
Nightscout User Guide Documentation

Release 1

Nightscout User Guide

Dec 30, 2020

1	Grundlagen	3
1.1	Grundvoraussetzungen	3
1.1.1	Grundsätzliche Funktionsweise/Ausblick	3
1.1.2	finanzielle / technische Voraussetzungen	3
1.2	DIY/Disclaimer	4
1.2.1	Inhalt des Onlineangebotes	5
1.2.2	DIY - Eigenbau und Eigenverantwortung	5
1.3	CGM	5
1.3.1	Dexcom	5
1.3.2	Enlite	6
1.3.2.1	MiniLink Transmitter	6
1.3.2.2	Guardian 2 Link Transmitter	8
1.3.3	FreeStyle Libre	8
1.4	MMCommander	8
1.4.1	Hardware	9
1.4.2	Software	11
1.4.3	CC1111 anschließen	12
1.4.4	CC1111 Programmierung	14
1.4.5	Medtronic Uploader einrichten	15
1.4.6	Fehlerbehebung	20
1.5	xDrip	21
1.5.1	Bausatz	22
1.5.1.1	Weiterführende Infos / Schaltbilder für xBridge	27
1.5.2	xDrip App	31
1.5.2.1	Voraussetzungen:	31
1.5.2.2	Grund-Funktionalität	32
1.5.2.2.1	Download & Installation	32
1.5.2.3	Verbinden mit dem Wixel	32
1.5.2.3.1	Bluetooth-Scan / Hardware data source	32
1.5.2.4	System Status	32
1.5.2.5	Transmitter Battery Info	33
1.5.2.5.1	xDrip Battery Info	33
1.5.2.6	Screenshot des Hauptbildschirms	34
1.5.2.7	Starten des Sensors	35
1.5.2.8	Stoppen des Sensors	35
1.5.2.9	Kalibrieren	35

1.5.2.10	Calibration Alerts	36
1.5.2.11	Alarme	37
1.5.2.11.1	BG-Level-Alerts	37
1.5.2.11.2	BG-Alert-Settings	38
1.5.2.11.3	Profil: High, Ascending, ...	38
1.5.2.11.4	Smart snoozing/alerting: Empfehlung am Anfang diese zu deaktivieren.	38
1.5.2.11.5	Rise-Fall-Alerts	38
1.5.2.12	Werte vorlesen lassen	38
1.5.2.13	Statistik	39
1.5.2.14	Widget	41
1.5.2.15	BG History	42
1.5.2.16	Weiteres	42
1.5.2.16.1	Verbinden mit einer Smartwatch	42
1.5.2.16.2	Android Wear	42
1.5.2.16.3	Pebble	42
1.5.2.17	Upload zu NightScout	42
1.5.2.18	Fehlerbehebung	42
1.6	640G	43
1.6.1	Konfiguration des Android Smartphones mit dem Contour Next Link Messgerät	43
1.6.2	Anlegen eines Kontos auf https://carelink.minimed.eu	56
1.7	Libre	62
1.7.1	Nightscout mit FreeStyle Libre	62
1.7.2	Daten des FreeStyle Libre mit Glimp hochladen	62
1.7.3	Libre als CGM	68
2	Nightscout	69
2.1	Account-Information (Arbeitsblatt)	70
2.2	Einrichten der Mongo-Datenbank	70
2.2.1	Mongo DB Administration	77
2.3	GitHub	79
2.4	Azure	82
2.4.1	Einstieg	82
2.4.2	Einbindung des GitHub Source Codes	84
2.4.3	Kosten	86
2.5	Der Deploy Azure Button	87
2.6	Nightscout Plugins	92
2.6.1	Pflichtwerte	92
2.6.2	Eigenschaften	92
2.6.3	Alarme	93
2.6.4	Basiseinstellungen	94
2.6.5	Vordefinierte Werte für Browser-Einstellungen (optional)	94
2.6.6	Plugins	95
2.6.6.1	Standard Plugins	95
2.6.6.1.1	Delta (BG Delta)	96
2.6.6.1.2	Direction (BG Direction)	96
2.6.6.1.3	Upbat (Uploader Batterie)	96
2.6.6.1.4	Timeago (Time Ago)	96
2.6.6.1.5	Devicestatus (Gerätestatus)	96
2.6.6.1.6	Errorcodes (CGM-Fehlercodes)	96
2.6.6.1.7	Ar2 (AR2 Forecasting)	96
2.6.6.1.8	Simplealarms (Simple BG Alarme)	97
2.6.6.2	profile (Treatment Profile)	97
2.6.6.3	Erweiterte Plugins:	97
2.6.6.3.1	Careportal (Careportal)	97

2.6.6.3.2	Boluscalc (de: Bolus Rechner)	97
2.6.6.3.3	Food (de: Nahrungsmittel)	97
2.6.6.3.4	rawbg (Roh Blutzuckerwerte)	97
2.6.6.3.5	Iob (en: Insulin-on-Board, de: Wirksames Insulin)	97
2.6.6.3.6	Cob (Carbs-on-Board)	98
2.6.6.3.7	Bwp (BolusExpert Vorschau)	98
2.6.6.3.8	Cage (en: Cannula Age, de: Pumpenkatheter Setz-Alter)	98
2.6.6.3.9	Sage (Sensor-Alter)	99
2.6.6.3.10	Iage (Insulin-Alter)	99
2.6.6.3.11	Treatmentnotify (Behandlung Benachrichtigungen)	99
2.6.6.3.12	Basal (Basal-Profil)	99
2.6.6.3.13	Bridge (Share2Nightscout Brücke)	100
2.6.6.3.14	Mmconnect (MiniMed Connect-Brücke)	100
2.6.6.3.15	Pump (Pumpenüberwachung)	100
2.6.6.3.16	Openaps (OpenAPS)	101
2.6.6.3.17	loop (Loop)	101
2.6.7	Extended Settings	102
2.6.7.1	Pushover	102
2.6.8	Behandlungsprofil	104
2.6.9	Setting environment variables	104
2.6.10	Vagrant installieren	104
2.7	Die Nightscout Website	105
2.7.1	Haupt Seite	105
2.7.2	Einstellungen	107
2.7.2.1	Berichte	109
2.7.2.2	Profil - Editor	110
2.7.2.3	Nahrungsmittel - Editor	112
2.7.2.4	Administrator Werkzeuge	113
2.7.2.4.1	Verwalten von Berechtigungen	113
2.7.3	Behandlungsportal	114
2.7.4	Bolus Rechner	116
2.8	Installation ohne Cloud Service Provider	118
2.9	Programmierbarkeit	118
3	Smartwatch	121
3.1	Android Wear	121
3.2	Pebble	121
3.3	Apple Watch	121
3.4	Garmin	121
4	Fehlerbehebung	123
4.1	Nightscout Seite bleibt einfach weiß:	123
4.2	Eingabe von Treatments (Insulin, Kohlenhydrate usw.) wird nicht auf der Hauptseite angezeigt:	123
4.3	Eingegebene Werte erscheinen in der Nightscout Verlaufsgrafik an einem anderen Tag:	123
4.4	Die Website reagiert sehr langsam, wenn sie längere Zeit nicht verwendet wurde:	124
5	Sonstiges	125
6	Glossary	127
	Index	129

Contents:

Contents:

1.1 Grundvoraussetzungen

Für den Anfang benötigt man:

- Neugierde
- Interesse daran, etwas für die eigene Gesundheit (bzw. die der Angehörigen) zu tun
- technisches Interesse, aber nicht wirklich Vorkenntnisse!
- ein bisschen Zeit und Geduld

1.1.1 Grundsätzliche Funktionsweise/Ausblick

Man kann *Nightscout* auf verschiedene Weise nutzen:

- als Kontroll- und Alarmsystem für Angehörige von Diabetikern, insbesondere wenn Kinder oder ältere Menschen vom Diabetes betroffen sind
- als Visualisierung deiner eigenen CGM-Daten direkt am Handgelenk, auf dem Smartphone oder nur am PC
- als Blutzucker Tagebuch mit umfangreichen Auswertungsmöglichkeiten
- als Monitoring System für ein DIYAPS (Do-It-Yourself Artificial Pancreas System - oder “Closed Loop”, “Automated Insulin Delivery” oder “Künstliche Bauchspeicheldrüse”)
- als Schnittstelle, um diese Daten weiterzuverwerten

1.1.2 finanzielle / technische Voraussetzungen

- CGM, durch die Krankenkasse oder privat finanziert

- Dexcom G4, G5 und G6 funktionieren sehr gut,
- aber auch Medtrum A6 oder gar der Libre 1 mit Bluetooth-Kappe ist möglich,
- langsam kommt auch das Enlite dazu.
- Ein Stück Hardware, um die CGM-Daten hochzuladen, z.B. **xDrip** (ca. 50 bis 60€) für den Dexcom G4 oder den MMCommander oder ..
- Smartphone (Android-Telefon)
- mit Betriebssystem-Version 4.3 (besser 4.4 oder höher) und
- Bluetooth-Low-Energy (BLE)-Funktionalität. Diese ist in der Regel ab Bluetooth 4.0 vorhanden.
- Datenvertrag (z.B. 0 € bei Netzclub mit 100mb Freivolumen, danach gedrosseltes Internet.)
- eine Kreditkarte (wird nicht belastet)

Nicht notwendig, aber:

- Du wirst sehr bald eine Smartwatch haben wollen! (Android / Pebble)

1.2 DIY/Disclaimer

Warning: Verwende die Informationen aus Nightscout oder den Programmcode niemals, um medizinische Entscheidungen zu treffen.

Warning: Dies ist ein Projekt, das ausschließlich von Freiwilligen erstellt und unterstützt wird.

Warning: Es gibt in diesem Projekt keine Gewährleistung oder garantierte Unterstützung in irgendeiner Art. Wenn Du dich entscheidest es zu nutzen, hängt die Qualität und Leistungsfähigkeit dieses Projektes von dir selbst ab.

Note: Alle Informationen und Gedanken der hier beschriebenen Programmcodes sind ausschließlich zu Informationszwecken und für pädagogische Zwecke bestimmt. Die Verwendung von Programmcodes der Homepage **www.github.com** ist ohne jede Garantie oder Unterstützung jeglicher Art. Jedes Element dieses Systems kann jederzeit ausfallen, wodurch das System unbrauchbar werden kann!

Note: Es gibt keine Passwort geschützte Privatsphäre oder irgendeine Sicherheit der Tools, die hier zur Verfügung gestellt werden. Alle Daten, die du hochlädst, sind für jeden im Internet zu lesen, wenn du die spezifische URL (Internetadresse) freigibst oder weitergibst.

Note: Dies ist kein Projekt des **Dexcom Unternehmens**. Nightscout hat keine Verbindung mit der Firma **Dexcom** oder zu anderen Unternehmen, Dienstleistungen oder Produkten, die im Rahmen dieser Internetpräsenz erwähnt werden.

Note: Alle hier erwähnten Produkt- und Firmennamen sind Marken der jeweiligen Eigentümer bzw. Hersteller.

