

MMCOMMANDER

What's this?

An interface for Medtronic CGM devices.



I've created this document to help you build and configure your MMCommander for your Medtronic device.

Note - currently MMCommander only supports the Medtronic MiniMed Veo or 530 pump with MiniLink CGM transmitter system. The Medtronic MiniMed 640G pump with Guardian 2 CGM system is not supported by this system.

MMCommander works by listening to the data being sent wirelessly from the Medtronic CGM transmitter to the pump. Then it sends that data to an Uploader App running on an Android device, which in turn sends the data (either via Wi-Fi or the Mobile Phone Network) to a Nightscout website in the cloud.

The Nightscout website can then be viewed on any web connected device - your tablet, computer, another mobile phone, watch (Pebble or Android Wear), etc.

MMCommander is only capable of passively receiving data packets, not transmitting. This makes this version totally safe - it won't be able to interfere with the pump or sensor at all.

Oops... I nearly forgot.

As you can imagine, Medtronic is not related to this project in any way and the author (me) assumes no responsibility for any damage arising from use of any of the information found in this repository.

USE IT AT YOUR OWN RISK.

Information obtained via this software MUST NEVER BE USED to take medical decisions.

What is the MMCommander

The MMCommander is composed of three things:

1) CC1111 USB Dongle



<http://www.ti.com/tool/cc1111emk868-915>

This is a USB RF transceiver dongle made by Texas Instruments, and is used to "listen-in" on the data packets being sent wirelessly over-the-air between the Medtronic CGM and the pump.

2) CC Debugger



<http://www.ti.com/tool/CC-DEBUGGER>

The CC Debugger is a device which you'll only need in order to download software to the CC1111 (and after that you can share it with others if you want).

The cost of the whole MMCommander pack at October 2015 is:

- \$75 for the CC1111 on TI's website (also available from other distributors)
- \$49 for the CC Debugger on TI's website

3) MMCommander software

Lastly, you need the software found in this repository.

What else do I need?

Apart from the MMCommander, to use the Nightscout Uploader App you'll need an Android device that supports USB OTG.

Note that not all Android devices support USB OTG, so check carefully.

You'll also need a suitable USB-OTG cable (Micro-USB male 90 degree connector to regular USB female socket) to connect the Android device to the MMCommander transceiver.

Note with OTG it's not possible to use a USB "Y" splitter to charge the phone and power the CC1111 at the same time.



For further information about what you need, check the Nightscout webpage <http://www.nightscout.info/wiki/welcome/basic-requirements>

Bear in mind that Nightscout originated using the Dexcom CGM system, so lots of the information on the Nightscout Website relates to this.

Nightscout 'cloud' services setup

Once you've ordered all the items you need, while you're waiting for them to arrive you should set up the cloud services that are needed for Nightscout. These steps may take a while, but just keep going - you'll get there in the end.

Follow the process for configuring the MongoLab data backend, as described in the Nightscout guide:

<http://www.nightscout.info/wiki/welcome/configuring-the-data-backend>

Note - don't continue to the page about "Installing the Android App", as we will deal with that separately.

After that continue the process for setting up the Azure website, again, as described in the Nightscout guide:

<http://www.nightscout.info/wiki/welcome/monitor-cgm-web>

Note - ensure that the Azure website does not have authentication of any form turned on, as this will prevent data uploads from working.

Make a note of the following items, as you will need them for configuring the Nightscout Uploader App later on:

- Your Azure Web site address
- Your API_SECRET

Where can I get further help?

If you get stuck or want more information (e.g. what mobile phone plan are people using in your country), you can get support from the following Facebook groups:

CGM in the Cloud <https://www.facebook.com/groups/cgminthecloud/>

The original Nightscout support group

Nightscout for Medtronic <https://www.facebook.com/groups/Nightscoutformedtroincusersuk/>

Although UK based, this group has members from outside the UK too

Also if you search on Facebook there are several country specific groups for Nightscout. For example

Nightscout España <https://www.facebook.com/groups/774263866005262/>

Nightscout Italia <https://www.facebook.com/groups/1535088246753506/>

Nightscout Romania <https://www.facebook.com/NightscoutRomania/>

Nightscout New Zealand <https://www.facebook.com/groups/875614605859053/>

...and many more.

Feedback

If you have any comments on these instructions or suggestions for improvements, please post on the Facebook groups.

Now that I have the hardware... What do I do next?

You need to install the software that Texas Instruments provides to program the CC1111.

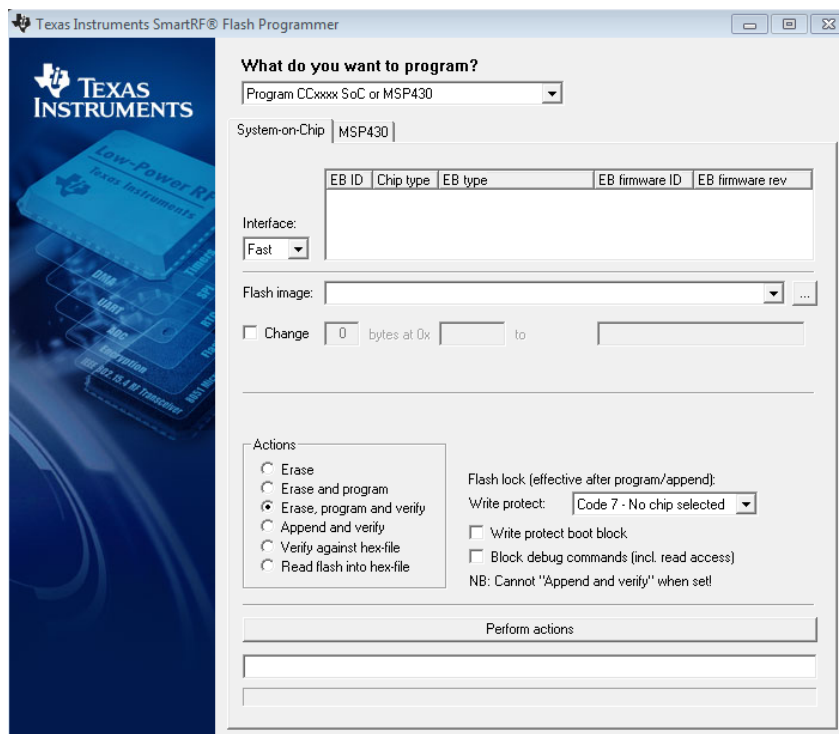
It's called SmartRF Flash Programmer and can be downloaded from here: <http://www.ti.com/tool/flash-programmer>

