User Guide

Boot Loader Framework User Guide

Version: 1.0
Date Released: June 6, 2017
Document Number: AY006UBL3-1

Ayla Networks

Referenced Documents

Ayla Networks does not supply all documents that are referenced in this document with the equipment. Ayla Networks reserves the right to decide which documents are supplied with products and services.

Contact Information

Ayla Networks TECHNICAL SUPPORT and SALES

Contact Technical Support: https://support.aylanetworks.com
or support@aylanetworks.com

Contact Sales: https://www.aylanetworks.com/company/contact-us

Ayla Networks REGIONAL OFFICES

HEADQUARTERS

Silicon Valley
4250 Burton Drive, Suite 100
Santa Clara, CA 95054
Phone: +1 408 830 9844
Fax: +1 408 716 2621

Chicago
10 N. Martingale Road, Suite 400
Schaumburg, IL 601073

Boston
275 Grove Street, Suite 2-400
Newton, MA 02466

Table of Contents

List of Figures

1 Introduction

This guide introduces you to the OTA related tool chain, such as the image packager utility and embedded MCU bootloader.

1.1 Intended Audience

This document is written for all users.

1.2 Related Documentation

Getting Started Guides:

- Getting Started (AY006AIO2)
- Hello World! (AY006GHW2)
- Product Architecture Comparison (AY006OPR2)
- Product Development Overview (AY006GOV1)
- Demo API (AY006MDA2)
- Design Kit – USB and TTL Serial Communication User Manual (AY006TTL2)
- Mobile Getting Started (AY006GSB6)
- Device OEM Domain Transfer (AY006UDT3)

1.3 Customer Support

Technical support is available through the Ayla Support website at: https://support.aylanetworks.com, or via email at support@aylanetworks.com.

1.4 Required Hardware & Software

1.4.1 Hardware

- A workstation grade computer with the following configuration
  - Windows based: **8 GB RAM** and **256 GB storage** or better, preferable SSD based storage for best performance.
  - Apple based: **8 GB RAM** and **256 GB storage** or better.
- Ayla 'Design Kit'

1.4.2 Software

- Workstation 'on the metal' OS:
  - Windows 7 Pro or better (Windows 10 Pro version 1511 or better)

NOTE If you are new to Windows 10 then use the 'Settings App' to find out which build and version of Windows 10 you have. Navigate to System > About and scroll down. You'll see the "Version" and "Build" numbers here.

  - Mac OS (OS X 10.11.6 or better)
- Ayla Design Kit Demo Source Package **v1.6** – This is the demo source kit code. It is already loaded on the Design Kit MCU and you will be modifying this code in this document.

1.5 Abbreviations and Acronyms

Most abbreviations and acronyms are spelled out the first time that they are used in the document and are not always listed in the following table. However, abbreviations and acronyms that are frequently used in Ayla user documentation or commonly known by the intended audience of the document may also or only be spelled out in this table.

- Over-the-air programming (OTA): Various methods of distributing new software, configuration settings, and other details (i.e., encryption keys) to devices like cellphones, set-top boxes or secure voice communication equipment (encrypted 2-way radios). For more information see:

  https://en.wikipedia.org/wiki/Over-the-air_programming

2 Development OS Environment

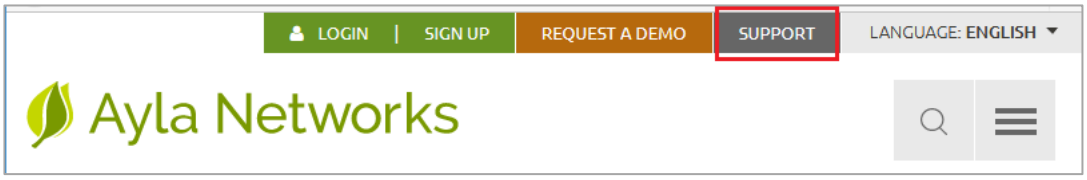
The Ayla Design Kit-related toolset in this document are all Microsoft Windows based. A Mac or Linux-based workstation with a hypervisor installed to host a virtual Windows instance can also be used.

2.1 Get the Ayla full demo 1.6 package

You need to have a developer account.

