Nightscout FDA presubmission

Release 0.0.1

Nightscout contributors

August 03, 2014

Contents:

CHAPTER

ONE

COVER LETTER

1.1 a. Cover Letter

To whom it may concern, Dr. Stayce Beck, et al.

This presubmission is to discuss open source projects and FDA oversight. Specifically, this is to discuss the Nightscout project, aka "CGM in the Cloud."

The sponsor, collectively known as Nightscout contributors, may be contacted through one of the core developers:

Ben West 4521 17th St. Apt 5 San Francisco, CA 94114

Additionally, core contributors openly discuss administration of the project via an email list administered by Google groups:

- https://groups.google.com/forum/?utm_medium=email&utm_source=footer#!forum/nightscout-core-dev
- nightscout-core-dev@googlegroups.com

CHAPTER

TWO

NIGHTSCOUT

Nightscout is a suite of open source projects. A smartphone provides ubiquitous network connectivity to Dexcom's wireless receiver. After a polling period the last reading from the Dexcom reciever is transmitted to a database on the internet. A website renders near-real-time views of the records stored in that database. Additionally, the website offers an http endpoint that a pebble watch can use to display the last known alarm status, trend, and glucose level as reported by the Dexcom receiver.

2.1 Device Description

2.1.1 Nightscout project

The Nightscout project is actually a suite of several independent projects:

- android-uploader - Android app to poll dexcom, upload to cloud
- cgm-remote-monitor - A node.js web application that displays values stored by the Dexcom.
- cgm-pebble - A pebble watchface that reads and displays values from cgm-remote-monitor.

When assembled, the completed device is called a "Nightscout rig." In addition to the raw source code for these applications, other community maintained resources exist to help people learn how to assemble their own rigs. These include groups, photos, shared documents, videos on a variety of social media, including a centralized community curated website for documentation as well as community maintained forum software.

cgm-remote-monitor

This is a web app which simulates the display of a Dexcom receiver. In addition to showing the last known glucose level, it displays when the reading was taken, and offers a way to pan several hours retrospectively.

Every 5 minutes, a node.js server polls a mongo database, emitting the last readings over the last two days to any listeners subscribed to the server's "sgv" websocket event. The server also serves a combination of html, css, and javascript to simulate a near-real-time display of the Dexcom receiver.

The web display works on most modern web browsers.

android-uploader

android-uploader is an Android application implemented in java. The application starts when a Dexcom receiver is detected using the operating system's usb management system. The application reads data from the serial port made available by Dexcom's usb connection, and uploads the latest record to a specified data backend. The backend may either be a RESTful API or a mongo db, and is configured using a preferences panel inthe application.

The android-uploader source code must be compiled and distributed as an APK before it can run on an Android smartphone. Once installed and configured to upload data to the preferred cloud "backend", a USB OTG cable is used to connect the smartphone to the Dexcom receiver's micro usb port. The Dexcom receiver is a device cleared by the FDA for continuously monitoring glucose levels sampled from interstitial fluid. The receiver is designed to store and display values transmitted by the Dexcom sensor. android-uploader uses the serial connection provided by this usb capability to exchange data with the Dexcom receiver. The behavior of the uploader has been designed to behave as Dexcom would expect any data management system to behave. There is no expected difference in Dexcom's behavior when the uploader smartphone is attached or while our software is auditing the records on the Dexcom receiver.

cgm-pebble

cgm-pebble is a C and javascript watchface developed using PebbleSDK. The javascript code runs on a Smartphone maintaining bluetooth connectivity to the Pebble watch. The javascript code retrieves information from cgm-remote-monitor and sends the last reading to over bluetooth to the pebble watch. The C code runs on the watch, receiving messages over bluetooth from a smartphone, and rendering the date, time, value, and trend as reported by a running instance of cgm-remote-monitor.

2.1.2 Development

Development takes place using github, from the nightscout organization page: https://github.com/nightscout/. Modifications, upgrades, development, and issue tracking happen using the resources connected to assets shared by a community of people. Each and every change to the source code is tracked by git and discussed through a github pull request. Upgrades are provided by providing git merge requests, often using the Github UI, by identifying the last commit hash in use, and a verified change controlled path to apply latest updates from trusted contributors.