Note - there are now two versions of this software. You need to choose the "SmartRF Flash Programmer" (which is for older 8051 based modules like the CC1111) - not "SmartRF Flash Programmer 2" (which is for newer ARM based modules).

Unfortunately SmartRF Flash Programmer is only available to run on Windows.

However you can still use this if you have a Mac or use Linux (Ubuntu, Debian, Fedora, etc.), by installing Windows as a Virtual Machine using tools such as VMWare, VirtualBox, or Parallels Desktop for Mac.

SmartRF Flash Programmer should look something like this.



Connect the CC Debugger and CC1111

Connect the CC Debugger to the CC1111 board using the ribbon cable that comes with the Debugger (connecting to CC1111's "Debug" connector, not the "Test" connector).

Pay close attention to the red line on the ribbon cable marking pin 1: connect the cable as you can see in the following picture.

Note: some people have reported their cables were incorrectly assembled and they had to connect the ribbon cable with the red line facing the opposite side. Some people have even found the cable would not work whichever way it was connected (if that's you, see the Troubleshooting section for help). Don't be afraid to test, you won't break anything.

Plug both CC Debugger and CC1111 USB connectors into your computer.

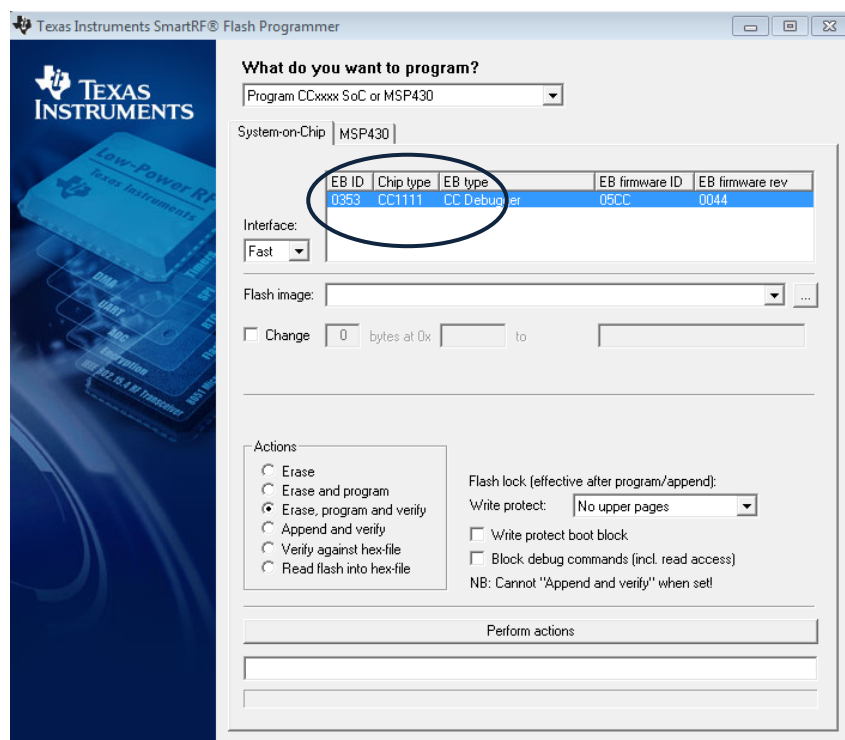


The CC Debugger should be detected by Windows and its USB driver should be installed automatically.

Note - Some CC Debuggers are shipped with an old Firmware version and the SmartRF Flash Programmer will ask you if you want to update it. You can do it if you want, but it's not mandatory. I would only do it if you have problems programming the CC1111 - up to you. (See the section near the end of this document for how to do this).

Now check the LED on the CC Debugger. If the LED is red, detection of the CC1111 failed - press the reset button on the CC Debugger and the LED should turn green.

If the LED is green, the Debugger has successfully detected the CC1111, and the CC1111 should appear in the devices list:



Note - The CC1111 only appears in the devices list if the CC Debugger is correctly connected with the ribbon cable, and you've pressed the reset button on the CC Debugger and have a green LED. It does not appear here just because the CC1111 is plugged into a USB port, or when the CC Debugger LED is red.

If the CC Debugger LED is still red, something is not connected correctly. Check the ribbon cable, its orientation, and that both USB ports are properly connected to the computer (both CC Debugger and CC1111 need USB power). Take a look again at the earlier picture to check.

If you still can't get it to work see the Troubleshooting section for further help.

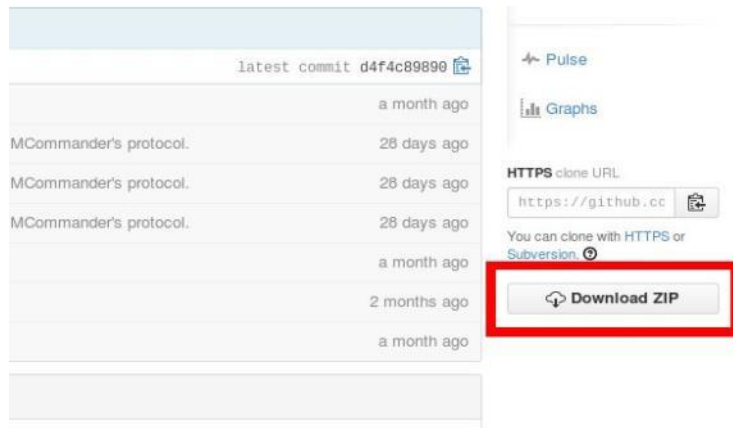
Note - if the LED is amber or flashing red that means something is wrong with the Debugger itself. See TI's CC Debugger User Guide for help: <http://www.ti.com/lit/pdf/swru197>

Download the MMCommander software

Is it working? Well... let's download the MMCommander software (if you haven't already).