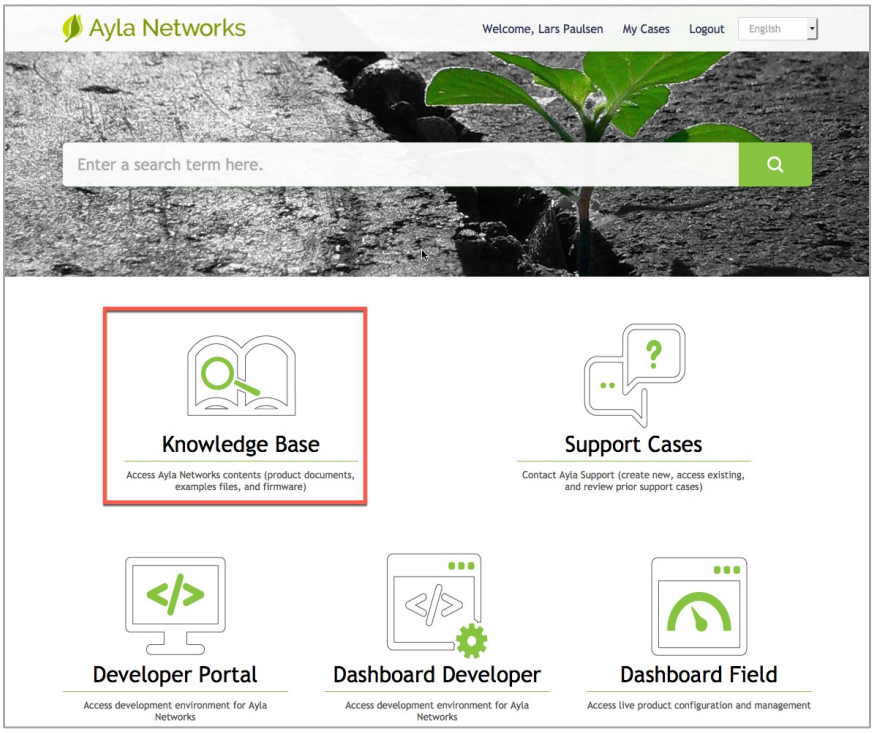
1. Go to http://www.aylanetworks.com
2. At the top of the page, click **SUPPORT** (Figure 1).

Figure 1 – Support Link



3. Log-in to your developer account.
4. On the **Support Center** opens, click **Knowledge Base** (Figure 2).

Figure 2 – Knowledge Base link

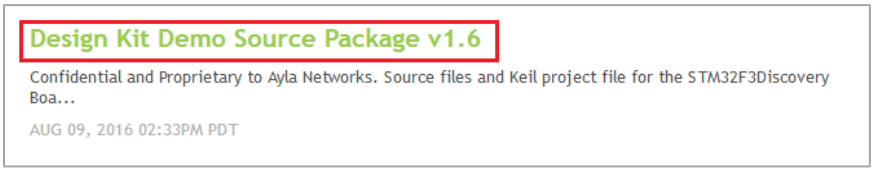


5. In Getting Started section, click **View all articles** (Figure 3).

Figure 3 – View All Articles Link

6. On the list of all articles, click **Design Kit Demo Source stm32f3 v1.6** (Figure 4).

Figure 4 – Download Link



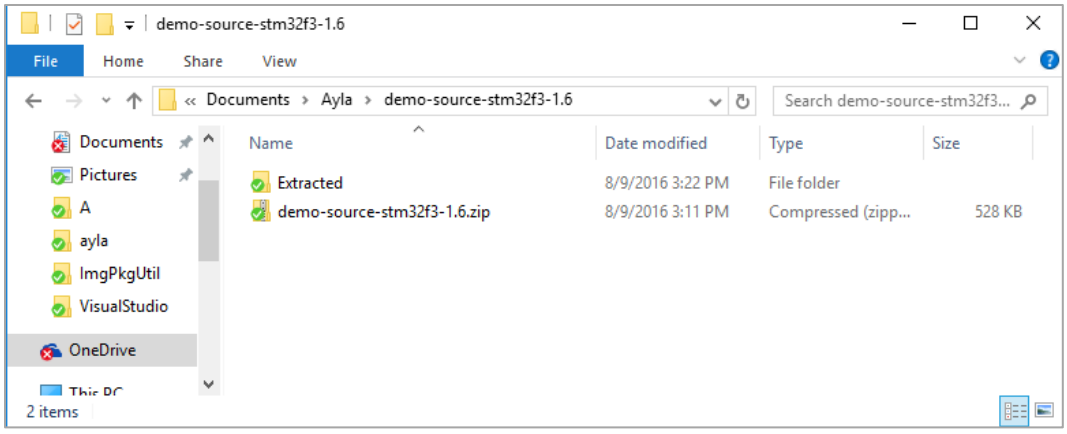
7. Save the file to your system.