1.2.1 Inhalt des Onlineangebotes

Die Übersetzer/Autoren übernehmen keinerlei Gewähr für die Aktualität, Richtigkeit und Vollständigkeit der bereitgestellten Informationen auf dieser Internetpräsenz und denen der aufgeführten Links. Haftungsansprüche gegen die Übersetzer/Autoren, welche sich insbesondere auf Schäden körperlicher, materieller oder ideeller Art beziehen, die durch die Nutzung oder Nichtnutzung der dargebotenen Informationen bzw. durch die Nutzung fehlerhafter und unvollständiger Informationen verursacht wurden, sind grundsätzlich ausgeschlossen, sofern seitens der Übersetzer/Autoren kein nachweislich vorsätzliches oder grob fahrlässiges Verschulden vorliegt. Alle Angebote sind freibleibend und unverbindlich. Die Übersetzer/Autoren behalten es sich ausdrücklich vor, Teile der Seiten oder das gesamte Angebot ohne gesonderte Ankündigung zu verändern, zu ergänzen, zu löschen oder die Veröffentlichung zeitweise oder endgültig einzustellen.

1.2.2 DIY - Eigenbau und Eigenverantwortung

Do-It-Yourself (engl.) selbstgemacht. DIY-Projekt bedeutet in unserem Fall, dass es keine Firma gibt, von der man ein fertiges, funktionierendes System kaufen kann. Die Elemente des Systems sind von Teilnehmern des Projektes entwickelt und zusammengesetzt. Sie werden dir kostenlos zur Verfügung gestellt. DIY bedeutet für dich, dass du deinen Teil dazu beitragen musst, damit das System für dich funktioniert. Wir hoffen, dass DIY dich so sehr begeistern wird wie uns - sodass du wiederum irgendwann einen kleinen Teil dazu beitragen kannst, dass das System für alle besser wird.

1.3 CGM

Kontinuierlich messender Glucosesensor — Insulinpflichtige Diabetiker müssen zyklisch ihren Blutzucker prüfen. Eine Möglichkeit, die Messungen besonders oft durchzuführen, ist ein System zur kontinuierlichen Messung der Glucose.

Diese Kontinuierliche Glucosemessung (engl.: Continuous Glucose Monitoring, CGM) wird in der Regel bei Menschen mit Diabetes mellitus eingesetzt, um die Therapie besser steuern zu können. Im Gegensatz zur herkömmlichen Blutzuckermessung wird nicht der Glucosegehalt in einem dafür gewonnenen Blutstropfen gemessen (Blutzuckerwert, BZ), sondern im Gewebe des Körpers (Gewebeglucosewert, GZ). Veränderungen treten dort in der Regel mit einer kurzen zeitlichen Verzögerung im Vergleich zum Blutzucker auf.

Kommerzielle CGM-Systeme bestehen derzeit aus einem Nadelsensor, einem Transmitter (Sender, zur Übertragung der GZ-Werte) und einem Empfangsgerät, das alle Werte speichert und auf dem der Gewebeglucosewert zusammen mit einem Trend auf dem Display angezeigt werden kann.

Auf den folgenden Seiten wird die Vorgehensweise von drei bisher erprobten verschiedenen Systemen vorgestellt. —

1.3.1 Dexcom

ACHTUNG - Alle Angaben zu diesem Produkt sind ausschließlich vom Hersteller **DEXCOM** oder vom deutschen Vertrieb **DEXCOM Deutschland GmbH** (früher Nintamed) und dienen ausschließlich als Information. Es werden hier keine Bewertungen dieses Systems abgegeben oder anderweitige Informationen als die des Herstellers oder des Vertriebs zur Verfügung gestellt.

Beschreibung	Link
Link zur deutschen Seite	Dexcom G4 PLATINUM
Link zur deutschen Bedienungsanleitung	Dexcom G4 PLATINUM (mg/dl)
Link zum deutschen Quick Start Guide	Dexcom G4 PLATINUM (mg/dl)

Fünf Möglichkeiten mit Dexcom-Sensoren:

1. G4-Empfänger mit Kabelverbindung
2. G4-Empfänger “with Share” (Import aus den Vereinigten Staaten von Amerika) mit Share2NS-Bridge
3. G4-Empfänger “with Share” (Import aus den Vereinigten Staaten von Amerika) mit xDrip-App
4. G5 mit Share2NS-Bridge
5. G4 mit selbstgebaute Empfänger ([siehe Kapitel XDrip](#))

1.3.2 Enlite

Eine generelle Beschreibung des Enlite Sensors bietet der Hersteller Medtronic an: [Link Enlite Glukosesensor](#). Auch gibt Medtronic eine [Infobroschüre](#) zu diesem Thema heraus, die allerdings von 2012 ist und sich daher nur mit dem Enlite Sensor in Verbindung mit der VEO Insulinpumpe beschäftigt.

Aktuell gibt es zwei verschiedene Transmitter von Medtronic, mit denen der Enlite Sensor verbunden werden kann:

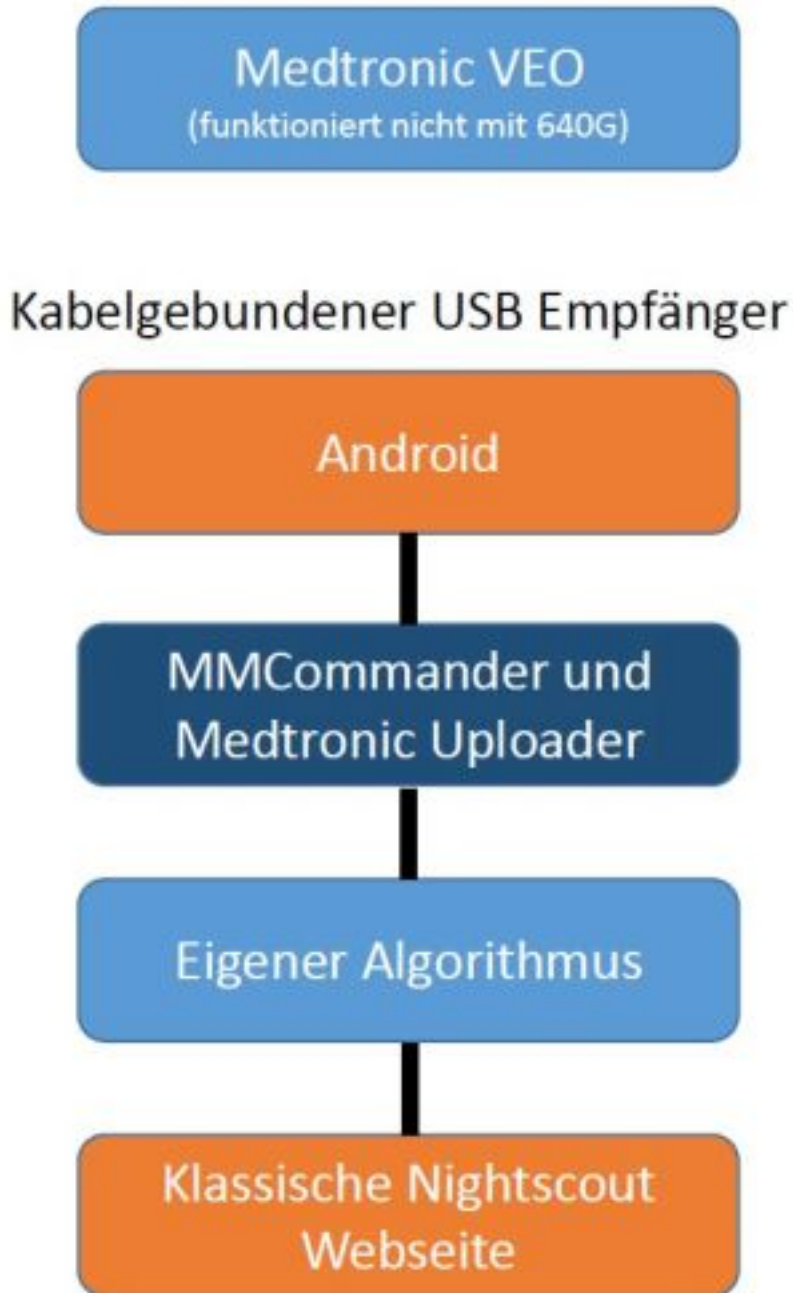
1. der MiniLink Transmitter, kompatibel mit dem **MiniMed Veo** System und dem Guardian REAL-Time System und
2. den Guardian 2 Link Transmitter, der nur mit der **MiniMed 640G** Insulinpumpe kompatibel ist.

Da sich die Wege zum erfolgreichen Nightscout-Betrieb unterscheiden, werden diese im Folgenden getrennt betrachtet.

1.3.2.1 MiniLink Transmitter

Mit dem MiniLink Transmitter und der Medtronic Veo Insulinpumpe gibt es folgende Möglichkeit, die Daten zur Nightscout Webseite zu übertragen.

Übersichtsgrafik



Danach stehen einem die in den Kapiteln [Die Nightscout Hauptseite](#) und [Smartwatch Integration](#) genannten Möglichkeiten zur Verfügung.

Um das zu realisieren, muss man den MMCommander bauen und mit der entsprechenden Software bespielen. Zusätzlich muss noch auf einem kompatiblen Android Smartphone die Medtronic Uploader App installiert und konfiguriert werden. Beides ist im Kapitel [MMCommander](#) beschrieben.

Im Netz findet man auch Hinweise auf eine Lösung für das Iphone (ab dem 4s und IOS 8.2). Diese Lösung, der **RileyLink**, kontaktiert die Pumpe per Funk und stellt dann über Bluetooth eine Verbindung mit dem Iphone her. Dort läuft dann die RileyLink IOS App und transportiert die Daten zu Nightscout. Dies **funktioniert bisher noch nicht mit den deutschen Pumpen**, eventuell da der RileyLink sich gegenüber der Pumpe als MySentry ausgibt, um von dieser die Daten zu bekommen. MySentry wird, lt. Medtronic-Hotline, aber nur von den amerikanischen Pumpen

unterstützt. Auch die mittlerweile offiziell von Nightscout unterstützte Lösung mit Hilfe von Minimed Connect und IOS (s. [hier](#)) betrifft derzeit nur den amerikanischen Markt.

1.3.2.2 Guardian 2 Link Transmitter

Bei der Minimed 640G funktioniert es seit Mitte Mai 2016 recht ähnlich. Hierbei wird das Messgerät Bayer Contour Next Link 2.4 per USB Kabel mit einem [passenden Android Smartphone](#) verbunden. Auf diesem Smartphone wird dann der 640gAndroidUploader installiert und konfiguriert.

Dazu sind die folgenden Einzelschritte erforderlich: Zunächst muss das Kapitel *Nightscout* durchgearbeitet und die Webseite entsprechend eingerichtet werden. Danach geht es weiter mit dem Kapitel *Anlegen eines Kontos auf <https://carelink.minimed.eu>*. Ist dies geschehen, wird noch das Android Smartphone mit dem Contour Next Link Messgerät konfiguriert siehe *Konfiguration des Android Smartphones mit dem Contour Next Link Messgerät*.