If you're not used to downloading info from GitHub I would recommend downloading the complete repository in a ZIP file.

Go to the repository's main page <http://github.com/jberian/mmcommander> and click on the “Download ZIP” button on the right side of the page.



Decompress the ZIP file and, in the “hex” folder, you'll find two HEX files:

- MMCommander_EUR_0.89_NoTx.hex
- MMCommander_US_0.89_NoTx.hex

These files are the result of compiling the sources for US and European frequencies WITHOUT transmission (Tx) enabled. This is the safest implementation possible and using these binaries you won't be able to affect in any way your pump or sensor. You'll be a passive listener and will be able to listen to your sensor and glucometer.

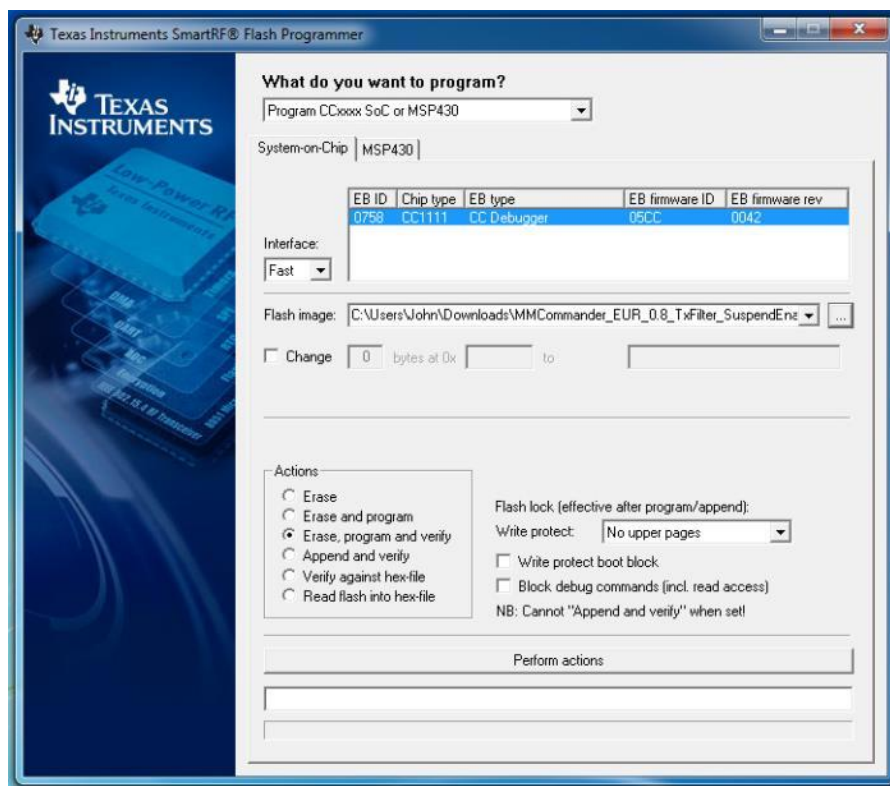
Now that you know which one is the one you need, get the HEX file and let's go back to the SmartRF Flash Programmer and program the CC1111.

Programming MMCommander into the CC1111

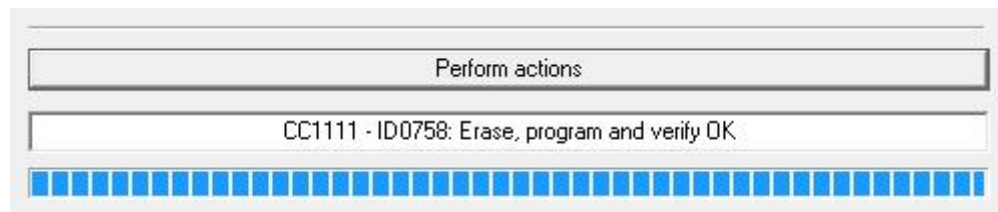
Now everything is connected, the LED on the CC Debugger is green... It's time to configure the SmartRF Flash Programmer to program the CC1111.

Check list:

- Choose the 'Program CCxxxx SoC.....' pull-down
- Select 'System-on-Chip' tab
- If everything is correctly connected you should see the CC1111 in the devices list
- In "Flash image" click the button with the 3 dots and select the MMCommander HEX file you downloaded
- In actions: choose Erase, program and verify



After configuring the programmer, click on "Perform actions" and wait until you see this:

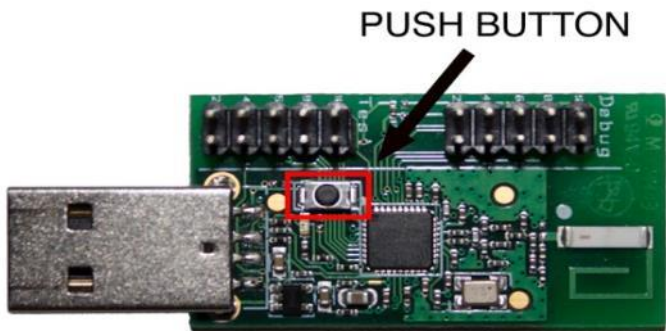


At this moment your computer should detect a new "MMCommander" device.

Although Windows may start trying to install a driver for the device, we aren't going to make use of that here so you can cancel that dialogue.