2.2 Install demo-source-stem32f3-1.6 files.

When the file is downloaded, decompress the demo-source-stem32f3-1.6 zip file (Figure 5) to a convenient location, such as:

`…\Documents\Ayla\demo-source-stm32f3-1.6\Extracted\...`

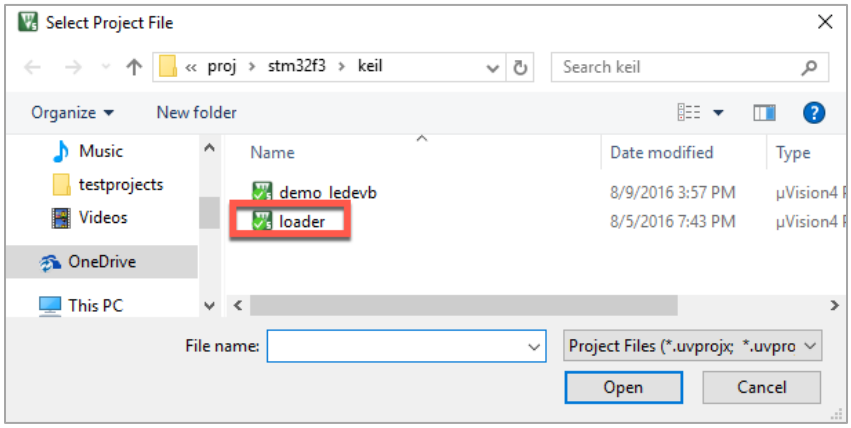
Figure 5 – Decompressed File Location



3 Boot Loader Framework Components

The second project file in the **keil** folder, named **loader** (Figure 6), represents the code for the boot manager application (often referred to as boot loader).

Figure 6 – loader file location



The loader program initially executes when the processor is powered-up or restarted. Some initial tests check if a new firmware image was downloaded since last power-up or restart. If a new valid image is available, it is copied to the persistent executing memory space.

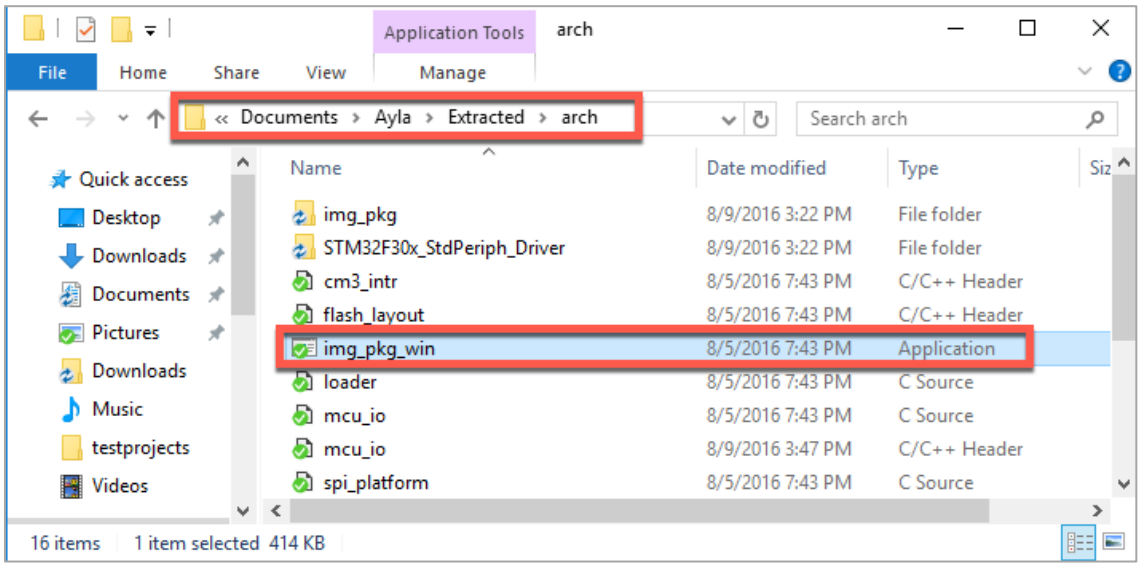
If no new firmware image is available, the 'loader' proceeds directly to transfer control of the main application image.

