

**Developers Guide**

**Innomatix Data Collection System with Coprocessor Support**

|  |  |
| --- | --- |
| Revision | 1.2 |
| Revision Date | September 6, 2017 |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Change** |
| 1.0 | Feb 2016 | Staff | Initial Creation |
| 1.1 | Apr 2016 | Staff | Updates after final integration |
| 1.2 | Sep 2017 | Staff | Update instructions to create a new app and import the Support library |

Table of Contents

1.0 Introduction 2

2.0 User Background 3

3.0 mbed Development Environment – Initial Setup 3

3.1. Creating an Account 3

3.2. Programmable Coprocessor support library 5

4.0 Programmable Coprocessor Support Library 8

4.1. Metadata support 8

4.2. Firmware Update 9

4.3. Remote DataStore 9

4.4. Auxiliary CAN Networks 10

4.5. Example Application 10

5.0 Development Process 11

5.1. USB via DAP-III+ 11

5.2. Over The Air 12

6.0 Debugging 12

6.1. Remote DataStore 12

7.0 Over The Air Updates 12

8.0 Frequently Asked Questions 15

9.0 Troubleshooting 15

1. Introduction

This document is for users who will be creating or maintaining applications for the Innomatix DAP-III with Programmable Coprocessor support.

Please refer to the *Innomatix DAP3 Quick Start* and *Innomatix DAP3 Coprocessor Quick Start* guides for information about connecting and confirming operation of your DAP-III with Programmable Coprocessor support.

The “coprocessor” portion of your DAP-III with Programmable Coprocessor support is an ARM Cortex M3 core microcontroller in the form of a “mbed NXP LPC1768” module. The module includes 32KB RAM, 512KB FLASH, Ethernet, USB Host/Device, 2 CAN networks plus a variety of common microcontroller peripherals.

The following diagram provides an overview of your DAP-III with Programmable Coprocessor system.

Key features provided by DAP-III with Programmable Coprocessor include:

* Data Store – The Data Store is a central repository of data internal to the DAP-III+. The Data Store contains values for every signal received from a CAN message as well as every value calculated by the DAP-III+. The Programmable Coprocessor uses a simple Set / Get interface to retrieve values from, and publish values to, the Data Store.
* Display Manager – The Display Manager controls what Data Store values appear on the display. By updating values in the Data Store, the Programmable Coprocessor can provide information to the user.
* Host Communications – The DAP-III+ communicates with the Innomatix Data host for several reasons. The primary reason is to report data collected by, and calculations performed by, the DAP-III+ to the host for users to evaluate. By publishing values to the Data Store, the Programmable Coprocessor can have values parsed from the Auxiliary CAN networks, values calculated by the coprocessor and even status reported by the coprocessor included in the data stream sent to the host.
* System Update – The System Update functionality allows over-the-air updates to the Programmable Coprocessor application software.
* CAN Network – The Programmable Coprocessor support library provides functionality to transmit and receive messages on the coprocessor-attached CAN networks. The simple Init / Read / Write interface frees you from the details of using the mbed CAN interfaces.
* Simple development tools –
  + The mbed microcontroller development environment is hosted online freeing you from the need to install and configure compilers. The online environment provides functionality and conveniences comparable with modern workstation-installed development environments.
  + Applications are installed to the mbed microcontroller over-the-air by the Innomatix Data host, by installing from a USB drive. These mechanisms eliminate the cost and learning curve of in-circuit emulators and programming devices traditionally used for microcontroller application updates.

1. User Background

Users intending to create or maintain applications on the Innomatix Programmable Coprocessor should be familiar with the C and C++ programming languages.