Die vorherige, seit Anfang 2016 existierende Lösung sei auch noch erwähnt. Speziell, da damit noch mehr Daten wie zum Beispiel Informationen über Temporäres Basal zu Nightscout übertragen werden können, die derzeit noch nicht mit der App übertragen werden. Allerdings ist diese Lösung deutlich komplizierter und noch nicht Bestandteil dieses Handbuches. Informationen dazu gibt es auf der Nightscout Webseite (s. [Nightscout and Medtronic 640g \(engl.\)](#)) und zusätzlich hat der Autor, Matt, noch einen englischen Blog, der sich mit dem Thema beschäftigt [Matt's Little D Blog](#) an.

Speziell im [Beitrag vom 18.01.16](#) geht Matt auf die erforderlichen Schritte ein. Mittlerweile hat Matt drei verschiedene Lösungen für die 640G erarbeitet. Details in seinem [Beitrag vom 16.4.16](#).

1.3.3 FreeStyle Libre

Der [FreeStyle Libre](#) der Firma Abbott ist seitens des Herstellers kein CGM System, sondern ein FGM (Flash Glucose Messsystem). Der Unterschied ist, dass bei einem CGM System ein Transmitter die Daten automatisch z.B. alle 5 Minuten an ein geeignetes Anzeigegerät sendet und bei einem FGM die Daten nur durch aktives Scannen des Sensors "abgeholt" werden. Derzeit wird der Sensor zumeist durch das [passende Lesegerät](#) ausgelesen. Auch gibt es mittlerweile die [LibreLink Android App](#), die zur Nutzung mit dem FreeStyle Libre zugelassen ist und das Auslesen direkt mit einem Smartphone ermöglicht. Auch bei Nutzung des FreeStyle Libre als FGM ist eine Einbindung in Nightscout möglich. Wie das funktioniert ist hier [HIER](#) beschrieben.

Libre als CGM

Mittlerweile gibt es es ein DIY Projekt, welches die Herstellung eines Transmitters für dieses System ermöglicht und die Firma BlueToolz UG arbeitet derzeit an einer Lösung, die für den Benutzer ohne Bastelkenntnisse eine CGM auf FreeStyle Libre Basis ermöglichen soll. Details zu den Projekten finden sich unter [Libre als CGM](#).

1.4 MMCommander

Der MMCommander stellt die Schnittstelle zwischen dem MiniLink Transmitter (für die Veo) und einem (kompatiblen) Android Smartphone dar. Derzeit wird damit das "alte" System, bestehend aus Enlite Sensor, MiniLink Transmitter und VEO Insulinpumpe, unterstützt.

Benutzer der Medtronic 640G und deren CGM Komponenten sollten sich den Unterpunkt Guardian 2 Link Transmitter des Kapitels [Enlite](#) ansehen.

Das Grundprinzip ist, dass der MMCommander die vom Transmitter an die Pumpe übertragenen Werte "mithört" und diese dann per Medtronic Uploader Android App entweder per Wi-Fi oder mobilem Telefon Netzwerk an Nightscout überträgt (vgl. Übersichtsgrafik im Unterpunkt MiniLink Transmitter des Kapitels [Enlite](#)).

Wichtig:

Der Hersteller Medtronic hat mit dem hier beschriebenen Projekt nichts zu tun und alle Informationen, die sich hier befinden, werden auf eigenes Risiko benutzt. Informationen, die mithilfe dieses Projektes erlangt werden, dürfen niemals als Grundlage für eine medizinische Entscheidung benutzt werden.

1.4.1 Hardware

Um den MMCommander zu bauen, wird folgende Hardware benötigt:

- ein “geeignetes” **Android Smartphone** mit USB OTG Unterstützung.

Ob ein bereits vorhandenes Smartphone geeignet ist, lässt sich leider nicht sicher vorhersagen. Als Orientierung kann man sich auf www.nightscout.info eine Liste mit bestätigten Smartphones ansehen: ([Liste](#)). Zusätzlich gibt es eine Liste mit Smartphones, die laut Nightscout.info **nicht** funktionieren:

- Kyocera Hydro
- Samsung Mini – alle Modelle
- Moto E
- Straight Talk LG L34C Optimus Fuel
- LG Realm
- HTC Desire (außer AT&T Version)
- Core Prime
- Droid Bionic
- Droid Razr Maxx
- Droid Razr
- Geräte mit **Blackberry** Betriebssystem
- Geräte mit **iOS** Betriebssystem (Apple - Geräte)

Zusätzlich zu den Infos der Nightscout-Seite gibt es einen Bericht, bei dem ein HTC One (M7) mit Android 5.1 “Lollipop” nicht stabil funktionierte. Erfolgreich läuft der MMCommander mit einem Sony Xperia Z3 - ebenfalls mit Android 5.1 “Lollipop”. Dieses Smartphone hat den Vorteil, dass man es über einen weiteren magnetischen Anschluss laden kann, während die Micro USB Buchse durch den MMCommander blockiert ist. Im Internet scheinen sich übrigens viele Nightscout-Benutzer für das kostengünstige Motorola Moto G entschieden zu haben.

- ein **USB-OTG Kabel** (Micro-USB Stecker auf normale USB Buchse) - zumeist empfiehlt es sich, einen gewinkelten USB-Stecker zu nehmen, wobei die sinnvolle Richtung vom verwendeten Smartphone abhängt. Man findet eine relativ große Auswahl bei Ebay oder Amazon. Die Kosten betragen etwa 5 €.



- einen **CC1111 USB Dongle**, diesen Funk-Sender-Empfänger kann man z.B. für 67 € bei [Farnell](#) oder [Digi-Key](#) bestellen.



- zumindest einmalig zum Programmieren des CC1111 USB Dongles einen **CC Debugger**. Dieser kann für rund 44 € z.B. ebenfalls bei [Farnell](#) oder [Digi-Key](#) bestellt werden.



- nicht zwingend notwendig, aber durchaus sinnvoll ist ein **Schutzgehäuse** für den CC1111. Es gibt bereits ein Plastikgehäuse, welches man [hier](#) für knapp 20 € erwerben kann. Da der Anbieter mit dem Gehäuse kein Geld verdienen, sondern nur andere Nutzer unterstützen möchte, bietet er auch alternativ die Pläne [hier](#) umsonst an, so dass man diese auch bei einem 3D Druck Service nach Wahl realisieren kann. Eine weitere Möglichkeit ist die Benutzung einer Tic-Tac Dose.



1.4.2 Software

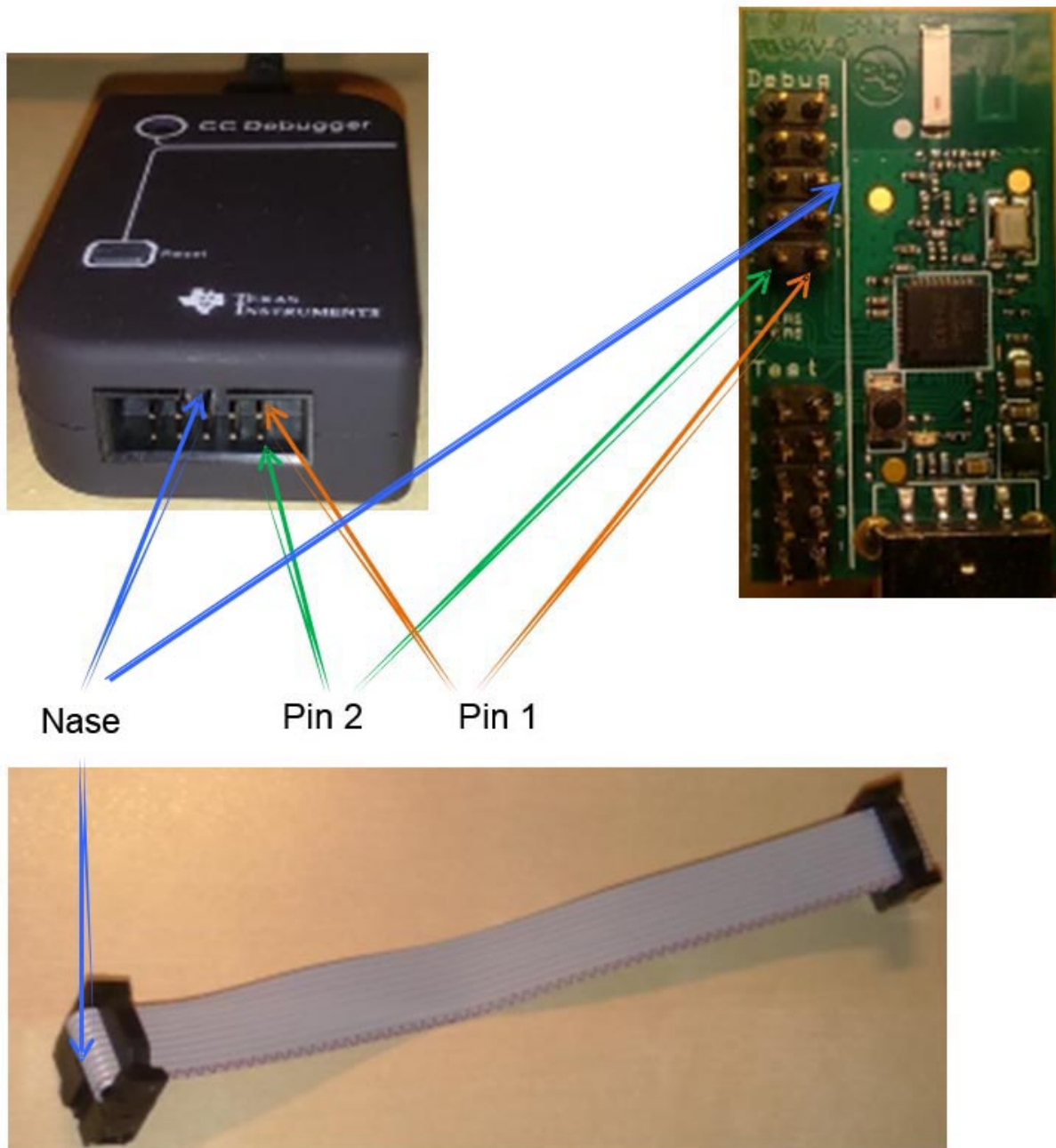
Um den CC1111 USB Dongle mit der entsprechenden Software zu beschreiben, braucht man ein Windows Programm, welches [Smart RF Flash Programmer](#) heißt (Achtung: nicht den SmartRF Flash Programmer 2 runterladen!) und natürlich die aufzuspielende Software. Diese befindet sich [hier](#).

Am besten man installiert bereits jetzt SmartRF Flash Programmer, damit Windows später den CC Debugger erkennt. Es gibt alternativ auch die Möglichkeit, den TI Stick unter Linux mit Hilfe des cc-tool's zu beschreiben, einfacher ist es aber unter Windows. Für Mac ist derzeit keine Lösung bekannt.

Später braucht man noch den Medtronic Uploader, der auf dem Android Smartphone die empfangenen Daten weitertransportiert. Dieser Link steht hier https://github.com/arbox0/MedtronicUploader/releases/download/MEDTRONICUPLOADER_R1_V7/NightScout.apk ausgeschrieben, da es sinnvoll sein kann, das Android Programm direkt vom Smartphone runterzuladen. Sollte dieser nicht mehr funktionieren, kann es auch helfen, [hier](#) zu prüfen, ob es eine aktuellere Version gibt.