Test MMCommander / CC1111 dongle

To test, just plug the MMCommander into a USB port and hold the pushbutton down for a few seconds. If the LED turns on and, after pressing the pushbutton again for another few seconds, it turns off then it's ready ;-)



Congratulations! You have a new MMCommander!

Installing and Configuring the Nightscout Uploader App for Medtronic

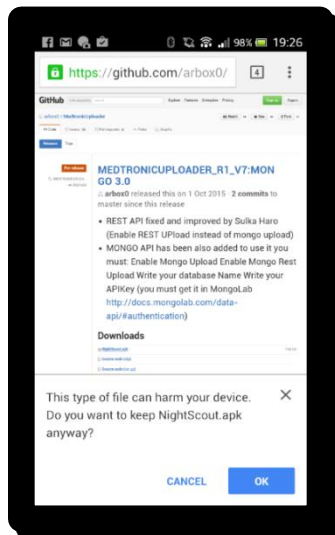
Next you need to install the Nightscout Uploader for Medtronic onto your Android device.

Note - the Android Uploader for Medtronic is not on the Google Play store - there is a Nightscout App there, but that one is the Dexcom version. (Don't use that one, and if you have already installed it you'll need to uninstall it before going any further).

Installing the Uploader

You can either install the Nightscout App directly on the Android device itself (as described in the instructions below), or if you prefer you can download it to your PC first, then copy it to the device's Download folder and install it on the device (for example using another App such as AppInstaller which you can install from the Google Play store).

1. As you will be installing an App from somewhere other than the official Google Play store onto your device, you need to enable "installation of software from unknown locations".
 - Go to Settings > Security, and find "Unknown sources". Turn on the setting that allows installation of apps obtained from unknown sources. There's a warning message, which you need to OK.
 - *Note: you can turn this off again after you have completed the installation.*
2. Now on your uploader device, download the Android Uploader for Medtronic from this link.
 - <https://github.com/arbox0/MedtronicUploader/releases>
 - Note - Choose the most recent App file called "NightScout.apk", under Downloads. Don't worry if this says pre-release.
3. Tap on the file to download it. You may get a pop-up warning that this type of file can harm your device. Accept the warning.



Note - the most recent Nightscout App version may differ to the one shown in the screenshot

4. Once you have downloaded the NightScout.apk file, locate the file on your device and tap on it (for example, swipe down from the top of the screen to see recent Downloads, and tap on the filename).

Accept the "Do you want to install this application" prompt.

Next you need to configure the uploader.

How do I configure the uploader?

Let's do a step by step configuration.

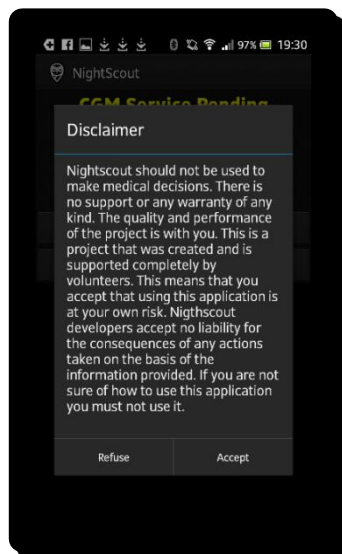
Note - this section is just a quick guide. For further details see

https://github.com/arbox0/MedtronicUploader/blob/master/Medtronic_NighScout_Manual.pdf

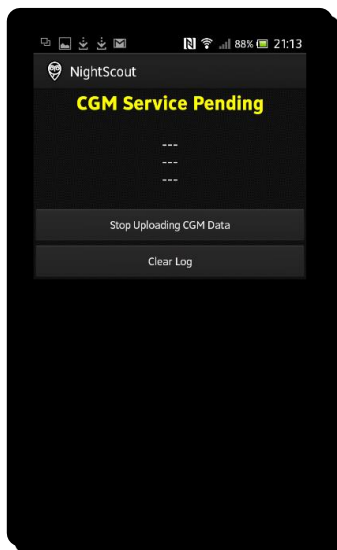
(yes there is a typo ;-)

1. Open the Nightscout uploader App. Note - there is no need to connect the MMCommander via the USB OTG cable at this stage.

Read the pop-up Disclaimer notice carefully and accept it if you want to continue.



2. Next you should see a Yellow message saying “CGM Service Pending” which means you need to configure the App settings:



3. Open the Preferences menu and configure the uploader:

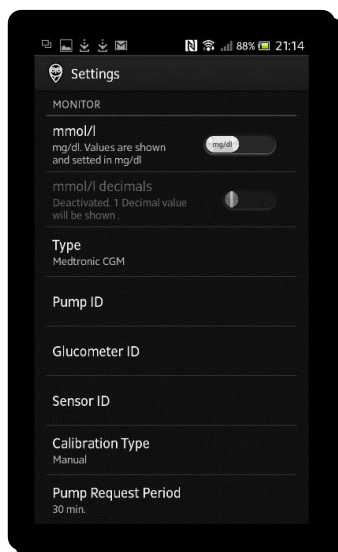
Note - depending on your Android device, 'Preferences' may be accessed either via an on-screen button (e.g. three dots in the action bar at the top of the screen, or elsewhere) or your device may have a physical button – usually the same button as settings.

Some items in the Preferences are no longer in use, so if you see an entry on screen that is not mentioned in the list below, you can safely ignore it.

Under 'Monitor' section

- mmol/L -> Select the units you want to use - mg/dl or mmol/l.

Note this setting must match the units you have set on your pump, otherwise the values uploaded will appear to be out of range and you will receive errors on the Nightscout website.