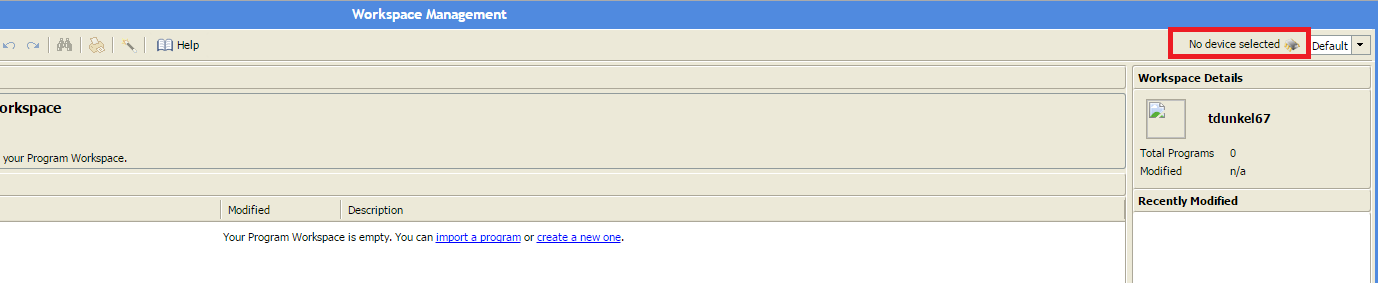
Understanding of CAN message formats is useful for those wishing to process CAN messages.

Users will need an account on, and familiarity with, the mbed online development website and the InnomatixData website.

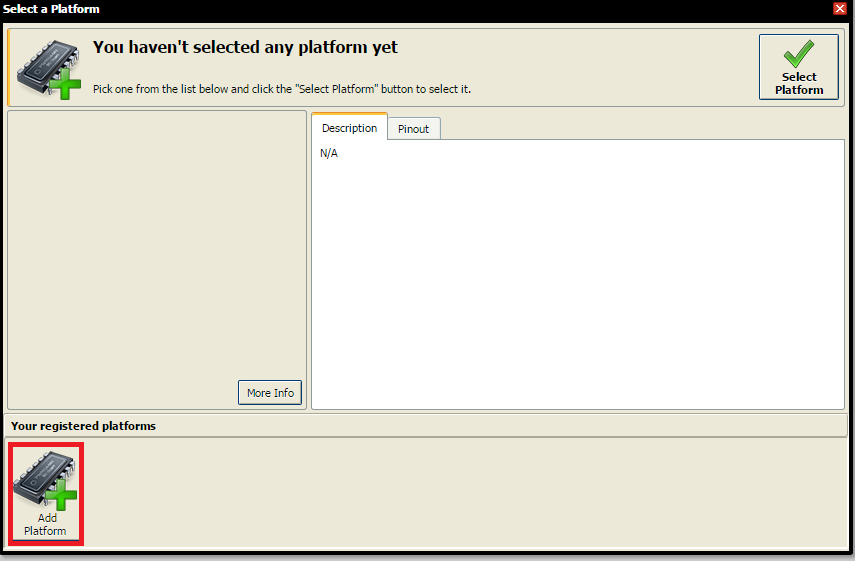
1. mbed Development Environment – Initial Setup
   1. Creating an Account

The mbed development environment is at: *https://developer.mbed.org/compiler*

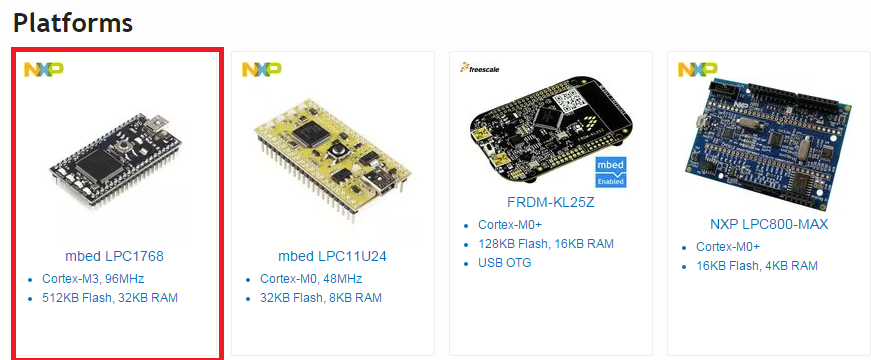
Create an account if you do not have one already. After creating the account you be taken to the development environment page. Here you will need to configure the environment for the mbed LPC1768 device. Click the “No device selected” in the upper right corner.