1.4.3 CC1111 anschließen

Beim Verbinden des CC1111 USB Dongle mit dem CC Debugger ist darauf zu achten, dass dieselbe Ader des Flachbandkabels am Pin 1 des CC Debuggers und am Pin 1 des Debug Anschlusses des CC1111 angeschlossen wird. Da es verschiedene Lieferversionen gibt (Ader 1 rot oder Ader 10 rot), sollte man versuchen, das Kabel anhand des folgenden Bildes richtig anzuschließen.



Nun müssen der CC1111 und der CC Debugger mit den USB Anschlüssen des Computers verbunden werden.

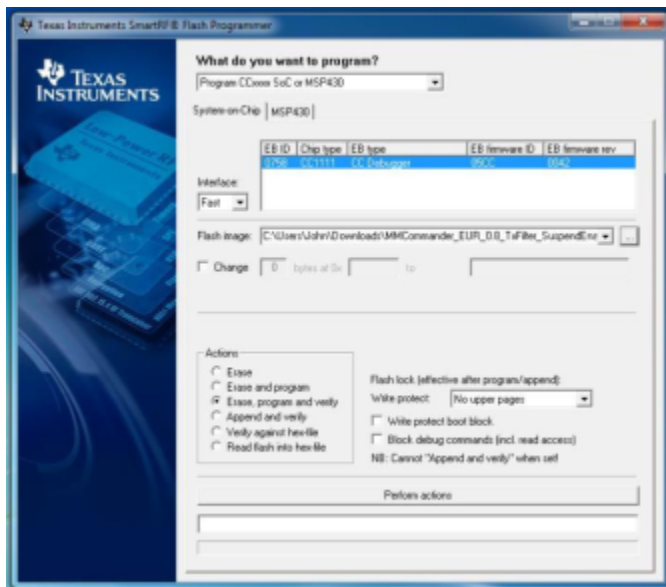


Danach sollte der CC Debugger von Windows erkannt und der USB Treiber installiert werden (vorausgesetzt man hat die Software SmartRF Flash Programmer bereits installiert).

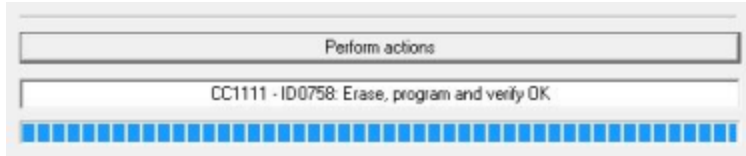
Wenn alles richtig angeschlossen ist, leuchtet die LED am CC Debugger grün, wenn nicht, hilft eventuell ein Blick in das Unterkapitel [Fehlerbehebung](#).

1.4.4 CC1111 Programmierung

Um den CC1111 USB Dongle zu programmieren, muss der SmartRF Flash Programmer entsprechend konfiguriert werden. Im “System-on-Chip” Fenster sollte nun der CC1111 erscheinen. Im “Flash image” Fenster auf den Button mit den drei Punkten drücken und die bereits heruntergeladene HEX-Datei auswählen (MMCommander_EUR_0.89_NoTx.hex). Bei den “Actions” wählt man den Punkt “Erase, program and verify” aus.



Anschließend nur noch auf “Perform actions” klicken und kurz warten. Hat alles geklappt, sollte es so aussehen:



Danach sollte Windows ein neues Gerät namens "MMCommander" finden.

Um zu prüfen, ob der MMCommander erfolgreich programmiert wurde, reicht es, diesen in einen USB Anschluss zu stecken und den Druckknopf auf dem CC1111 einige Sekunden gedrückt zu halten. Die LED des CC1111 geht an. Drückt man den Knopf nochmal für ein paar Sekunden, so geht diese LED wieder aus. Der Stick funktioniert.

1.4.5 Medtronic Uploader einrichten

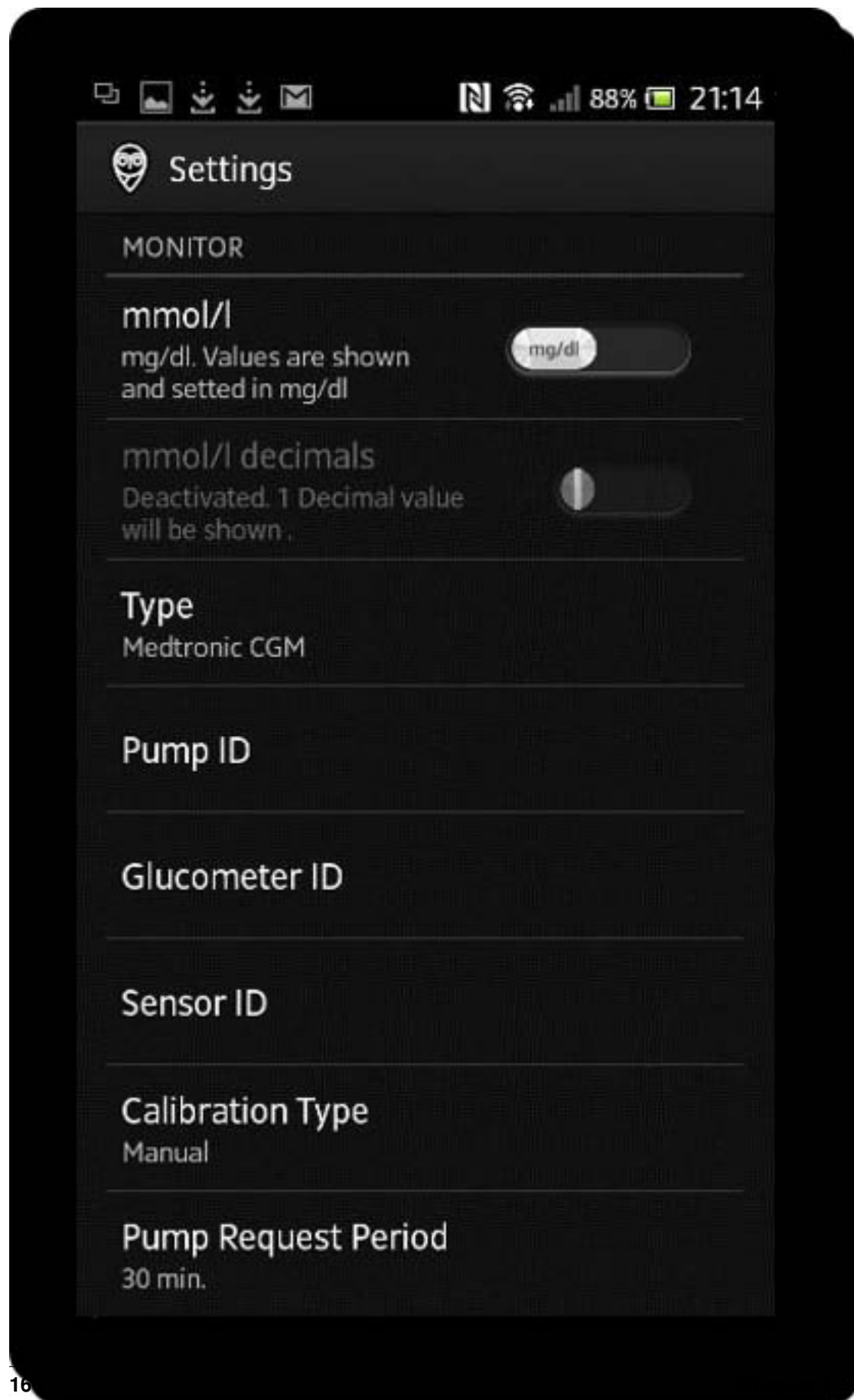
Bevor die Medtronic Uploader App auf dem Android Smartphone konfiguriert werden kann, muss man die Schritte des Kapitels [Nightscout einrichten](#), außer denen des Unterkapitels "Care Portal", vollzogen haben.

Nun installiert man die Datei "NightScout.apk" auf dem Smartphone. Eventuell muss man dazu unter Einstellungen - Sicherheit - Unbekannte Herkunft die Installation von Apps aus unbekannten Quellen zulassen, da diese App nicht aus dem Google Play Store stammt. Dort gibt es nur den für den Dexcom Empfänger gedachten "Nightscout Uploader".

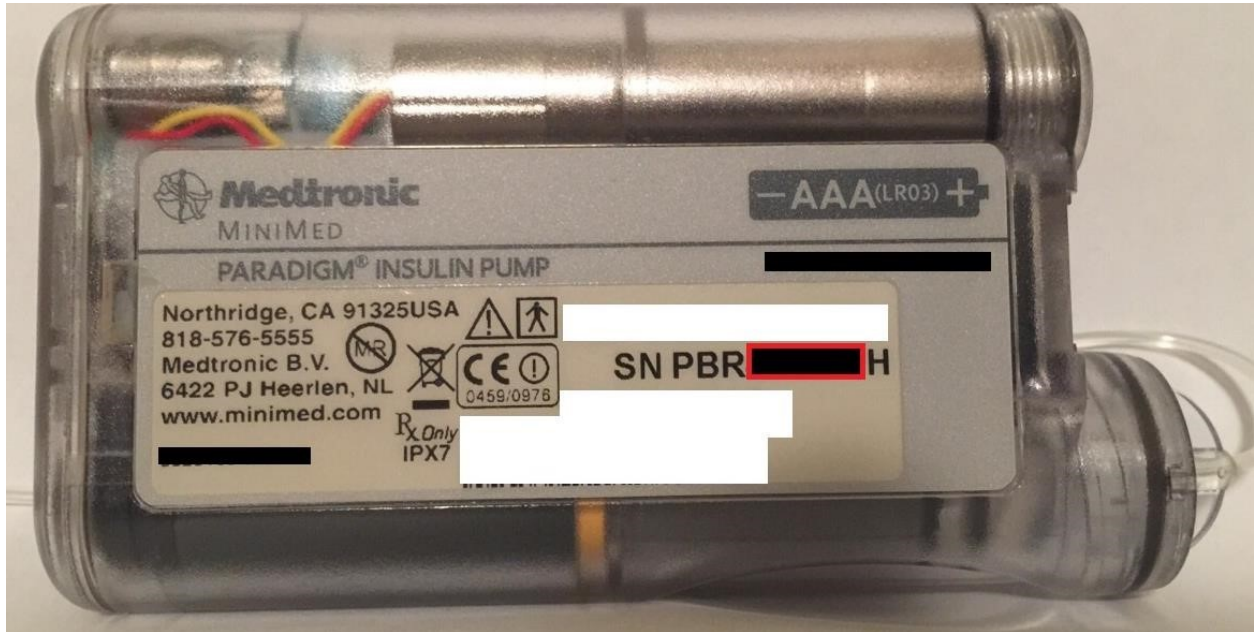
Die Account-Informationen, die man auf dem Arbeitsblatt aus Kapitel dem "Account-Information". eingetragen hat, werden nun zum Konfigurieren der Medtronic Uploader App benötigt.