The Ayla cloud device upgrade function can, theoretically, works with an image up to 4 GB. This could be a problem since Ayla protocol downloads in 2KB chunks. Even images of several megabytes could be difficult to process.

The MCU demo 'loader' code has an internal structure of 64K. As needed, this limitation can be customized to support larger images.

The embedded 'loader' has a matching package encoder utility used in tandem with the embedded 'loader' functionality (Figure 7).

Figure 7 – Package encoder



At a high level, the 'img_pkg_win.exe' utility adds image meta-data to the original application image as follows:

```
struct image_hdr {
    u32 ih_magic;             /* IMAGE_MAGIC */
    u16 ih_crc;               /* CCITT crc-16 over the image (ih_crc = 0) */
    u16 ih_len;               /* length */
};

struct image {
    struct image_hdr i_hdr;
    u8 i_vers[72];            /* version string */
};
```

The new packaged image gets uploaded to the Ayla Developer or Field Portal Dashboard. On the OEM-facilitated command, it is downloaded to the device's persistence memory area.

Upon restart or reboot, the embedded Loader uses the added metadata to verify image integrity. On validation, the 'loader' copies the new execution code to the active application execution space.

The default example loader implementation for the stm32f3 MCU implements the following memory Flash layout:

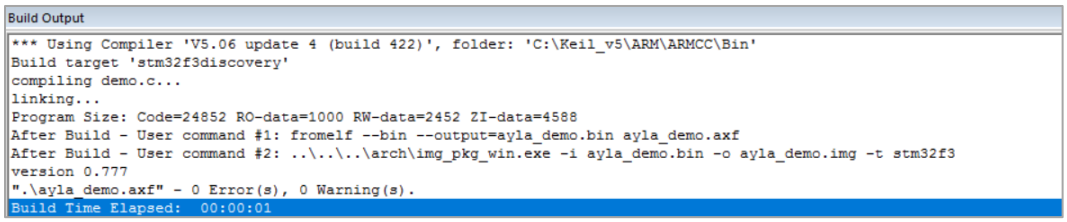
```
 *  0x08000000 - bootloader
 *  0x08001c00 - copy_progress
 *  0x08002000 - active image
 *  0x08020000 - inactive image
 *  0x0803e000 - scratch area (when switching bw active/inactive)
 *  0x08040000 - end of flash
```

For general understanding of the of the 'loader's internal state machine, these are the defined boot-states:

```
enum mcu_boot_state {
    MCU_BOOT_OK = 0,
    MCU_BOOT_INACTIVE = 1,
    MCU_BOOT_TEST = 2,
    MCU_BOOT_FALLBACK_START = 3,
    MCU_BOOT_FALLBACK = 4
};
```

The Design Kit - Hello World! (AY006GHW2) document describes compiling MCU code with Keil IDE. This compilation generates the MCU image during the post-build (Figure 8).