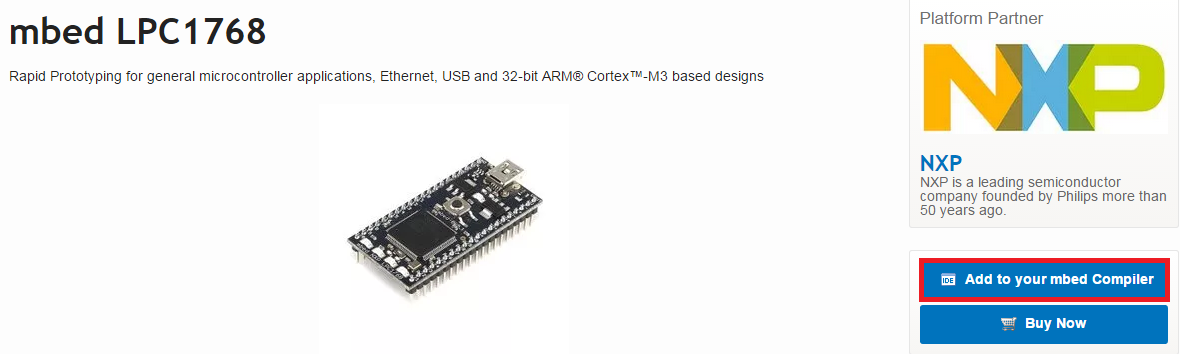
Then choose “Add Platform” in the lower left corner



Select the mbed LPC1768 device from the list.

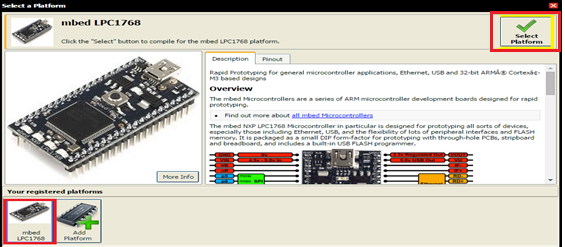


After selecting the mbed LPC1768, you will be redirected to a product page with information about the particular device. Choose the “Add to your mbed Complier” button on the right.



The page will refresh with a confirmation that the device was added to your compiler. Return to the compiler page and again choose the “No device selected” in the upper right corner.

Choose the mbed LPC1768 from the lower left.

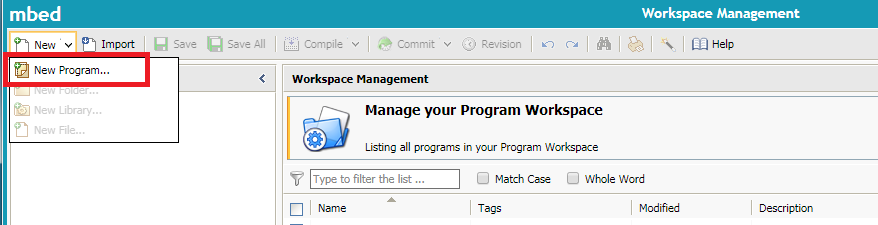


The main compiler page will now show mbed LPC1768 in the upper right corner.

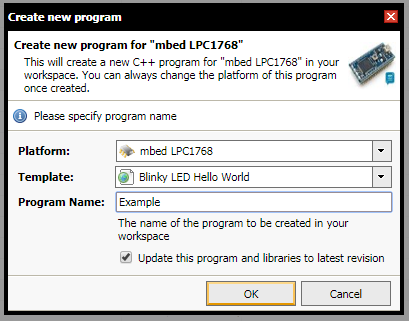
Your development environment is now configured and ready to use.

* 1. New Example Program

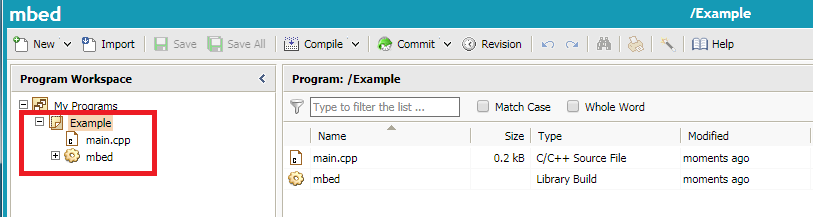
Development on the Programmable Coprocessor begins by creating a new program. Choose “New” then “New Program” from the menu



Provide a “Program Name” – we chose Example.