- Type -> Select "Medtronic CGM"

- Pump ID -> The ID on the back of your pump
 - Enter just the digits (numbers) from the Serial Number. For example, if your serial number is SN PBR123456H, just enter 123456 (see the red rectangle in the picture)



- Glucometer ID -> If you have a Blood Glucose meter that is connected wirelessly to the pump, enter its Id, also configured in your pump
 - Enter all the digits and letters from the Serial Number. For example, C1AB2C (see the red rectangle in the picture)



- Sensor ID -> The ID on the back of the MiniLink transmitter, also configured in your pump (see the red rectangle in the picture)
 - Enter just the digits (numbers) from the Serial Number. For example, if your serial number is SN GT1234567N, then just enter 1234567 (see the red rectangle in the picture)



- Calibration Type -> Choose “Manual”
- Glucose Value Source -> Choose “Medtronic Sensor” (not Both)

Under 'Sharing' section

Note - These instructions assume you are following the recommended / most common route of using Azure to host your website. If you are using Heroku (or another provider), you will need to use other settings, and currently you'll need to ask on the Facebook groups for support. If you are reading this and have already set up with Heroku, please add to these instructions!

- API Upload (REST) -> turn this On

- API Base URL -> Enter

[YOURAPISECRET@https://YOURWEBSITE.azurewebsites.net/api/v1/](https://YOURWEBSITE.azurewebsites.net/api/v1/YOURAPISECRET)

Note - precise formatting of this URL is essential, including all the special characters (forward slash, colon, etc.). If anything is incorrect, the upload will fail.

- Make sure you enter “https” and not “http”
 - YOURAPISECRET is the "at least 12 character value" you set for the API_SECRET, using your own upper/lower-case formatting
 - YOURWEBSITE.azurewebsites.net is your Azure website address, using your own upper/lower-case formatting
 - Note - this URL format is different to that recommended for the Dexcom Uploader on the Nightscout.info website (as the uploader is different)
 - Double and triple check that the URL is exactly correct after entering it (e.g. there are no spaces, your device has not auto-corrected anything, etc.)
- MongoDB Upload -> ensure this is Off
- MongoDB REST Upload -> Ensure this is DEactivated

Other settings

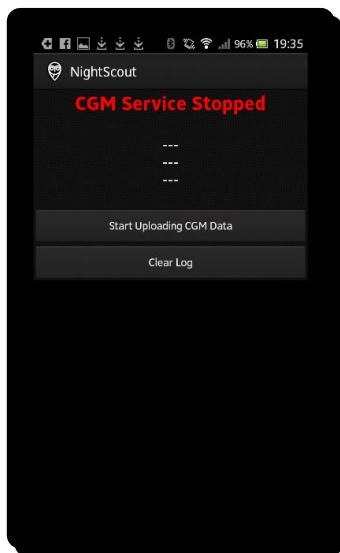
- Wifi Hack -> Set it to OFF
- 2 Days at Startup -> Set it to ON
- Disclaimer -> You must accept this final disclaimer too and set it to YES. The uploader won't start the service until you accept it.

Under 'Debug'

- Logging Level -> Error

Set up is now complete, so exit the Preferences menu.

4. When you launch the Nightscout App without the USB dongle being connected, you may see a 'Connecting...' message and then should see a notice saying the CGM service is stopped:

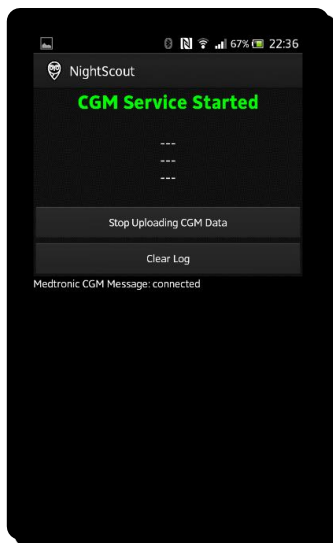


5. Now connect the MMCommander dongle via the USB OTG Cable.

Your device may prompt you whether the Nightscout uploader App should be automatically started and be associated with the connected device. This capability may depend on your device or settings.

If the CGM service does not change status automatically, tap on "Start Uploading CGM Data" which should result in a pop-up asking you whether to allow the Nightscout application to access the USB device. Select OK.

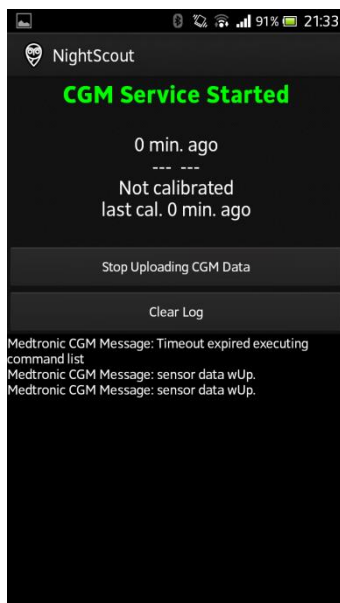
After a few seconds the status message should change from Connecting to CGM Service Started. A message should appear saying "Medtronic CGM Message: connected".



6. If you still see a CGM Service Pending screen, ensure that you have entered all the settings correctly, and check that your web site has been configured correctly.

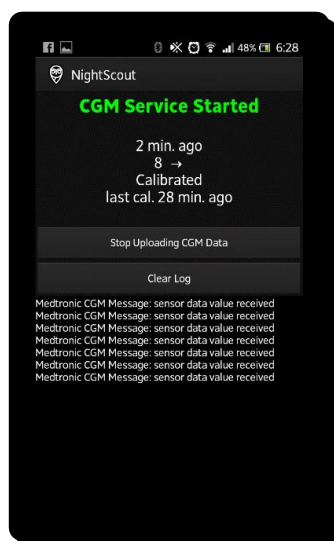
7. If you started the Nightscout App when the Enlite CGM sensor was still within its warm-up period, you will need to wait until the warm up period ends.

During the sensor warm up period you should see messages saying "sensor data wUp"



8. After the Enlite sensor has completed its warm-up, once the uploader receives data from the sensor you'll see a text line in the log (lower part of the screen) that says: "Medtronic CGM Message: sensor data value received".

Don't do anything else until you see that message for the first time.