Zuerst muss die Medtronic Uploader App geschlossen werden, falls diese bereits läuft. Nun den MMCommander über das USB OTG Kabel an das Smartphone anschließen. Es sollte sich automatisch ein Fenster öffnen, welches fragt, was getan werden soll. Dort die Nightscout App auswählen und diese sollte danach automatisch starten.

Wenn jetzt der Uploader läuft, klickt man rechts oben auf die drei Punkte oder, abhängig vom jeweiligen Android Smartphone, man benutzt die Einstellungstaste des Gerätes. Es öffnet sich ein Menü, und dort wählt man den Punkt "Preferences". Hier werden folgende Einstellungen vorgenommen:



- mmol/L -> hier kann man wählen, ob man lieber mit mmol/l oder mg/dl arbeitet.
- Type -> "Medtronic CGM" auswählen.
- Pump ID -> die ID von der Pumpenrückseite eintragen. Dies sind die sechs Ziffern, die auf dem folgenden Bild im Bereich des roten Rechteckes liegen.



- Glucometer ID -> die ID des Blutzuckermessgerätes eintragen (rotes Rechteck).



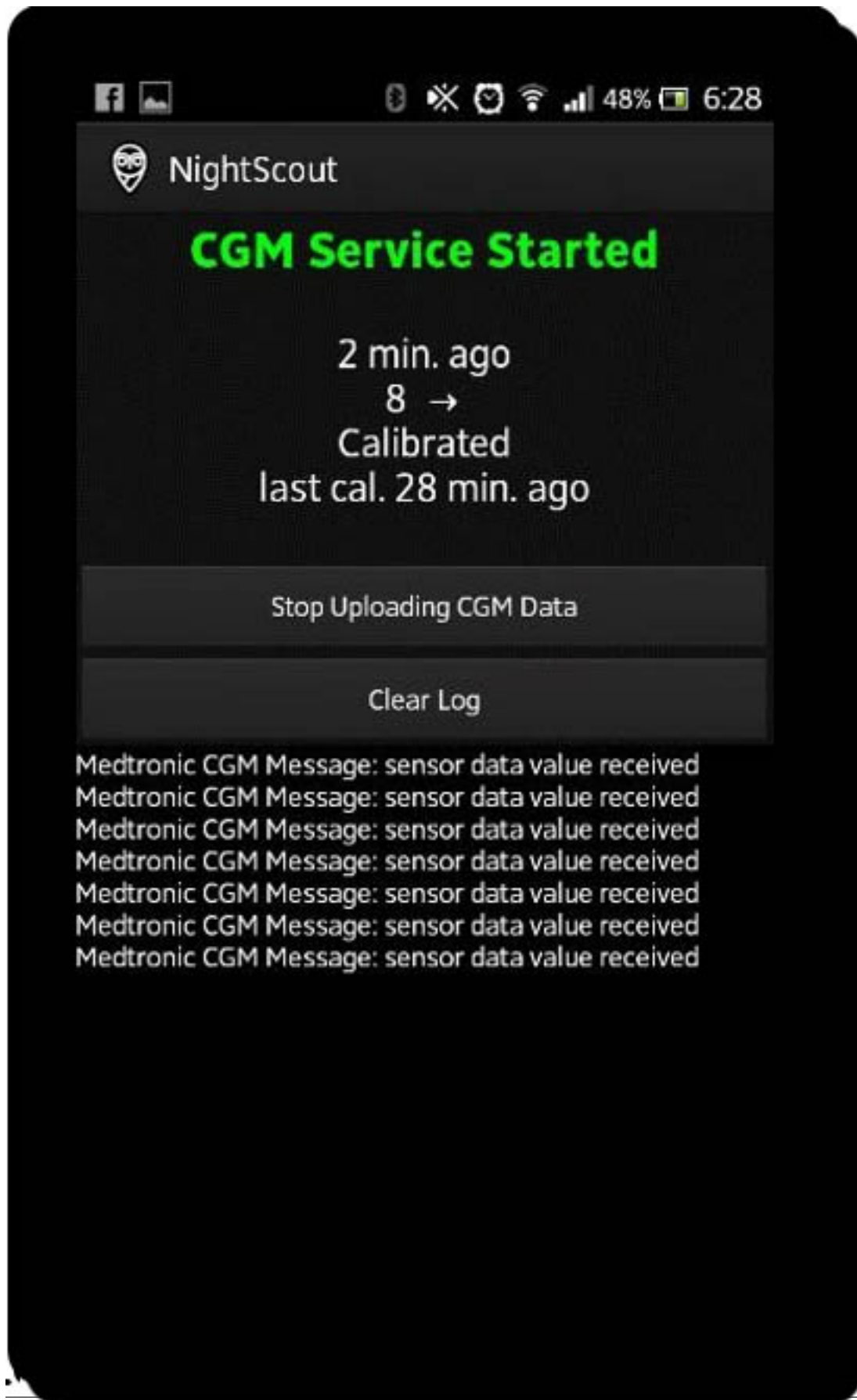
- Sensor ID -> die ID von der Rückseite des Minimed Transmitters eintragen (rotes Rechteck).



- Calibration Type -> “Manual” auswählen.
- Glucose Value Source -> “Medtronic Sensor” auswählen.
- API Upload (REST) -> aktivieren
- API Base URL -> `APISECRET@https://WEBSEITENNAME.azurewebsites.net/api/v1` eintragen. Sowohl das “APISECRET”, als auch den “Webseitennamen” findet man auf der letzten Seite des Arbeitsblattes aus Kapitel 2.1. - vorausgesetzt man hat das Arbeitsblatt ausgefüllt. Mit den Beispielen aus Kapitel 2.5. [Azure](#) ergäbe sich dann: `Y3KmrFA12jmk@https://nscgm01.azurewebsites.net/api/v1`.
- MongoDB Upload -> deaktivieren
- MongoDB REST Upload -> deaktivieren
- Wifi Hack -> deaktivieren
- 2 Days at Startup -> aktivieren
- I UNDERSTAND -> Hier steht ganz klar, dass Nightscout nicht genutzt werden darf, um medizinische Entscheidungen zu fällen. Es gibt weder Unterstützung noch irgendwelche Garantien. Die Qualität und Leistung dieses Projektes hängt einzig von Dir ab. Dieses Projekt wurde von Freiwilligen erstellt und weiterentwickelt. Dies muss man akzeptieren. -> aktivieren
- Logging Level -> Error

Nun geht man raus aus den Einstellungen. Man sollte Striche sehen, einen grünen Text “CGM Service Started” und einen weiteren Text “Uncalibrated”.

Jedes Mal, wenn der Uploader Daten vom Sensor empfängt, erscheint folgender Text im unteren Teil des Bildschirms: “Medtronic CGM Message: sensor data value received”. Sobald man den ersten Wert empfangen hat, klickt man wieder rechts oben auf die drei Punkte. Diesmal wählt man “Instant Calibration” und gibt den Wert ein, den man gerade auf der Pumpe abliest. Anschließend sollte der Uploader alle 5 Minuten einen Wert empfangen und diesen in die Mongo Datenbank hochladen.



Damit wäre der Medtronic Uploader eingerichtet. Es dauert nun noch etwa eine Viertelstunde, bis die ersten Werte über die persönliche Internetseite im Netz abrufbar sind. Ansonsten empfiehlt sich ein Blick in das Unterkapitel [Fehlerbehebung](#).

Man sollte die Kalibrierung mittels “Instant Calibration” jedes Mal durchführen, wenn der MMCommander erneut mit dem Smartphone verbunden wird.

1.4.6 Fehlerbehebung

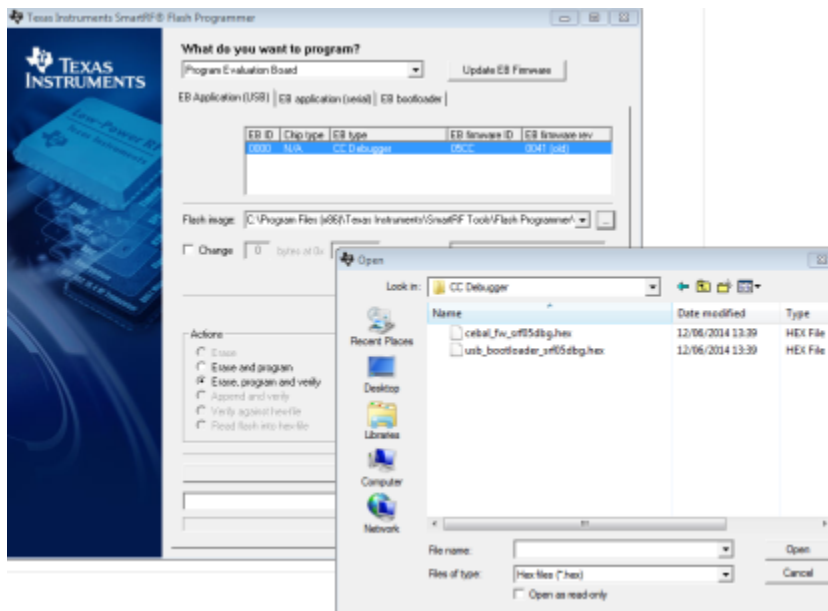
Da die meisten Informationen in diesem Handbuch zum Thema MMCommander und Medtronic Uploader frei aus der entsprechenden englischen Installationsanleitung übersetzt wurden und dabei auch einige, nicht zwingend notwendige, Informationen weggelassen wurden, kann man eventuell [dort](#) in der Datei INSTALL.pdf auch weitere Informationen finden.

Probleme beim Verbinden des CC Debuggers mit dem CC1111

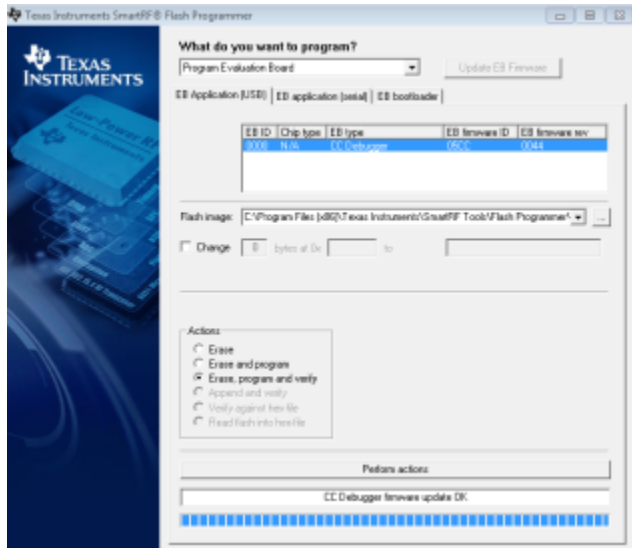
Wenn die Verbindung zwischen den beiden Geräten nicht funktioniert, versucht man am besten, das Kabel auf dem CC1111 andersherum zu stecken. Dabei kann nichts kaputt gehen. Wenn auch diese Maßnahme nichts bringen sollte, dann könnte im Ausnahmefall das Kabel selbst das Problem sein. Als letzten Versuch könnte man die Nase des einen Steckers vorsichtig mit einem Messer entfernen und auch den Stecker im CC Debugger drehen.