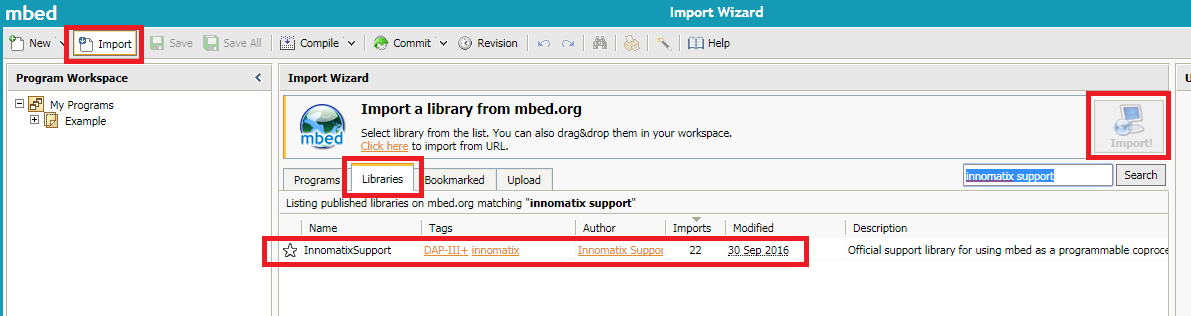
This will create a new, empty, program in your workspace. The program has a “main.cpp” file and it has the mbed core library.



* 1. Programmable Coprocessor support library

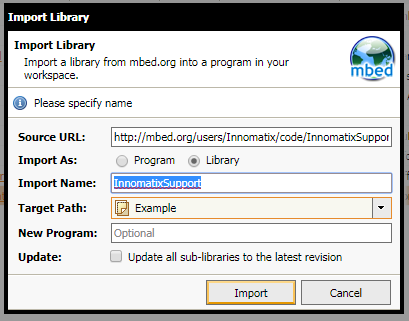
The mbed development environment has an integrated repository of programs and libraries provided by members of the mbed community. These programs and libraries provide a large variety of examples for how to use the various features of the mbed device. The community repository is available via the “Import” button on the main compiler page.

The Innomatix Programmable Coprocessor Support library is distributed via this community repository. The library can be located by choosing the “Import” button, then the “Libraries” tab and searching the repository for “Innomatix Support”.

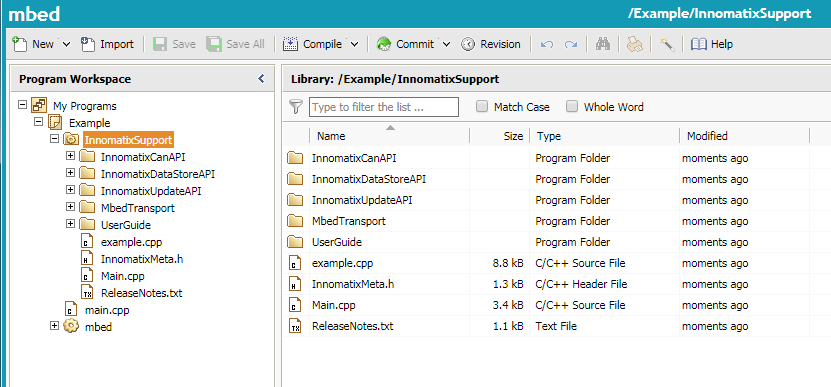


Highlight the entry in the search results and click the “Import” button above the library search button.

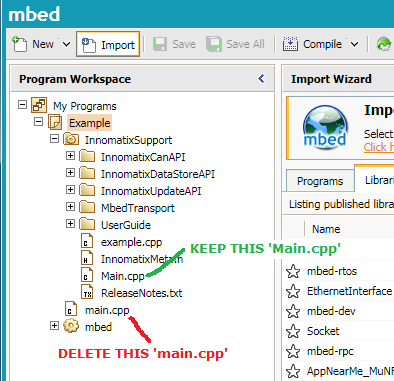
Import the support as a “Library”. Leave the “Import Name” as it is and ensure that the Example program is selected for “Target Path”.



The resulting project now includes the InnomatixSupport library in addition to main.cpp and the mbed core library.