9. The MiniLink CGM transmitter sends a message approx. every 5 minutes, so if you still haven't seen any sensor data value log messages after about 10 minutes, go back and check the configuration.

Note - although the MMCommander should work up to a maximum range of around 9 meters, it might be much less depending on your local RF environment. Also human bodies are very good at absorbing RF.

In case you still don't receive anything, I've added a section on debugging the MMCommander at the end of this document.

10. You have received the first data from the sensor! As soon as your pump starts displaying glucose values from the sensor, from the Nightscout uploader menu select "Instant Calibration" and enter the BG value displayed on your pump. After that, on the main screen, you should see your actual BG and that the App is calibrated.

11. Once you have seen a few values being received, check your web site to see if data is being displayed. In your web browser, navigate to the web site that you previously configured for Nightscout:

https://YOUR_WEBSITE.azurewebsites.net

12. At this point, your Mongo database should be recording data from the uploader. If you want, you can check it by looking at the number of documents containing "sgv" values that are stored in the "entries" collection.

Each time a sensor message is received the number of documents should increase (one unit).

- Note - when you open one of those documents, you should see numbers for the SGV. For example "sgv": "95". However if instead you see blank entries "sgv": "---" then you should perform calibration again.

Note - You should calibrate the uploader using "Instant Calibration" every time you reconnect the MMCommander.

Reminder: Did you want to turn off "installation of apps from unknown sources"? Go to Settings > Security > Unknown sources, and turn it off.

Android device configuration etc.

There are various settings on your Android device that will affect the Nightscout uploader App.

Note that these settings will vary between different Android versions and different devices, so you will need to research about your own device to find out what you need to do.

Wi-Fi, mobile data and power saving

Power saving features may sometimes disable mobile data and/or Wi-Fi in order to extend battery life.

It may be possible to add the Nightscout uploader App to a list of exceptions, or to turn off those features entirely if you need to, in order to ensure data can be uploaded to your Nightscout website.

There may also be separate Wi-Fi and/or mobile data settings to control whether these features are turned off when the device is in sleep mode.

Battery life

Be aware that the MMCommander will put some extra demands on the battery of the Android device.

Also if you haven't used CGM with your pump before, you may find its batteries are used up more quickly too.

More power saving tips here:

<http://www.nightscout.info/wiki/faqs-2/uploader/how-can-i-improve-the-battery-life-on-my-uploader-phone>

Alarms

Depending on the circumstances in which you use your Android device and the Nightscout uploader App (home, school, college, work, sleeping, shopping, holidaying...) you may want to turn audible and vibration alarms on or off, and volumes up or down.

Note - some of the above may also apply to the devices on which you are viewing the Nightscout website.

Nightscout website configuration

Finally you'll also need to configure alarm/alert settings that are specific to the Nightscout website itself:

<http://www.nightscout.info/wiki/welcome/website-features>

Now you have Nightscout up and running!

REMEMBER, USE IT AT YOUR OWN RISK

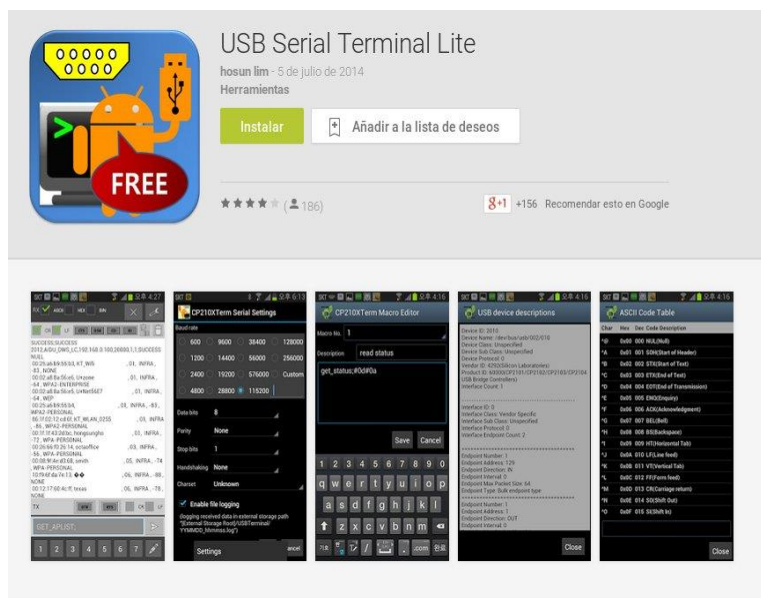
Troubleshooting

How can I test if my phone and the MMCommander are receiving data from the MiniLink transmitter?

If you're reading this it means you're probably having problems with the Medtronic Uploader and don't know how to continue... well...

Let's see if your phone and MMCommander are capable of receiving data. To test this we are going to use an application named “USB Serial Terminal Lite”.

Look for it in the Play Store and install it.



Once it's installed, connect the MMCommander with the USB-OTG cable to your phone and start the Terminal application. If the Medtronic Uploader launches itself automatically when you connect the MMCommander, press “Stop Uploading CGM Data” and go back to the Terminal App.

Click on the configuration icon on the right top corner and configure the serial port.

1. Baudrate: 57600
2. Data bits: 8
3. Parity: None
4. Stop bits:
5. Handshaking: None
6. Charset: Unknown
7. Enable file logging: unchecked

8. Press "Save".
9. After pressing "Save" you'll go back to the main screen.
10. Select "HEX" format on the top bar.
11. Press on the phone icon on the top right corner and it should transform into an "X" icon.
12. Select "RTS" at the bottom. They should highlight in green.

Now you should be able to see in hexadecimal format everything that is being received by the MMCommander.