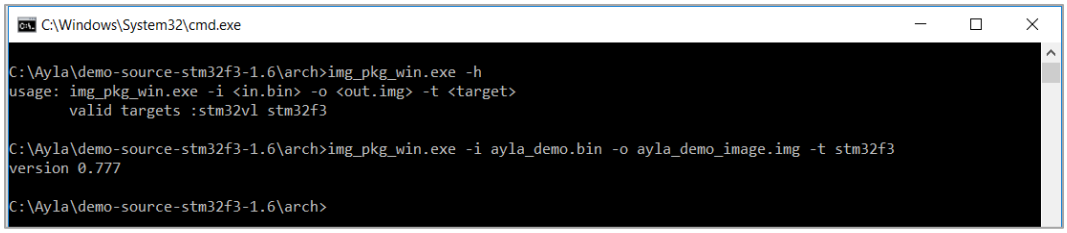
Figure 8 – Compiling MCU Image



The img_pkg_win.exe tool can generate the MCU image from a command line prompt (Figure 9). This is an example of how to generate the MCU image.

```
../img_pkg_win.exe -i <in.bin> -o <out.img> -t <target>
       valid targets :stm32vl stm32f3
```

Figure 9 – img_pkg_win.exe execution



4 View Default 'loader' Packager Implementation Code

Visual Studio can view the default 'loader' packager code.

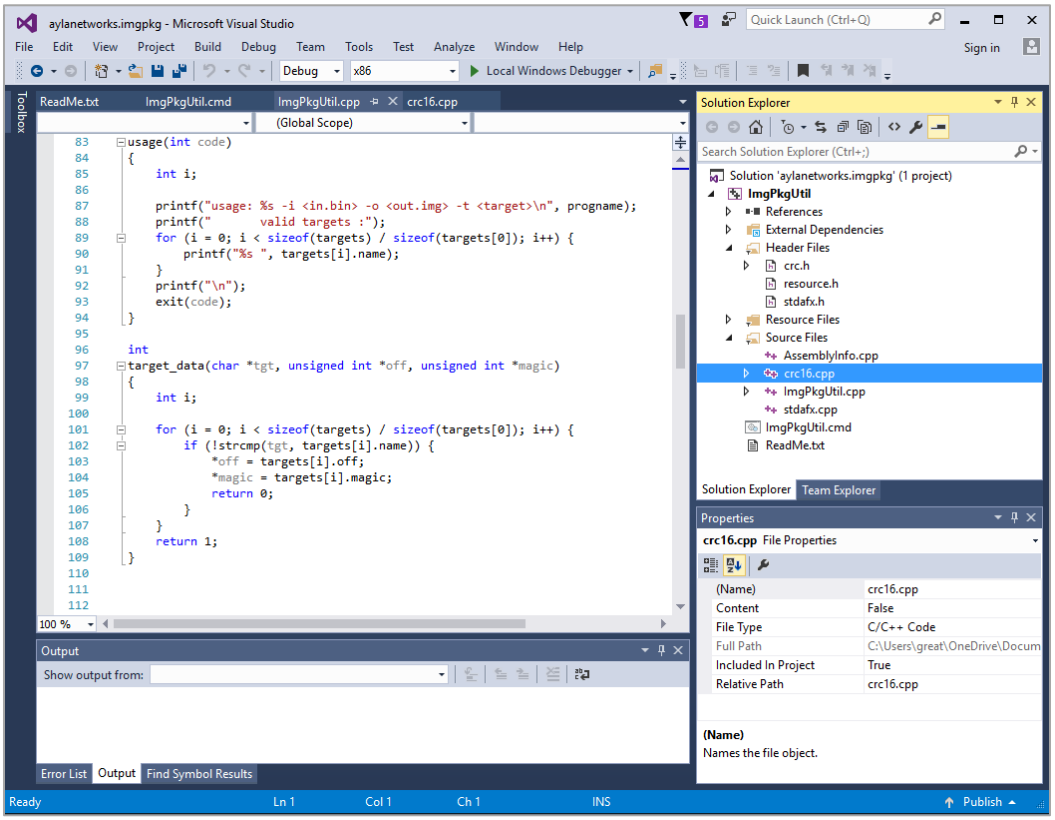
A free copy of the **Visual Studio 2017 (v15) Community Edition** can be downloaded from the following location:

https://www.visualstudio.com

After Visual Studio is installed, open the aylanetworks.imgpkg.sln file (Figure 10):

```
C:\Users\great\OneDrive\Documents\Ayla\demo-source-stm32f3-
1.6\Extracted\arch\VisualStudio\aylanetworks.imgpkg.sln
```

Figure 10 – Visual Studio UI



4250 Burton Drive, Santa Clara, CA 95054
Phone: +1 408 830 9844
Fax: +1 408 716 2621