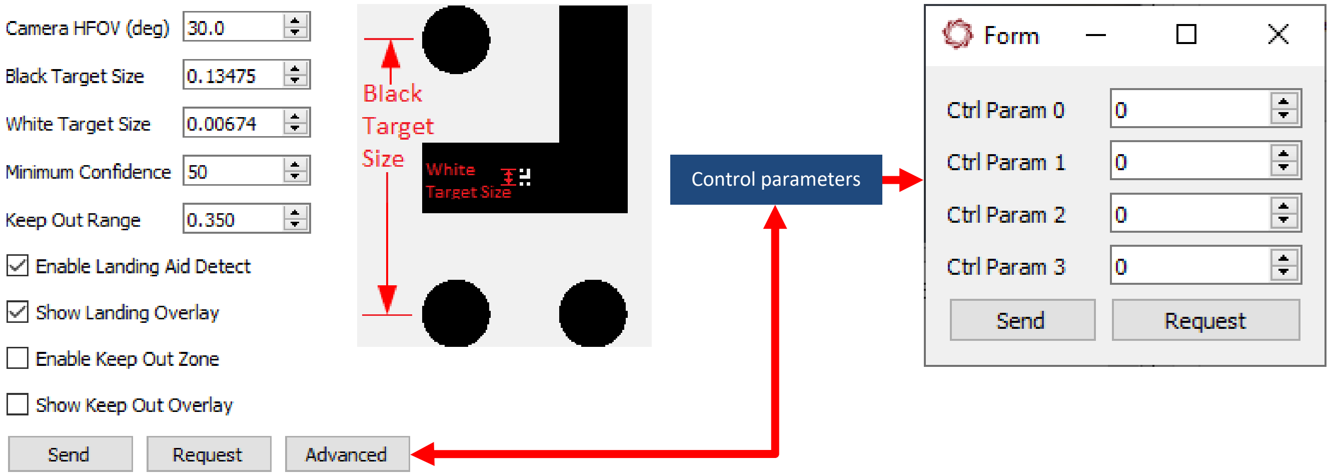
If the confidence score is above this threshold, detection is considered a success. Confidence scores below this threshold will not be sent. As the confidence value approaches the minimum confidence threshold (40%) the Landing Overlay will change color from green to yellow.

2.4 Control Parameters

Four generic control parameters are available that can be sent to the SightLine system using the **Landing Aid (0x81**) command and will be included with the **Landing Position (0x83)** output telemetry packet. In the ARM example application these parameters are used to set the PID gains in the landing controller.

These parameters are 16-bit unsigned integers in the interface; any scaling or conversion is left up to the end user of the landing position packet.

The control parameters can be accessed by clicking on the *Advanced* button in the *Landing* dialog window shown in Figure 4.



Camera HFOV (deg) 30.0
Black Target Size 0.13475
White Target Size 0.00674
Minimum Confidence 50
Keep Out Range 0.350
☑ Enable Landing Aid Detect
☑ Show Landing Overlay
☐ Enable Keep Out Zone
☐ Show Keep Out Overlay

Ctrl Param 0 0
Ctrl Param 1 0
Ctrl Param 2 0
Ctrl Param 3 0

Figure 4: Control Parameters

2.5 Keep Out Zone

The keep-out region is within the circle boundary. If there are no detections found within the circle it will be green. If a detection is found it will turn red.

Figure 5 shows an example of a perimeter landing zone detection. See the Landing Aid Zone Detection video for a demonstration of this feature.

In certain detection modes the landing pad can cause detections. These can be filtered without changing the end detection state.

Additional information about detections are available in the Landing Positions (0x83) packet in the Interface Command and Control (IDD).