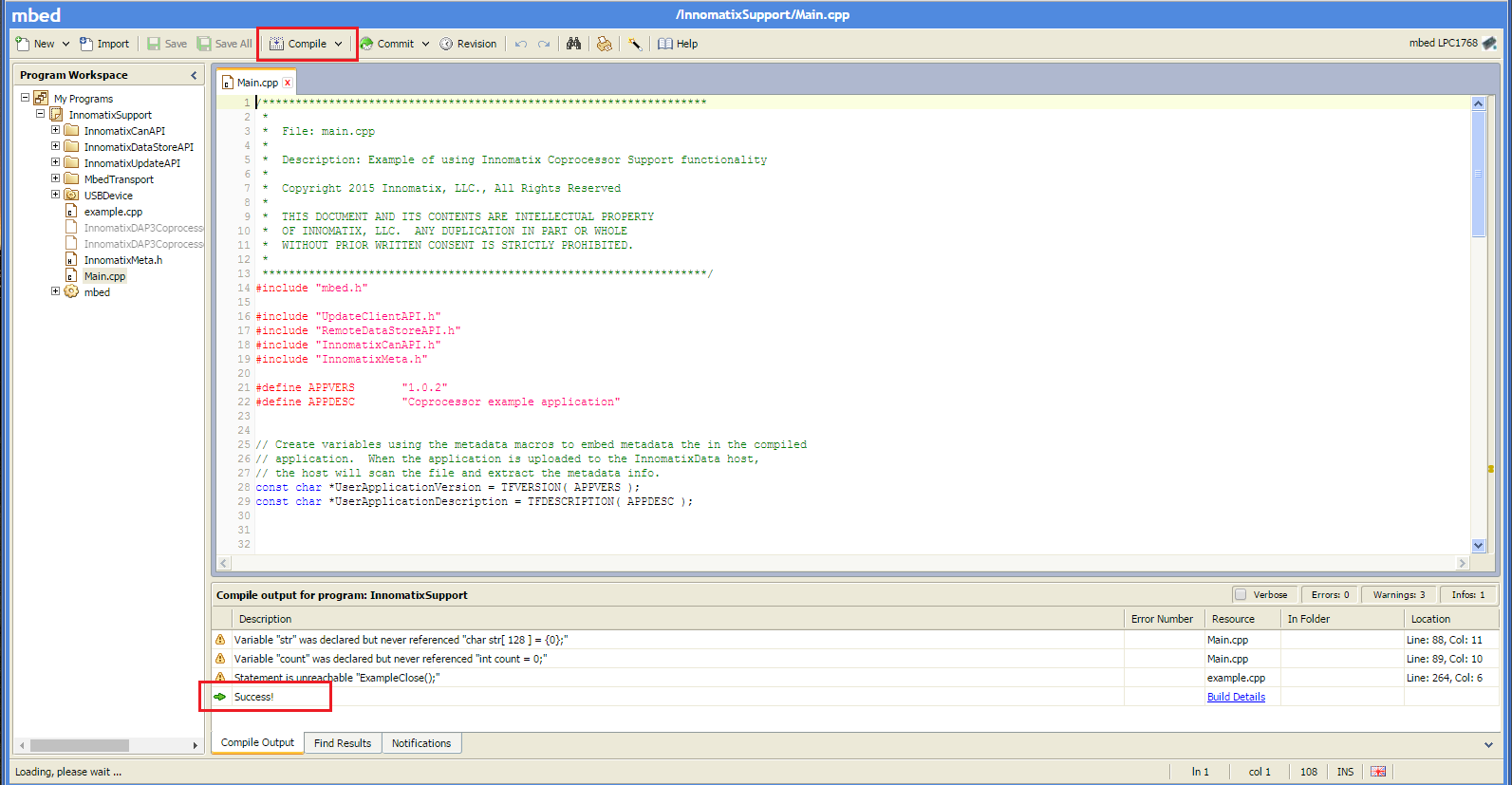


The resulting project also includes two copies of “main.cpp”. Remove the file created when the project was created and keep the file that was imported with the InnomatixSupport library.



The Programmable Coprocessor support library includes an example program which is now ready to build. Highlight the “Main.cpp” file and choose “Compile” from the menu.

If all goes correctly the output pane will show “Success” and, depending on your browser, the compiled program will begin downloading to your computer.



The compiled coprocessor application will be named according to the program name in the development environment. For our example, the file is named: *Example\_LPC1768.bin*.

This is the file that will be used to program the coprocessor.

1. Programmable Coprocessor Support Library

The *Innomatix Programmable Coprocessor Support Library* provides functionality to both integrate the Programmable Coprocessor into the Innomatix Telematics System and to simplify tasks for end-users developing applications for the Programmable Coprocessor.

The Innomatix Support library contains several API documents. The online mbed development environment will show these documents in the project files list but will not show the contents. The files can be downloaded from the online environment by right-click on the name and choose Save As from the pop-up menu.