Probleme mit der Firmware des Debuggers

Normalerweise sollte es nicht nötig sein, diese Firmware zu aktualisieren, da es auch mit einer älteren Version der Firmware klappen sollte. Wenn es doch nötig sein sollte, weil man zum Beispiel Probleme beim Programmieren des CC1111 hat, dann trennt man zuerst den CC1111 vom CC Debugger. Im Pull-Down Menü des SmartRF Flash Programmers wählt man nun den Punkt “Program Evaluation Board” und klickt danach auf “Update EB Firmware” - dies geht übrigens nur, falls der Debugger eine ältere Version der Software hat. Danach sucht man die Datei mit der neuen Firmware aus z. B. C:\Programme\Texas Instruments\SmartRF Tools\Firmware\CC Debugger\cebal_fw_srf05dbg.hex (Version kann abweichen). Dabei solltet ihr darauf achten, dass ihr nicht den Bootloader aktualisiert.



Anschließend sollte man eine aktualisierte “EB firmware rev” und “... update OK” sehen.



Probleme mit dem CC Debugger

Wenn die CC Debugger LED rot leuchtet, dann sollte man die Verkabelung überprüfen und sicherstellen, dass sowohl der CC111 als auch der CC Debugger an USB Ports angeschlossen sind. Eventuell auch nochmal den Reset Knopf des Debuggers drücken.

Wenn die LED des CC Debuggers rot blinkt, dann hat der Debugger selbst einen Fehler und man sollte im englischen [TI's CC Debugger User Guide](#) nach Rat suchen.

Grundsätzlich

Falls es immer noch Fehler gibt, die hier noch nicht gelöst wurden, dann empfiehlt sich ein Blick in die bereits oben erwähnte englische Originaldatei [INSTALL.pdf](#), die der Autor geschrieben hat.

1.5 xDrip

Eine Möglichkeit, das Dexcom G4-CGM-System zu nutzen, besteht mit der Alternative xDrip und der xDrip App auf einem Android-Smartphone.

Eine kurze Beschreibung zum Prozessablauf:

- Ein Dexcom-Sensor liegt auf/unter der Haut und misst den Glucosegehalt im Gewebe.
- Ein Dexcom G4 Transmitter ist auf dem Sensor eingeklinkt und sendet die Glucose-Werte über ein Funk-Protokoll und würde im regulären Fall vom Dexcom-Receiver empfangen und angezeigt.
- Durch den Einsatz von xDrip kann der Dexcom-Receiver entfallen. Die vom Transmitter gesendeten Werte werden dann von einer selbst gebauten Hardware (xDrip) empfangen und via Bluetooth an ein Android-Handy gesendet. Auf dem Handy wird die xDrip-App installiert.

Das System xDrip besteht aus zwei Komponenten:

1. Die Do-it-yourself-Hardware. Es werden einige HW-Komponenten (siehe [Bausatz](#)) miteinander verlötet.
2. Die [xDrip App](#) wird auf ein Android-Handy aufgespielt und mit einigen Handgriffen individuell konfiguriert. Sie hat folgende Funktionen:
 - Empfang von Gewebezucker-Messwerten
 - Anzeige (und ggf. Ansage) von GZ-Werten

- Weiterleiten der Werte an nachgelagerte Systeme (Pebble/Android-Watch, Nightscout)
- Alarm bei (zuvor individuell definierten) meldewürdigen GZ-Werten / GZ-Verlauf
- Export der GZ-Werte zur Auswertung in andere Systeme (zB. SiDiary, Excel)

Abgrenzung

- xDrip hat keine Freigabe als medizinisches Produkt,
- xDrip hat keine Garantie, keine Gewährleistung.
- xDrip ist kein käufliches Produkt.
- xDrip wird zu 100% in eigener Verantwortung genutzt.
- xDrip kann die Messung des Blutzuckers nicht vollständig ersetzen. (Kontrollen und Kalibrierungen sind nötig.)
- xDrip steuert nicht die Abgabe einer Insulinpumpe.
- xDrip wird ausschließlich von Freiwilligen und ohne jede Bezahlung entwickelt.
- xDrip hat keinen zentralen Ansprechpartner, keine Hotline, keinen Verantwortlichen
- xDrip arbeitet mit dem Transmitter „Dexcom G4“. Es funktioniert nicht mit “Dexcom G5” (Stand 03/2016)

1.5.1 Bausatz

Es gibt verschiedene Varianten, um ein xDrip zu bauen. Das betrifft die verwendete Hardware, die Art die Hardware zu verlöten und auch die Software. (Auf eines der Bauteile - das Wixel - wird eine Software aufgespielt.)

Daher gibt es im Netz unterschiedliche Anleitungen und verschiedene Namen für ähnliche Dinge.

Die Varianten haben unterschiedliche Eigenschaften, letztlich aber dieselbe Aufgabe: CGM-Daten vom Transmitter zu empfangen und an ein Bluetooth-Handy zu übertragen.

Hier soll zunächst der Schwerpunkt darauf liegen, eine leichte Anleitung zum Nachbauen zu haben. Es ist nicht der Anspruch, alle bekannten Bauformen zu beschreiben.

Hier werden drei verschiedene Bauformen vorgestellt. Eine “Bauform” meint eine Art, die Hardware-Komponenten miteinander zu verlöten.

Bauform 1.): “xDrip classic”

Dies ist die einfachste Bauform. Es hat sich der Name der “xDrip classic” etabliert.

Vorteil: Weniger Lötstellen. Einfach und robust. (Was nicht dran ist, kann nicht kaputt gehen.) Im Regelbetrieb leuchtet immer eine LED. Das verbraucht etwas Strom, aber man sieht auf den ersten Blick, ob der xDrip noch “lebt”.

Nachteil: Keine Anzeige der Akku-Ladung. Kürzere Laufzeit, als bei xBridge-Bauform.

Bauform 2.): “xDrip classic mit Spannungsteiler”

Diese Bauform lohnt sich eigentlich nur dann, wenn mit möglichst geringem Aufwand ein “xDrip classic” um eine Anzeige der Akku-Ladung erweitert werden soll.

Vorteil: Mit Anzeige der Akku-Ladung. Geringer Aufwand beim Umbau von “xDrip classic”.

Nachteil: Kürzere Laufzeit, als bei xBridge-Bauform.

Bauform 3.): “xBridge”

Die Stromversorgung des Bluetooth-Moduls läuft etwas anders, als beim “xDrip classic”. Im Vergleich zum “xDrip mit Spannungsteiler” müssen zwei Lötstellen mehr gelötet werden. Insbesondere müssen die Kabel für die Anbindung an das Bluetooth-Modul etwas anders geschaltet werden.

Vorteil: Von den hier vorgestellten Varianten die längste Akku-Laufzeit. Von der Anzeige der Akku-Ladung bitte keine Wunder erwarten - es ist nur eine grobe Orientierung. Der angezeigte Prozentwert sinkt nicht linear.

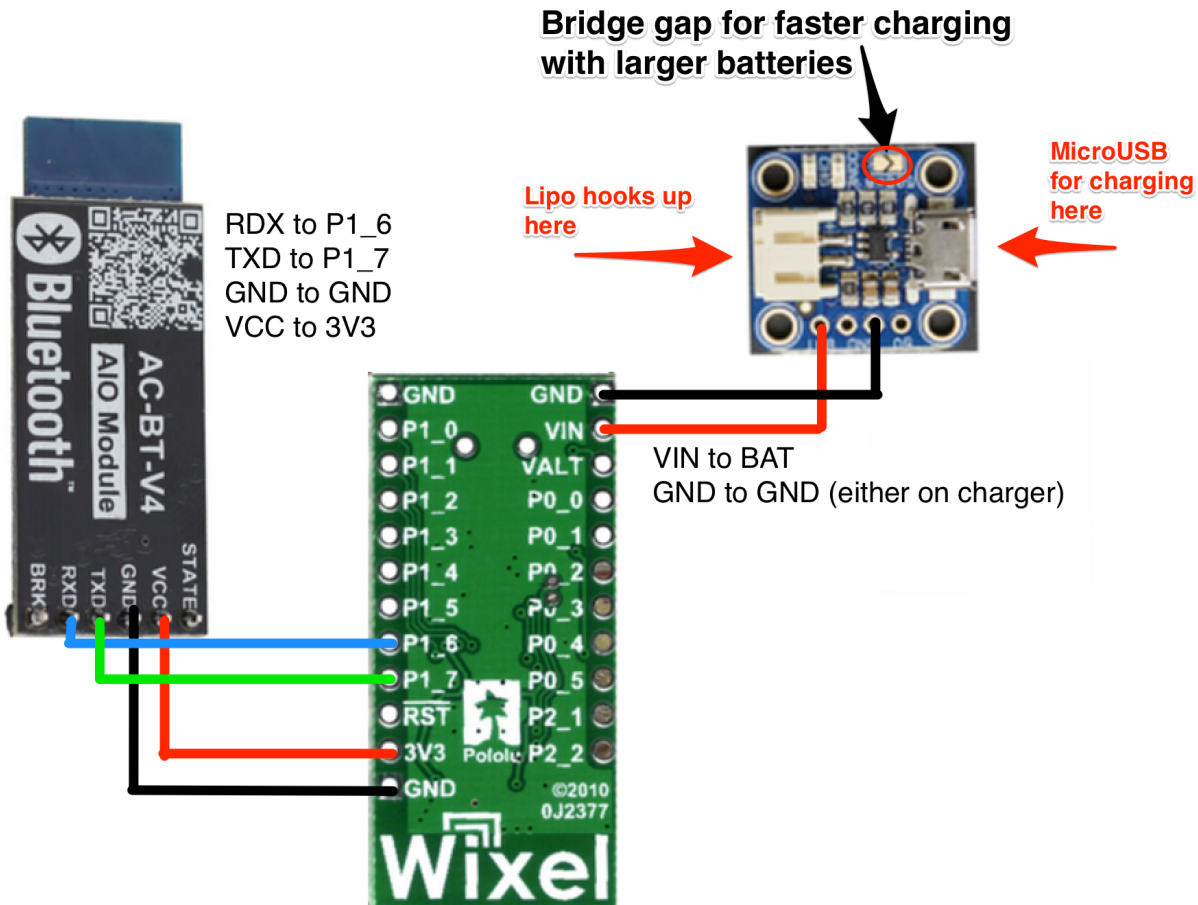
Nachteil: Ungeeignet, wenn sich die Bluetooth-Verbindung zum Handy sich schlecht oder langsam aufbaut.