2.1.3 Assembly and guides

The git repos merely provide the source code, and a verified way of exchanging source code for these projects. In order to be used, the source code must be configured, compiled, deployed, and installed.

While each repo contains instructions on how to test and work with that repo, the Nightscout development guides, forums, youtube videos, pictures, twitter account, and Facebook group provide "educational" material on how people have combined and configured these disparate parts to assemble something resembling a "medical device." The web guides also reside in a git repo, where improvements are proposed by the community, reviewed, and adopted in similar manner to the source code itself.

The guides explain how to configure and install each component, with warnings of "things that might go wrong" at each phase. When people experience issues following the guides or during use, they use social media to find people that have similar issues or ask for help. There are also recommendations, optimized for cost and predictability, on which service providers are available, as well as how to work with those service providers.

CHAPTER

THREE

PROPOSED USE

3.1 Nightscout

Nightscout is intended to be used as part of a data management system. The system provides for a "glanceable" secondary display of the information originating from the Dexcom CGM. A website allows the display to be presented on any device which can display websites to duplicate the display of the Dexcom.

3.1.1 Single pane of glass

The website url is typically shared with caregivers and interested parties. This allows multiple people to monitor a Dexcom user's glucose levels from concurrently from any internet connection. Multiple redundant displays eliminates transcription error and raises the fidelity of communicating current therapy status.

3.1.2 Glanceability

Displays are duplicated in multiple redundant locations. This alleviates people from needing to physically locate and attend to the receiver. The lowered burden enables people to be more persistently aware, and therefore respond to scenarios with treatment with greater ease.

For example, in scenarios where no therapeutic action is required, but the glucose levels must be considered, the glanceable display eliminates the 30 second interuption to an existing workflow.

Reliance on pre-existing work

The Nightscout project relies on commodity components, as well as the excellent work from the folks at Dexcom. The android software interacting with the Dexcom receiver attempts to faithfully transmit data from the receiver to a configured storage/data management service hosted on the internet. The android software is agnostic of the data management service, and can be configured to work with several different data management service providers. It also attempts to behave in the way that Dexcom expects all data management systems to behave. An open analysis of the source code listings and comparisons of behavior reveals that the behavior of the Dexcom receiver is unaffected when this system is in use. The use or non-use of Nightscout has no observable difference in the Dexcom equipment or system, either while Dexcom is in use or after. Eg, we believe that Nightscout has no effect on Dexcom's performance, quality, or safety.

When Nightscout is in use, the community recommends that users maintain their normal therapy. Nightscout should not alter therapy plans or decisions. Many of the community members recommend falling back to baby monitors, phone, sms, smbg finger-sticks, and physically checking the Dexcom receiver as tools to augment therapy, even while Nightscout is in use. The guiding philosophy behind this advice is that technology is a tool for managing therapy; that people administer therapy, not technology. Nightscout is another tool using commonly available technology, like

baby-monitors, to bring diabetes therapy, specifically communicating current satus of therapy, more in line with the way the users of these tools, like Dexcom, feel is acceptable.

Uses of Nightscout

Nightscout is useful any time remote near-real-time monitoring of Dexcom readings are desirable. People with diabetes find it useful to keep mindfulness of glucose levels while biking or other activities requiring both hands. People with diabetes find it useful for sharing and gaining empathy of their glycaemic states.

Due to the ease of use, parents have been able to co-ordinate with school Nurse to prevent or treat injuries which are otherwise common. In some cases, use of Nightscout has helped gain insight into how common these injuries are, and we believe that the community aggregator can be used to report these injuries to the FDA for increased oversight of Dexcom and Medtronic devices in the marketplace. The community has also received reports of some parents using Nightscout to co-ordinate sleep-overs or camp visits, and in some cases walks with Grandpa, many for the first time, that would not other wise happen. They all cite Nightscout's remote telemetry in liberating these activities, in some cases with pictures indicating injuries staved off or critical rescue care co-ordinated.

Adult users have cited Nightscout in increasing discretion. A common complaint among users of type 1 diabetes medical equipment is that the mandated use of the equipment combined with the time it takes to use the equipment often presents the unknowing public with a rude experience. It often appears that a PWD is ignoring someone by favoring a phone or pager or just producing rude beeps. When Nightscout is in use, the requirement to touch one of these medical devices disappears, which allows incorporating mindfulness more often and in a variety of different ways into the every day work flow. As a result, fewer interuptions from physically touching the medical device increases discretion because social disruptions are also reduced.