The support library contains the following support subsystems.

* 1. Metadata support

The support library contains functionality for creating metadata which is embedded in the compiled application. This metadata is used by the InnomatixData host to assist with system configuration and revision control.

The end-user application must provide the following metadata values:

* TYPE – This value identifies the type of coprocessor the binary is for. For the mbed, the type is “LPC1768”. This value is required and is set for you in the InnomatixMeta.h file included with the example.
* VERSION – This value identifies the firmware version. It is expected in the format “x.y.z”. This value is required and must be set by the end-user application. In the example program, the value is set at the top of the Main.cpp file.
* DESC – This metadata provides a general description of the application. There is not a specific limit on the length but it should be short – approximately 80 characters maximum. No specific format or content is specified. This value is required and must be set by the end-user application. In the example program, the value is set at the top of the Main.cpp file.

For information on using metadata, see the example application and the header file in the example project.

./InnomatixMeta.h

* 1. Firmware Update

The support library provides functionality for updating the coprocessor application firmware from the DAP-III+. Updates can be performed “over the air” by uploading the coprocessor application file to the InnomatixData website, or can be performed “locally” via a USB drive.

See section *5.0 Development Process* for information about performing a coprocessor application updates.

See the *Programmable Coprocessor Example Application* source code for an example of integrating system update functionality into your application.

* 1. Remote DataStore

The Remote DataStore (RDS) library allows a Programmable Coprocessor to exchange data with the DAP-III+ application. This functionality allows the coprocessor to receive data values from the DAP-III+, perform calculations on the values, and publish new values back to the DAP-III+.

Values published to the Remote DataStore can be displayed in the BlueView signal list, can be streamed to the InnomatixData host and can be used to trigger captures of CAN network traffic.

The *Programmable Coprocessor Example Application* source code demonstrates this type of read/calculate/publish scenario.

The Remote DataStore API reference is included in the support library.

./InnomatixDataStoreAPI/Innomatix Remote DataStore API.docx

./InnomatixDataStoreAPI/inc/RemoteDataStoreAPI.h

* 1. Auxiliary CAN Networks

The CAN library provides a simplified interface to the coprocessor’s CAN networks. The API provides functionality to initialize, transmit, receive and close the network plus basic statistics about the network.

The API does not provide any functionality to build or parse CAN messages. Its main purpose is to provide a simpler interface for the end user than is provided by the native microcontroller development system.

To keep the *Programmable Coprocessor Example Application* as simple as possible, the example does not use the CAN API. There is a separate usage example for the CAN API in library.

The CAN API example and reference are included in the support library.

./InnomatixCanAPI/Innomatix Coprocessor CAN API.docx

./InnomatixCanAPI/inc/InnomatixCanAPI.h

./InnomatixCanAPI/can\_example.txt

* 1. Example Application

The support library includes an example application to demonstrate the various functionalities of the library. See the example application in the support library project at:

./main.cpp

./example.cpp

./Innomatix Coprocessor Quick Start.docx

The flow of the example application is:

Initialize System Update functionality

Check for and perform a system update

Clean up System Update functionality

Reboot if an update was performed

Initialize Remote DataStore functionality

Retrieve information about data bins used in the example

While running

Retrieve the value of the “VehiclePower” data bin

If the value is updated since last time we checked

If the value >= threshold

Publish value of 1 to the “PowerStatus” data bin

Else

Publish value of 0 to the “PowerStatus” data bin

End

End

Sleep()

1. Development Process

See section *3.0 mbed Development Environment – Initial Setup* for information about preparing the online mbed development environment.

Developing and installing an application for the Programmable Coprocessor is fairly straight forward but does involve a few steps.

1. Use the online development environment to write your application code which includes the Programmable Coprocessor Support Library.
2. Compile your application, resolving any compile warnings and errors as you go.
3. Download the compiled application from the mbed online development environment
4. Install the application onto the DAP-III+. See below for installation options.

There are two options for installing the application onto the Programmable Coprocessor.

* 1. USB via DAP-III+

This method is recommended for updating the coprocessor during development without using the InnomatixData website.