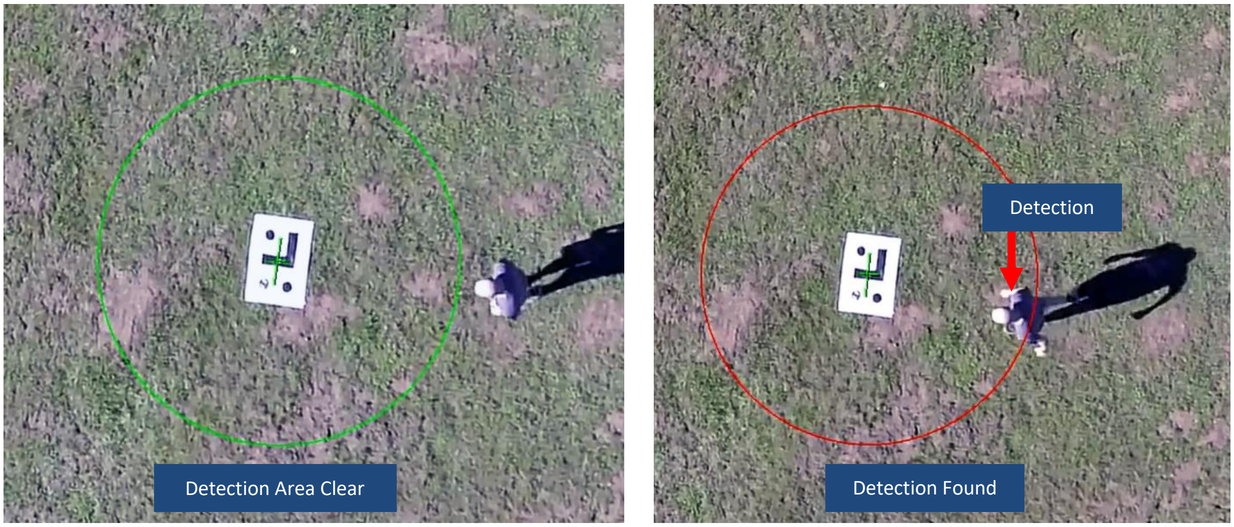
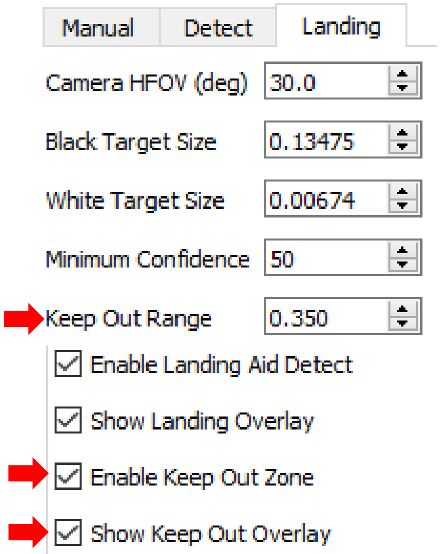


Figure 5: Landing Aid Keep Out Zone

2.5.1 Keep Out Zone Configuration

1. In Panel Plus » *Tracking* tab » *Landing*. From the Landing tab, check *Enable Keep Out zone* and *Show Keep Out Overlay*.
2. Set the *Keep Out Range*. Units are the same as those used in setting up the target size.



Camera HFOV (deg) 30.0

Black Target Size 0.13475

White Target Size 0.00674

Minimum Confidence 50

Keep Out Range 0.350

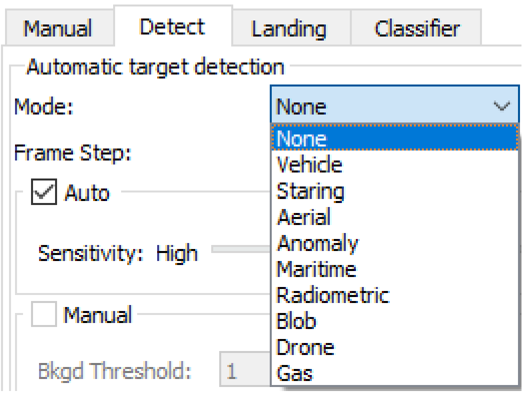
☑ Enable Landing Aid Detect

☑ Show Landing Overlay

☑ Enable Keep Out Zone

☑ Show Keep Out Overlay

3. Click the *Detect* tab. Enable a detection mode from the dropdown menu. For more information about choosing and working with detection modes, see the EAN-Detection-Modes document.
4. Verify that the detection mode is working correctly.



Automatic target detection

Mode: None

Frame Step:

☑ Auto

Sensitivity: High

□ Manual

Bkgd Threshold: 1

3 Integration with Auto Pilot (Flight Control System)

The landing aid feature has been tested in conjunction with Micropilot, Pixhawk, and Piccolo systems. The landing aid provides the location, orientation, and distance to the landing aid from the camera center. This information can be used to control the autopilot.

One method is to run a custom ARM application on the SightLine hardware that uses the telemetry from the landing aid along with the telemetry from the autopilot to generate autopilot commands. The format of these commands depends on the specific autopilot. Extensive telemetry data is required from the autopilot, which is also system dependent. For example, to use the telemetry from the landing aid to drive the vehicle in a specific direction or location, the attitude of the aircraft and the mount location and angles of the gimbal must be known. This type of integration is platform specific.

Information on creating a custom ARM application for these systems can be found on the Example Code page of the SightLine website.

The EAN-ARM Application Development describes how to setup a PC to develop applications that can be run on the ARM processor of the 1500-OEM, 3000-OEM, or 4000-OEM.

The SLA ARM examples installer includes the landing sample application *(…\SightLine Applications\SLA-Examples-ARM X.YY.ZZ\**SLALandingApp***), which can run on the ARM. This example receives landing aid telemetry from VideoTrack along with autopilot telemetry. The data is used to generate commands that are sent back to the autopilot to guide the landing. The examples in this code communicate with the Pixhawk autopilot using MAVLINK command. Familiarity with MAVLINK is recommend. Integrations have also been done with Micropilot and Piccolo autopilots. Contact Sales for more information.