Requirements

3.1.3 Nightscout uploader device

Android smartphone capable of "USB OTG" capability. These are commonly available. WIFI only versions, known as "android mini-pcs" or and "Android TV box" are also commonly available. The prices vary widely from vendor to vendor, and depending on the cell network carrier subsidies.

Without any help, the DIY version requires downloading the source code from the internet. Google's Android software development kit is required to configure and compile the source listings from the git repo. This process requires that users know, or learn how to, prepare their device for debugging, go through basic debugging steps in order to configure, compile, and deploy the software as binary android package, and then install and run the software on their own smartphone.

CHAPTER

FOUR

PREVIOUS WORK

A number of informal discussions have taken place between the FDA and the Nightscout contributors. The discussions have revealed a need to establish a framework for open source authors and FDA to work together in order to best protect and promote public safety. We seek the FDA's guidance in finding similar frameworks from other regulated or life-critical areas such as defense, aviation, and automotive industries.

CHAPTER

FIVE

OVERVIEW OF PRODUCT DEVELOPMENT

5.1 Community based, social technical development

5.1.1 Nightscout begins

As an open source project, the entire source code came into existence when people affected by type 1 diabetes with access to the best and safest therapy options found themselves unable to obtain therapy without any adverse events. In order to help monitor, communicate, and understand therapy, a few individuals created a data management system using commodity equipment allowing them to easily monitor the CGM without requiring physical access to the CGM receiver. Spurred by the improved family relationships and finding therapy easier to track, communicate, and manage, more and more people have added small improvements or helped others to gain liberties on their own. Many of these individuals cite "keeping their own children safe" as reasons for beginning their involvement with the project.

As of July 1, 2014, a dozen or so like-minded individuals record all proposed changes in their own Github forks or Github branches dedicated to discussing improvements or changes to a code base that is in active use by several dozen individuals and families. After the community reviews and tests these proposals in a public audit called a "pull request," one of the core contributors accepts the changes into the "master" branch. This workflow is sometimes called "gitflow" http://nvie.com/posts/a-successful-git-branching-model/.

After the "master" branch has updated with changes relevant to the community, specially crafted pull requests allow tracking the exact git deltas necessary to bring another repo up to date with the community accepted versions. When community members report bugs, this tracking system allows developers to reproduce and co-ordinate fixes, in some cases specifically tailored to members' needs.

For example, in one instance, a several individuals outside the U.S. needed displays in mmol/l vs mg/dl. A group of interested members teamed up to work on special mmol/l versions. The member actually completed the required changes, sharing the needed deltas with the group. As a result, we were able to re-use these same git tracking methods to compare and issue updates specifically for these users needing mmol/l.

5.1.2 Open source methodology

The development of Nightscout as an open source project follows a predictable development pattern to identify issues, incorporate bug fixes, as well as develop new features. The model, as discussed by Gabriel Coleman in Coding Freedom relies heavily on an open review process to share and distribute improvements.

The Software Freedom Law Center, in Transparent medical devices, also outlines the need for greater transparency in the operation of these medical devices. We encourage the FDA to adopt regulations that are consistent with the protections provided by open source methods, including Linus's Law to make all bugs shallow and accessible to those affected by them. In designing Nightscout, we try to adhere to the design principles outlined in Unix philosophy in order to ensure safe, predictable, and effective operation of the Nightscout rig.

5.1.3 Known issues

There are several proposed improvements and known issues. Notably, the system as-is is not HIPAA compliant. One of the key features in this system that has helped to liberate people, and thus make them safer, is the ease of use that accompanies pubically accessible data. While we will adopt optional controls for authorizing and accessing data, parents of this system value easily sharing data with a school nurse with minimum hassle.

5.2 Future plans

The sponsors would like to discuss appropriate regulatory controls that protect open source authors' free speech as well as provides FDA with an appropriate framework to fulfill their mission.