1. Rename the downloaded Programmable Coprocessor application to “mbupdate.bin”
2. Copy mbupdate.bin to the root of a USB flash drive, preferably with nothing else on it.
3. Install the flash drive into the USB Host connector of the DAP-III+.
4. The DAP-III+ will detect the flash drive, detect the Programmable Coprocessor application, copy the application to the DAP then restart to initiate reprogramming of the coprocessor. Watch the BlueView display for status and to determine when the process is complete. In particular, remove the USB drive when the display indicates that the DAP-III+ has shut down.
   1. Over The Air

This method has a few more steps and may take a little longer but does not require physical access to the DAP-III+ to perform the update. In fact the DAP-III+ does not even need to be turned on at the time that the update is initiated.

1. Upload the Programmable Coprocessor application to the InnomatixData host. See section *7.0 Over The Air Updates* for details on performing this step
2. Initiate an “update” of the desired DAP-III+ device and choose the desired Programmable Coprocessor application.
3. The update will be sent to the DAP-III+ the next time it communicates with the InnomatixData host. The Programmable Coprocessor will be updated based on the “Choose when to apply upgrade” setting during the update.
4. Debugging

Debugging facilities on the mbed are somewhat limited to begin with, but even more so because it is built into the DAP.

* 1. Remote DataStore

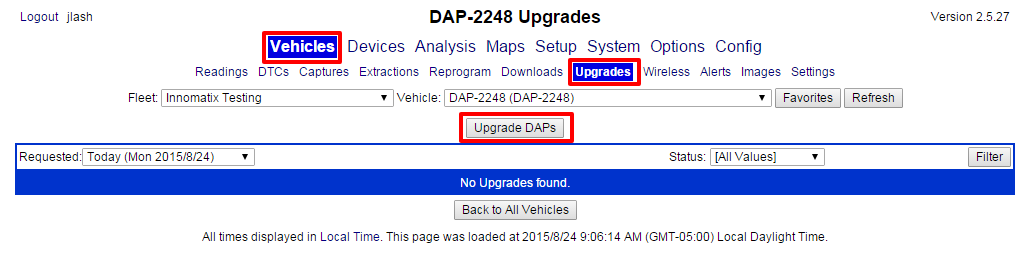
The Remote DataStore library provides a diagnostic data bin which your mbed application can update. The contents of the data bin are displayed on the DAP’s diagnostic display as the signal “CoprocDbg”, and are sent to the Innomatix Data host.

To use the diagnostic data bin in your coprocessor application, simply initialize the Remote DataStore library then call the function PutDebug( string ) with the desired string. The example application makes use of this feature.

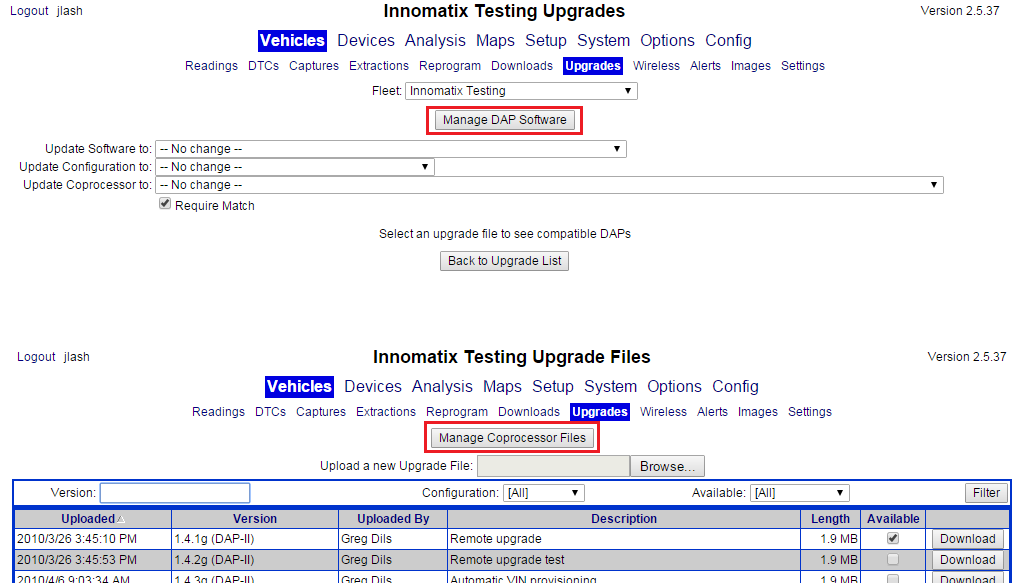
Your application can add as many other data bins as you desire for debugging purposes. The data bins you add to the coprocessor must also be added to the DAP-III+ configuration. Don’t forget to have the configuration put the data bins on the display and/or stream them to the host.