1. First, let's check that the MMCommander was programmed correctly
 - Press the pushbutton until the green LED turns on. Once the LED is on, you should see "13" in the terminal.
 - If you press the pushbutton again, the LED should turn off and you should see "03" in the terminal.
 - If this happens, the phone, the USB OTG cable and the MMCommander are all working.
2. Now let's check if the RF part of the MMCommander is working...
 - Take your pump and go to "Utilities" -> "Connect Devices" -> "Other Devices" -> "On" -> "Find Devices".
 - You should be able to see some data on the terminal screen starting with "02 0B A2..." in periods of some seconds.
 - If you see this your MMCommander is working perfectly.
 - The MiniLink transmitter is also sending data roughly every 5 minutes, so it may be possible to see this info as well as the bytes from the pump. Anyway, if you see data from the transmitter (also starting with "02" or "82") it's working.
3. Still having problems?
 - If it still doesn't work, reprogram the CC1111. Maybe you selected the wrong hex file by mistake?
 - If you can see the "03" and "13" bytes when turning on and off the LED, but don't receive anything when testing the RF, try putting the MMCommander really really close to the pump. If you receive data when near, you have a RF problem and you have 2 options: resoldering the RF parts of the CC1111 (ask in the group) or buy another CC1111 stick.
 - The Uploader keeps a log file on the Android device, called "MedtronicFile.log". This may contain some useful information.
 - The file can be located in the top level of the Internal storage folder, either by connecting the device to a PC, or by using an App like File Commander to locate the log file on the device itself.

Note - you can alter the logging level via the Uploader's preferences menu - Error/Info/Debug (Debug gives the most information).

Hope it helps ;-)

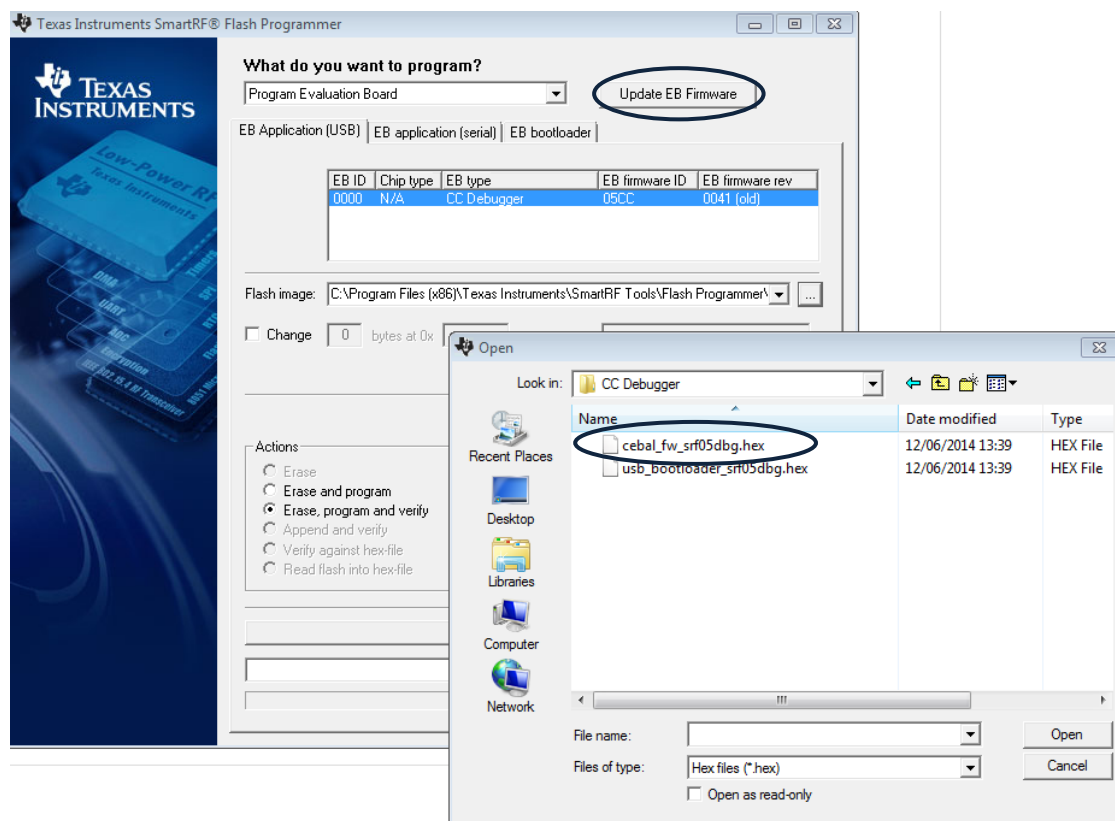
How to update CC Debugger firmware

First, just to be safe, disconnect the CC1111 from the CC Debugger (remove the ribbon cable).

In SmartRF Flash Programmer choose the 'Program Evaluation Board' pull-down, and click on "Update EB Firmware" (which is only enabled if the Debugger has older software), then find the Debugger firmware hex file.

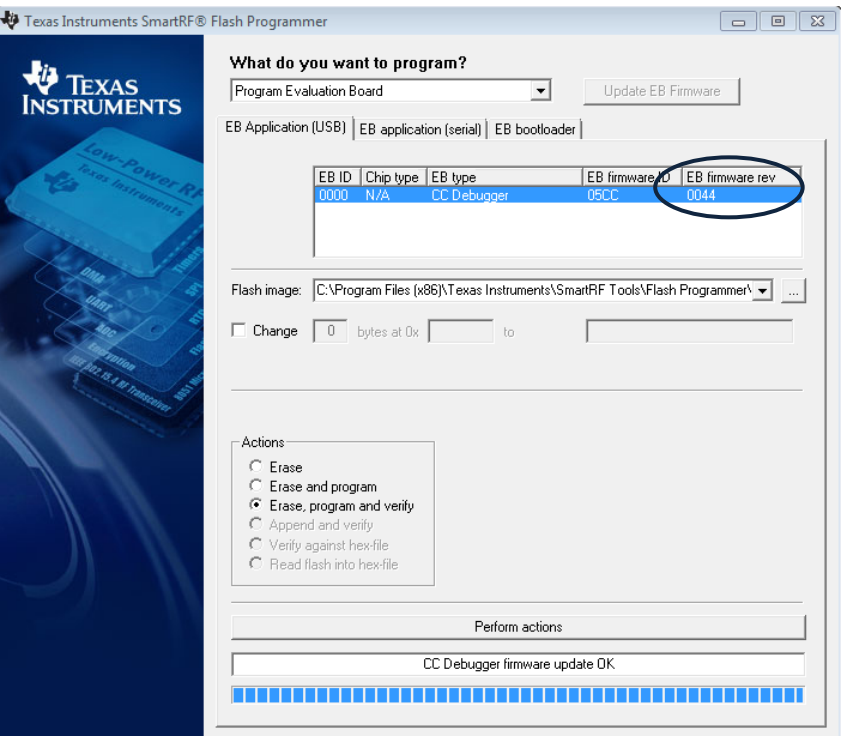
e.g. C:\Program Files (x86)\Texas Instruments\SmartRF Tools\Firmware\CC Debugger\cebal_fw_srf05dbg