Tipps und Rat für Einsteiger:

- Baut zunächst einen xDrip in der "klassischen Verschaltung". Es entfallen die beiden Widerstände, dafür fehlt in der App die Batterieanzeige. Wenn das später gewünscht ist, kann man diese Teile problemlos nachrüsten (und die Spannungsversorgung für das Bluetooth-Modul ggf. ändern).
- Verwendet für das Wixel die "xBridge2.wxl"-Software. Für alle hier gezeigten Schaltkreise/Bauformen ist diese Software geeignet.
- Große Akkus bringen (natürlich) eine längere Laufzeit. Es kann aber sein, dass ein großer Akku den Empfang der Antennen verschlechtert. Es kommen alle LiPo-Akkus mit 3,7V in Frage. Eine Empfehlung aus Erfahrungswerten ist, für xDrip classic mindestens 500 mAh verwenden. Große Akkus mit 1100 mAh und mehr bieten auch für xDrip classic mehrere Tage Laufzeit. Für "xBridge"-Schaltkreise können schon Akkus ab 200 mAh mehr als 24 Stunden Laufzeit erreichen. (Tipp aus der Praxis: In aller Regel kann man den Akku über Nacht aufladen - extrem lange Laufzeiten sind also in vielen Fällen unnötig. Das Handy wird ja in aller Regel auch über Nacht aufgeladen... Um etwas Puffer zu haben: Empfehlung etwa 400 bis 500 mAh nehmen.)
- Die Brücke "for faster charging with large batteries" beeinflusst den Ladestrom. Ist die Brücke geöffnet, fließen 100 mA. Wird sie geschlossen, fließen 500 mA. Ein kleinerer Ladestrom schont den Akku - aber die Ladezeit erhöht sich. Der Akku-Hersteller empfiehlt, die Brücke für Akkus ab 500 mAh zu schließen. Zur Information: LiPo Akkus können sich beim Laden erhitzen. Laut Wikipedia kann das eine ernste Brandgefahr mit sich bringen.
- Die Kabel können sehr dünn sein - das erhöht die Flexibilität. Verschiedene Farben machen es leichter, die Übersicht zu behalten, solange der xDrip noch im Bau ist.
- Das Schutzmodul für Lipo-Akkus (unten in der Stückliste mit Nr. 2) hat sich in einem Fall nicht bewährt: Es gab fortwährendes Ein- und Ausschalten. Das könnte am kleinen Akku gelegen haben. Evtl. war es aber Pech oder fehlerhaft verlötet. Im Prinzip ist ein Schutz vor Tiefenentladung sicher sehr sinnvoll. Die Schaltkreise funktionieren aber auch ohne dieses Modul.
- In den Schaltbildern sind keine Schalter eingetragen. Ein Schalter kann sinnvoll sein, um einen zweiten (geladenen, aber abgeschalteten) xDrip als Ersatz dabei haben zu können. Ggf. kann das xDrip abgeschaltet werden, wenn absehbar ist, dass für eine Zeit keine Daten empfangen werden können. (CGM vorübergehend abgelegt, während des Schwimmgehens, ...).

Schaltbild für xDrip in der "klassischen Verschaltung" (Empfehlung für Einsteiger)

(ohne Widerstände für Batterieanzeige)



** Hinweise zum Löten: **

- Wenn ihr ein anderes Bluetooth-Modul habt als in der Abbildung dargestellt, achtet auf die korrekte Verlötung vom Wixel zum Bluetooth-Modul. (Achtet auf die Beschriftung der Pins in der Abbildung).
- Anleitung zum Löten: [Film](#)

** Zur Software auf dem Wixel: **

- xBridge Firmware: [xBridge2.wxl](#) (Auch empfohlen bei Verlötung nach “xDrip classic”).
(Update 01/2017: Mit der xBridge-Verdrahtung wird alternativ diese wxl empfohlen:
<https://github.com/savek-cc/wixel-sdk/blob/master/apps/xBridge2/xBridge2.c>
Achtung: Nur in Zusammenspiel mit der Handy-App “xDrip+” lauffähig. Nicht mit xDrip.)
Die Ansteuerung der Dioden ist anders, als unten beschrieben. Die Laufzeit ist höher, als in den anderen Versionen. Wenn es Lücken in der Verbindung von xDrip-Box und Handy gab, werden diese nachträglich auch für mehrere Stunden aufgefüllt.)
- Software zum Aufspielen der wxl-Datei auf das Wixel [Wixel Configuration Utility](#)
- Weitere Infos: siehe [hier](#)
- Tipp: Wenn auf dem Wixel schon eine xBridge2.wxl Software aufgespielt ist, dann legt die Software das Wixel zyklisch in einen passiven Modus. In dieser Zeit kannst Du das Wixel nicht via USB-Kabel mit dem PC verbinden.
Es gibt zwei Möglichkeiten: Warten, bis das Wixel (alle 5 Minuten) aufwacht, die Verbindung aufbauen und in der “Wixel Configuration Utility” die Anwendung auf dem Wixel stoppen. Oder das Wixel kurz von der

Spannungsversorgung trennen und dann die Verbindung aufbauen.

** Info zu den Widerständen: ** (wichtig für die Batterie-Anzeige)

- “xDrip classic” (ohne Spannungsteiler) Also keine Widerstände verbaut.
- “xDrip classic zzgl. Spannungsteiler”. Dann muss der Spannungsteiler ein Verhältnis von 1:2,2 haben. Empfohlen werden 10k und 22k. (Aber die Kombination mit 1k und 2,2k geht auch.)
- “xBridge” (incl. Spannungsteiler). Dann muss der Spannungsteiler ein Verhältnis von 1:2,7 haben. Empfohlen werden 10k und 27k.
- Wenn Widerstände neu eingebaut wurden, wenn die eingebauten Widerstände verändert wurden oder wenn der Akku getauscht wurde, muss anschließend die xBridge2.wxl neu aufgespielt werden. Sonst stimmen die “angelernten” Werte nicht und die angezeigten Werte des Battery-Levels in der xDrip-App sind falsch.
- Metallschicht-Widerstände sind etwas größer und damit leichter zu handhaben. Es können auch CMD-Widerstände genutzt werden. Damit kann man etwas Platz sparen, aber man braucht dann wirklich eine ruhige Hand beim Löten.

Anzeigen der LED (bzw. in der App) und die Bedeutung (mit xBridge2.wxl auf dem Wixel).

Die Anzeigen sind von der Version der wxl-Datei und von der verbauten Hardware abhängig - in der folgenden Tabelle gibt es eine grobe Orientierung.

Erklärung zur folgenden Tabelle:

- **Die Inhalte sind noch nicht abgesichert.**
- “[STRIKEOUT:grün]” bedeutet: Es gibt verschiedene Bauformen für die Bluetooth-Module (HM10 und HM11), die Farbe der LED kann eine andere sein.
- “-” bedeutet: Diese Anzeige sollte nicht vorkommen.

Bauteile

Bauteil	Anzeige	xDrip classic	xDrip classic mit Spannungsteiler	xBridge
HM17				
HM11	Klärung nötig			
HM10	grün Dauerleuchten	Regelbetrieb	Regelbetrieb	
HM10	grün blinkend	keine BT-Verbindung zum Handy	keine BT-Verbindung zum Handy	keine BT-Verbindung zum Handy
HM10	grün aus, aber alle 5 Minuten kurzes Blinken, dann ca. 30 Sek. an			Regelbetrieb
Wixel	orange Dauerblinken	keine Verbindung zum Transmitter	keine Verbindung zum Transmitter	
Wixel	orange Dauerleuchten			keine Verbindung zum Transmitter
Wixel	orange alle 5 Minuten blinkend	Regelbetrieb	Regelbetrieb	
Wixel	orange alle 5 Minuten ca. 30 Sek. Dauerleuchten			Regelbetrieb
Wixel	ab und zu orange und rot gleichzeitig	?	?	Regelbetrieb
App Statusanzeige	“not connected” dauerhaft	keine Verbindung zum BT-Modul	keine Verbindung zum BT-Modul	keine Verbindung zum BT-Modul
App Statusanzeige	“not connected”, aber alle 5 Minuten für ca. 30 Sek “connected”			Regelbetrieb (!)
App Statusanzeige	“connected” dauerhaft	Regelbetrieb	Regelbetrieb	

Stückliste (Stand: September 2015)

Nr.	Bemerkung	Bauteil:	Vertrieb:	Verkaufspreis:	Artikelnummer:	Hersteller-Nr.:
1	optional	IC Lithium-Polymer Akku-Schutzschaltung	Voelkner	2,96 €	Q33939	S-8261ABJMD-G3J2G
2	optional (und Alternative zu 1)	Schutzmodul für Lithium-Polymer-Akkumulator S-8261ABJMD-G3J2G	Conrad	2,96 €	155151 - VQ	S-8261ABJMD-G3J2G
3	notwendig	Pololu Wixel Programmable USB Wireless Module	EXP TECH	18,60 €	EXP-R25-070	Pololu 1337
4	notwendig	Adafruit Mini Lipo w/Mini-B USB Jack - USB LiIon/LiPoly charger	EXP TECH	7,65 €	EXP-R15-446	Adafruit 1905
5	notwendig, alternativ auch HM10 möglich	Seeed Studio Bluetooth V4.0 HM-11 BLE Module	EXP TECH	12,28 €	EXP-R02-308	Seeed Studio 210005001
6	optional	Handgehäuse Minitec EL, 78x48x20, Lava/Grün	Voelkner	12,38 €	S23261	D9006178
7	optional, Alternative zu 6	Hand-Gehäuse Kunststoff Lava, Grün 78 x 48 x 20 OKW D9006178 1 Set	Conrad	13,57 €	535384 - 62	D9006178
8	notwendig, oder ein anderer 3,7v LiPo-Akku	Lipo Akku 1100 mAh 3,7V Einzelzelle	Lipopower	6,95 €	7.011.0-11_34	1C 603450
9	optional (für Batterieanzeige bei xBridge-Schaltung)	Metallschichtwiderstand 1%, 0,5 Watt je ein Stück 10k und 27k				
10	optional (für Batterieanzeige bei xDrip-Schaltung)	Metallschichtwiderstand 1%, 0,5 Watt je ein Stück 10k und 22k <>				
11	notwendig	flexible Leitungen (dünn) Beispiel	exp-tech.de	0,80 €	EXP-R15-686	Adafruit
12	notwendig	Lötzinn, LötKolben				
13	optional, aber empfehlenswert	Schrumpfschlauch				
14	optional, aber empfehlenswert	“dritte Hand” ein Beispiel: Link				
15	optional	Epoxid Trennschicht				
16	notwendig	Kabel mit Mini-USB-Stecker, um das Wixel zu bespielen				

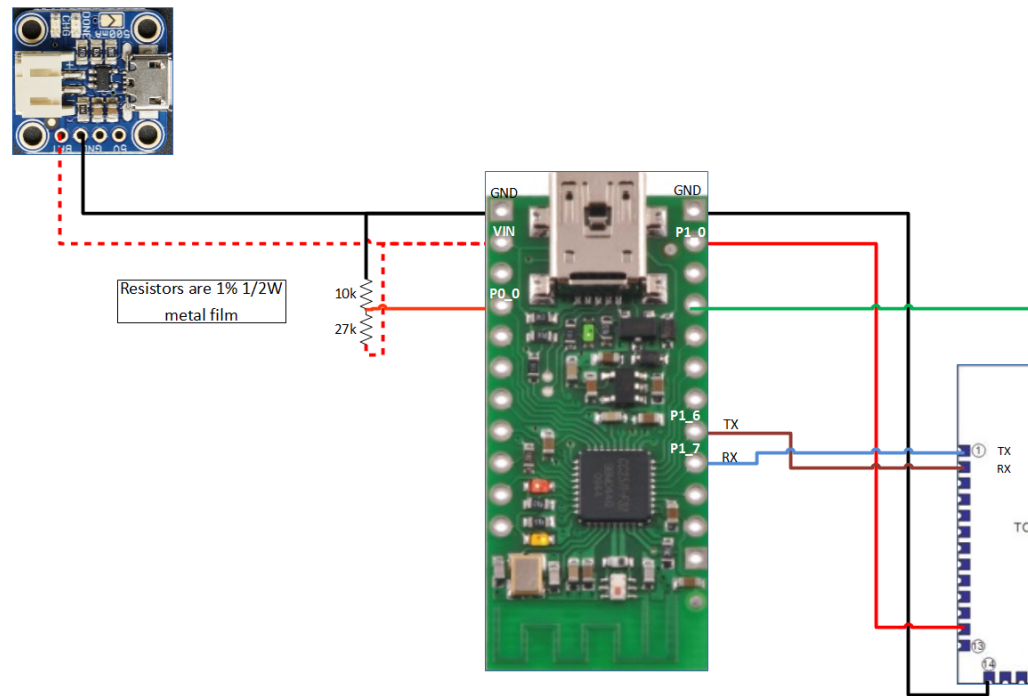
Hinweis: Es gibt Erfahrungsberichte, die besagen, dass ein HM17 sich ebenso verwenden lässt.