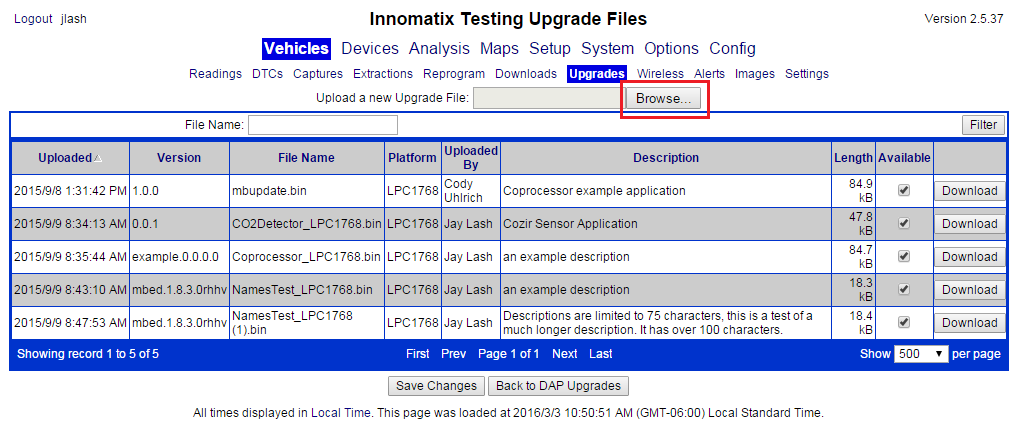
1. Over The Air Updates

The first step of installing your application over-the-air is uploading it to the Innomatix Data website. Log in to the site and navigate to the Upgrade DAPs page.



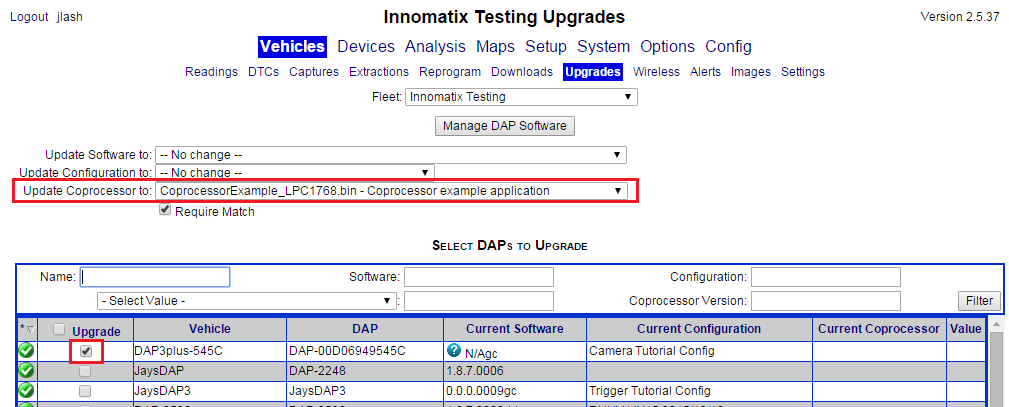
Then to the Manage Coprocessor Files page



Choose to Browse for a file, locate the application from the online development environment and upload it.

After uploading, the file is scanned for metadata. See section *4.1 Metadata support* for information about metadata used by the host.

After the application is uploaded and scanned, the page will refresh back to the Upgrade DAPs page. Choose the new Coprocessor application from the “Update Coprocessor To:” list. Select the DAP to be upgraded from the table and choose to begin the upgrade.



When the DAP communicates with the InnomatixData host, the upgrade will be transmitted. The upgrade will take effect according to the “when to apply upgrade” selection chosen when the upgrade began.

1. Frequently Asked Questions

|  |  |
| --- | --- |
| Q: | Is the online mbed development environment also a debugger? |
| A: | No, the online environment is for editing and compiling only. It does not have any debugging capability. |
|  |  |
| Q: | Since the online mbed development environment is not a debugger, how can I debug my application |
| A: | See section *6.0 Debugging* |

1. Troubleshooting

See the mbed online community for assistance with developing applications on the mbed.