(Note - your version may be different)



Make sure that you update the application (not the bootloader!).

After a successful update, you should see an updated "EB firmware rev" and "...update OK"



Troubleshooting the CC Debugger - CC1111 ribbon cable connection

Texas Instruments seem to have been sending out some ribbon cables where the connectors have not been assembled correctly.

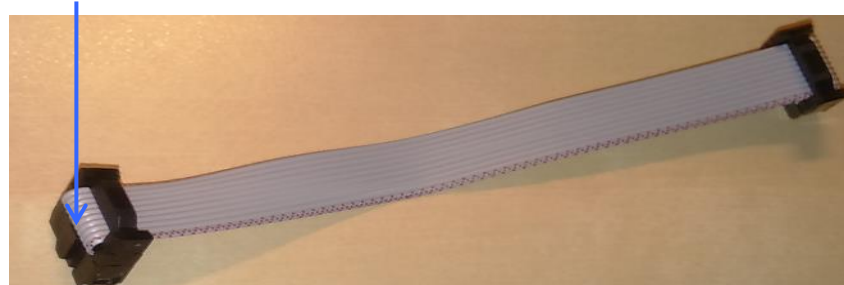
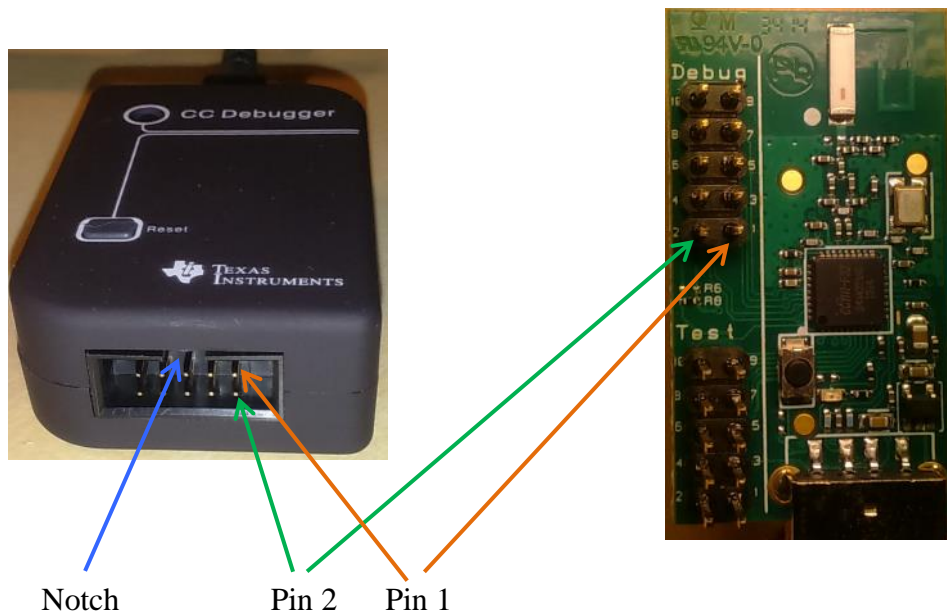
The first wire of the ribbon cable (often red or patterned to distinguish it from the others) should connect from Pin 1 of the CC Debugger to Pin 1 of the CC1111's debug connector.

Even if the cable has been assembled with the red wire going to Pin 10 instead, that doesn't matter provided it connects to Pin 10 at both ends. If that's the case, then everything should work.

However, you may find your ribbon cable has been assembled incorrectly and the CC Debugger does not connect successfully to the CC1111 whichever way around you plug in the cable (the CC Debugger has a polarised notch, which means the cable can only be inserted one way round at the Debugger), or how many times you press the Reset button.

If you're feeling brave, it's possible to remove one of the cable socket connectors and re-fit it the other way round using nothing more than a flat blade screwdriver to push the wires back onto the pins of the connector. Alternatively a small workbench clamp can be used to push down the cable cover and so force the ribbon cable back onto the pins. Obviously you will need to line everything up carefully first.

Another solution is to remove the locating notch from one of the cable's socket connectors, e.g. by using a modelling knife. This allows the cable to be inserted into the CC Debugger port in either orientation.



Note - the cable in this picture is an example of one that is incorrect and that needs to be modified

Glossary / Abbreviations

APK	Android application package	
BG	Blood glucose	<i>Usually referring to the value from a finger prick test</i>
CGM	Continuous Glucose Monitor	
Firmware	Software in electronic devices	
LED	Light emitting diode	
OTG	On-the-go	<i>USB feature</i>
RF	Radio Frequency	
SGV	Sensor glucose value	<i>Refers to the value derived from a CGM sensor</i>
SoC	System on Chip	
TI	Texas Instruments	
Tx	Transmitter / transmission	
URI / URL	Uniform Resource Identifier / Locator	<i>Path to a resource accessed over the internet / network</i>
USB	Universal Serial Bus	