5.2.1 Oversight

Given the community's frustration with safety in available medical devices to manage type 1 diabetes therapy, we believe there are opportunities for open source authors and FDA to work together. One such opportunity is in post-market surveillance. We have developed an aggregator which re-displays de-personalized many Nightscout remote monitors in a single "spaghetti plot." We propose modifying this aggregator to automatically compile and submit reports to the FDA in order to aide in post market surveillance of devices used in diabetes therapy.

5.2.2 Integration

In the interest of safety, we need a single display to contextually manage type 1 diabetes. We will add data transfer from Medtronic insulin pumps to obtain "treatment" data consisting of the bolus wizard and bolus records. Additionally, the display will automatically show both the treatment data, carbohydrates, insulin, and carb ration, from the insulin pump overlaid with glucose readings from the Dexcom CGM.

In addition, we will also explore integrating with many other health, fitness, and nutrition APIs.

We will follow up with additional pre-subs if required to discuss further development efforts.

5.2.3 Operational metadata

We anticipate adding indicators showing the connectivity status of the uploader device, as well as battery status, and other operational details of the system. These details will help quickly assess validity of the data, and whether or not the system is working and trustworthy.

5.2.4 Access controls

During development, the community has expressed an interest in developing access controls to help protect who can access displays. We anticipate development of "named views" which can be used to control who accesses the remote monitor website, as well as when and how. These views may optionally be protected by username/password type of login system, or through creating unique and opaquely encoded tokens explicitly for sharing. We have found that the flexibility in sharing information publically outweighs the risks in the data being made public. As the community and software matures, we anticipate personalizing the access controls to meet the needs of its users.

5.2.5 Support/Commercialization

One criticism of open source is the lack of commercial support for individuals who lack the ability to safely assemble and operate their own rig. While the open source culture provides a large community able to train and offer support, the project remains accessible only to those with sufficient technical ability to assemble and debug their own equipment. We propose that the community would be safer if the public could buy pre-assembled rigs on the market with support contracts to help ensure high quality operation for individuals lacking the time and effort. However, we are concerned that the current regulations considering this a "high risk" device prevents individuals unprepared from obtaining the help they need.

CHAPTER

SIX

SPECIFIC QUESTIONS

- Is this a secondary or passive or active display?
- What does distributing mean?
- What is the medical device? There are 6 separate open source projects, including this pre-sub and the forums, all with separate maintainers and contributors. An assembled rig without the "cloud services" up and running does nothing.
- If listing source code on the internet is widely considered "free speech" how does this relate to "distributing free speech?"
- Is there an API to upload data, relating to injuries or just for all therapy, to the FDA in order to aide in post-market surveillance?
  - Is there a way to prepare some kind of surveillance report, or observations of things we have found to the FDA outside of formal PMA/510k?
    * Should we build a "report to FDA" button into our UI to aide surveillance?
- Should we develop a risk assessment framework for Nightscout?
- Can we work on some framework to provide FDA with oversight for open source projects like Nightscout.
- How would this project be categorized, pending http://blogs.fda.gov/fdavoice/index.php/2014/06/fda-encourages-medical-device-data-system-innovation/ * Is this a realtime monitoring application?
  - Does classifying this as Class III, "high risk" medical device make sense, empower the public to pursue safety in their own therapy?
  - How can open source methodology be integrated into FDA controls?
    * We want FDA enforcing open-ness on existing vendors, can we

    explore a "time in range" endpoint which indicates absence of harm due to therapy? Can we nullify the proprietary protections given to vendors if they have not provided satisfactory evidence that their device enables safe therapy?
- Do any of the FDA certifications/approvals offer additional protection from liability for, Karent Sandler in "Killed by Code, Transparent Medical Devices" seems to suggest there are.

CHAPTER

SEVEN

FEEDBACK

The Nightscout contributors look forward to meeting and discussing these matters in depth. We will provide individuals to meet with, although as an open source project, no individual fully represents the community. We look forward to face to face converstations, emails, and phone calls per FDA availability. As members of the wearenotwaiting movement, we encourage swift and iterative responses, integrated with the MDDS guidance.

CHAPTER

EIGHT

ABOUT

In open source fashion, this document's raw source, pdf version and an html rendering is available online. The purpose is to create a framework for having a discussion with the FDA.

CHAPTER

NINE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*