The example code can run on ARM or PC systems. The PC versions compile under Visual Studio 2013. The ARM versions compile under TI Code Composer Studio (1500-OEM and 3000-OEM). It is recommended to do much of the development and test using a PC since the telemetry from the landing aid hardware is directed to the PC, and the commands to the autopilot come from the PC. This provides a simplified process for development and debugging. Most of the code is common to both platforms with SightLine provided abstractions for Linux and Windows (sockets, serial ports, threads).

A developer that is familiar with autopilots should be able to understand the example code and integrate it with an autopilot. Contact Sales for help in locating contract developers that are familiar this sample code.

3.1 Building the SLALandingAid (ARM) - Example

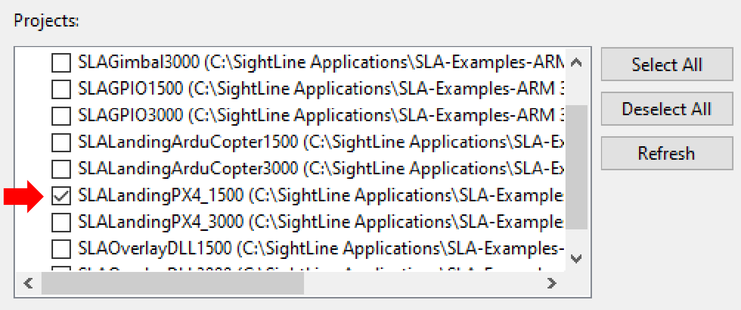
This example assumes the following conditions:

- The operator has followed instructions in EAN-ARM-Application-Development to set up IDE, build tools, Virtual Machine (for NFS), etc.
- The SightLine Example Code has been installed (e.g., *C:\SightLine Applications\SLA-Examples-ARM X.YY.ZZ\\)*.

1. Open Code Composer Studio v5.5 (CCS).

   For Windows 7/8, use Code Composer Studio v5.4

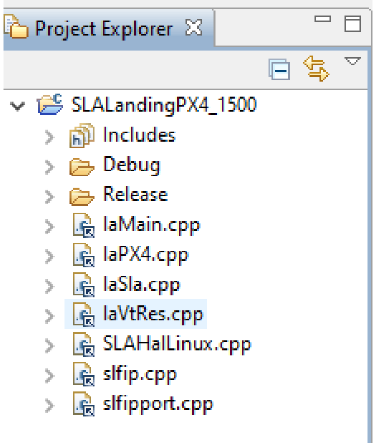
2. From the main menu go to *File* » *Import…*
3. Select *Existing Projects into Workspace*, then click *Next*.
4. Select the root directory and click *Browse*.
5. Select the directory of the SightLine Example code and click *OK*. A list of available projects will appear.
6. Select the *SLALandingPX4_1500* project from the list.



Projects:

□ SLAGimbal3000 (C:\SightLine Applications\SLA-Examples-ARN
□ SLAGPIO1500 (C:\SightLine Applications\SLA-Examples-ARM 3
□ SLAGPIO3000 (C:\SightLine Applications\SLA-Examples-ARM 3
□ SLALandingArduCopter1500 (C:\SightLine Applications\SLA-Ex
□ SLALandingArduCopter3000 (C:\SightLine Applications\SLA-Ex
☑ SLALandingPX4_1500 (C:\SightLine Applications\SLA-Example
□ SLALandingPX4_3000 (C:\SightLine Applications\SLA-Example
□ SLAOverlayDLL1500 (C:\SightLine Applications\SLA-Examples-

7. Click *Finish*. The Project Explorer shows the project manager and source code.



SLALandingPX4_1500
- Includes
- Debug
- Release
- laMain.cpp
- laPX4.cpp
- laSla.cpp
- laVtRes.cpp
- SLAHalLinux.cpp
- slfip.cpp
- slfipport.cpp

8. Right-click on the *SLALandingPX4_1500* project and select *Build Project*.

The Console window displays information as the build executes. Once complete the console window will show that the binary code has been pushed to the mapped drive (R:).

9. Move the binary to the target hardware using WinSCP (or any FTP client).

If the target has been configured to boot from NFS (as recommended from the EAN-ARM-Application-Development), then the file will already be available.

10. Use Tera Term to establish an SSH session to the target.

11. If necessary, set the binary to be executable:

```
root@sla1500:~# chmod a+x SLALandingPX4_1500
```

12. Run the executable. It will begin to report some diagnostics as well as communication information with the PixHawk.

```
root@sla1500:~# ./SLALandingPX4_1500
SLASocket init
Not using multicast group
COMMSTATE_INIT
GOT VERSION 3.1
COMMSTATE_INIT
```

4 Questions and Additional Support

For questions and additional support, please contact Technical Support. Additional support documentation and Engineering Application Notes (EANs) can be found on the Documentation page of the SightLine Applications website.

Landing Pattern

(shown below)

When scaling the landing pattern, it is important to maintain the aspect ratio of the pattern and the white space area within the red border. See EAN-Landing-Aid for setup and configuration.