1.5.1.1 Weiterführende Infos / Schaltbilder für xBridge

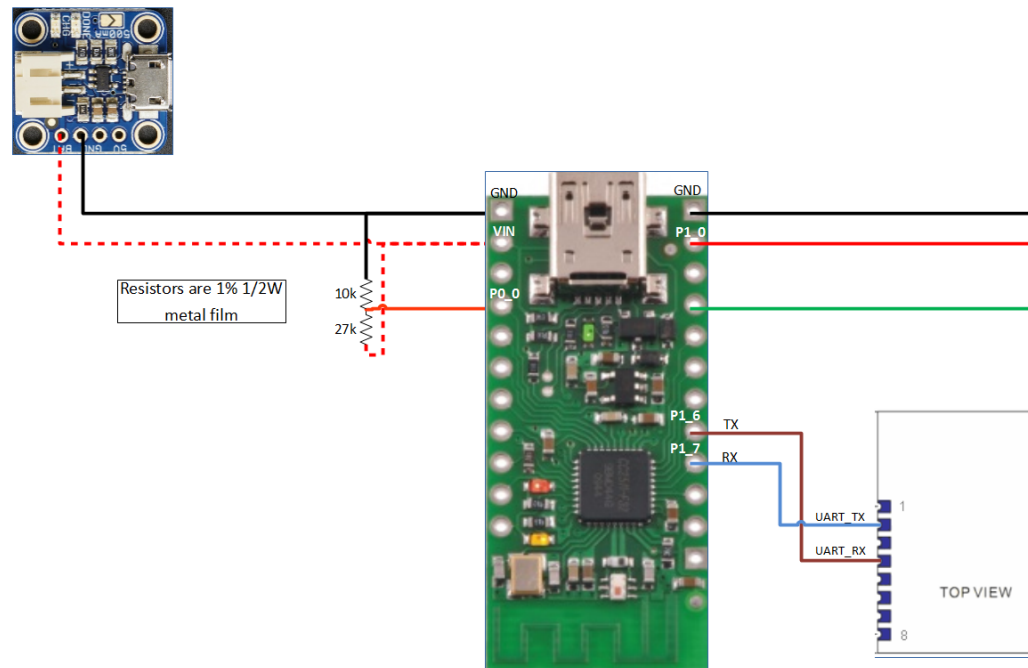
Alternative Schaltbilder / Varianten; wenn die Einsteiger-Variante “xDrip-classic” nicht gewünscht ist.

Es gibt verschiedene Bluetooth-Module: HM10, HM11 oder HM-17 für die es verschiedene Bauformen gibt. Deswegen hier mehrere Bilder zur Verschaltung. Sucht euch das Bild raus, das zu eurem konkreten HM Modul passt. Falls

ihr eine xDrip-Schaltung baut, könnt ihr aus den folgenden Abbildungen auch die Bezeichnungen der Anschlüsse ablesen.



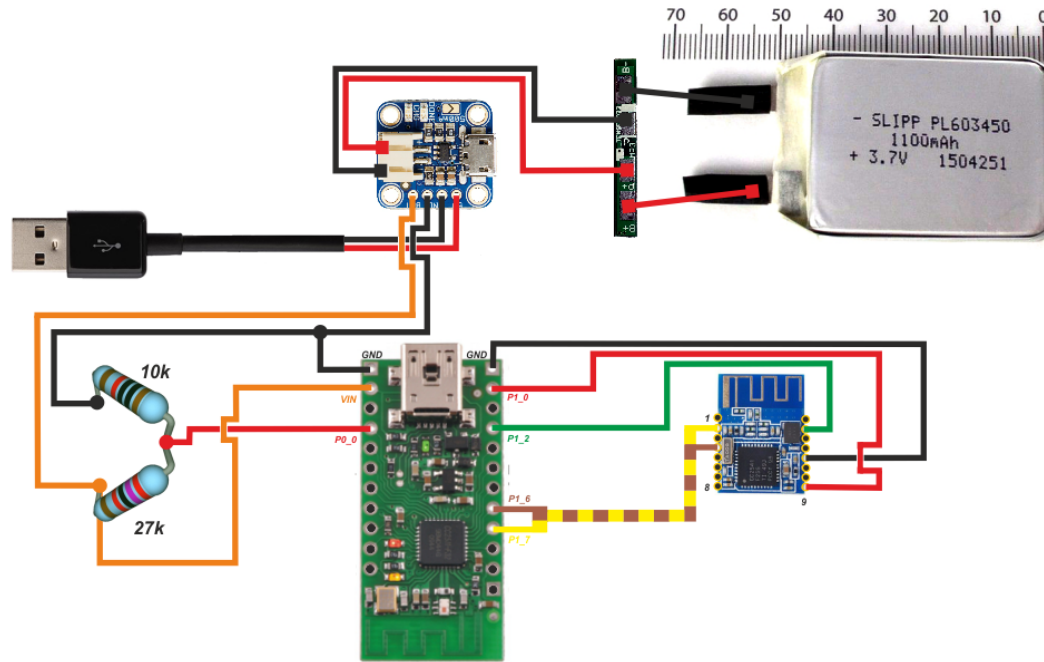
Schaltbild für xBridge mit HM-10



Schaltbild für xBridge mit HM-11



xDrip ^{USB}
v3



Stand: September

Schaltbild für "xDrip USB V3"

** Weitere Infos / empfohlene Links: ** (englische Seiten)

- zu xBridge2: Ein [PDF Dokument](#) findet sich in diesem [Repository](#)
- Eine weitere Seite mit Anleitungen findet sich [hier](#)
- Eine sehr hübsche Anleitung zu Bau der xDrip-Box findet sich hier: <http://www.hypodiabetic.co.uk/home/blog/hacking-diabetes/xdrip-and-xbridge-build-instructions>
- Wichtige Quellen zu xDrip finden sich an diesen Stellen: [hier](#) und [hier](#) und [hier](#)

1.5.2 xDrip App

Die xDrip-App ist eine Anwendung (eine App), die auf einem Android-Handy läuft. Sie dient dazu, CGM-Werte auf dem Mobiltelefon und - optional - einer Smart-Watch anzuzeigen, auszuwerten und bei Bedarf zu NightScout hochzuladen.

Um die Werte zu erhalten, benötigt man entweder ein Dexcom G4 *with Share* oder einen selbstgebauten Bausatz. Da *Share* in Europa nicht erhältlich ist, beschränkt sich dieses Handbuch auf den Bausatz.

1.5.2.1 Voraussetzungen:

1. Android-Telefon mit Betriebssystem-Version 4.3 (besser 4.4 oder höher)
2. Bluetooth-Low-Energy (BLE)-Funktionalität. Diese ist in der Regel ab Bluetooth 4.0 vorhanden und wird häufig mit "Bluetooth Smart" gekennzeichnet.
3. xDrip-Bausatz (s. Kapitel Bausatz) oder Dexcom-Share-Receiver (nur USA)

1.5.2.2 Grund-Funktionalität

1.5.2.2.1 Download & Installation

Die xDrip-App wird derzeit (Stand 01/2016) noch häufig angepasst. Der Download für eine lauffähige Version (*.apk-File) findet sich hier: [Link zur Beta-Version](#)

Die hier verlinkten Versionen sind so erstellt, dass man eine ältere Version mit einer neueren updaten kann, ohne Einstellungen und Messwerte zu verlieren.

Wer hinreichende technische Kenntnisse hat, kann sich aus den vorhandenen Quellen selbst die apk zusammenstellen.

“xDrip” oder “xDrip+”

Alternativ zur “xDrip”-App wird auch die Software “xDrip+” empfohlen. xDrip+ ist eine Variante, bzw. eine Erweiterung der “xDrip”-App. In “xDrip+” sind (erheblich) mehr Möglichkeiten zur Konfiguration enthalten. Das kann wunderbar hilfreich sein, aber die Fülle der Möglichkeiten kann Einsteiger auch verwirren.

Die Beschreibungen dieser Seite wurden für xDrip geschrieben, sie gelten aber ebenso für xDrip+.

Infos und Download zu xDrip+ finden sich hier: <https://jamorham.github.io/>

1.5.2.3 Verbinden mit dem Wixel

1.5.2.3.1 Bluetooth-Scan / Hardware data source

Hier wird eingestellt, mit welcher Hardware die Daten vom Transmitter empfangen und an die xDrip-App übergeben werden.

- Bluetooth Wixel xDrip in classic - xml. Selbst kompilierte wxl, in die die individuelle Transmitter-ID einkompiliert wurde. Erstellt gemäß dieser [Anleitung](#) Link: <https://github.com/StephenBlackWasAlreadyTaken/wixel-xdrip>
- xBridge Wixel xDrip in xBridge2 wxl. Aus dieser [Quelle](#)
- Wifi Wixel Weitere Infos, siehe: ?
- Wifi Wixel + BT Wixel Kombination aus Wifi-Wixel und BT-Wixel. Bsp: Zu Hause mehrere Wixel im WLAN, die im ganzen Haus den Empfang sicherstellen. Außerhalb des WLANs werden die Daten vom BT-Wixel empfangen. Weitere Infos: <https://github.com/jamorham/python-usb-wixel-xdrip>
- DexcomShare DexcomShare ist zZt. in Europa nicht verfügbar. Weitere Infos: <http://www.dexcom.com/dexcom-g4-platinum-share>

1.5.2.4 System Status

Hier findet man Informationen über das System:

- welche Hardware benutzt wird
- welchen Ladezustand die Batterie des Transmitters hat
- wann der Sensor gestartet worden ist
- wie viele Tage und Stunden er bereits liegt
- Connection Status: Ist die Hardware xDrip mit der App verbunden?
- Über Antippen von “Forget Device” kann man die Verbindung unterbrechen und über “Restart Collector” neu in Gang setzen.

1.5.2.5 Transmitter Battery Info

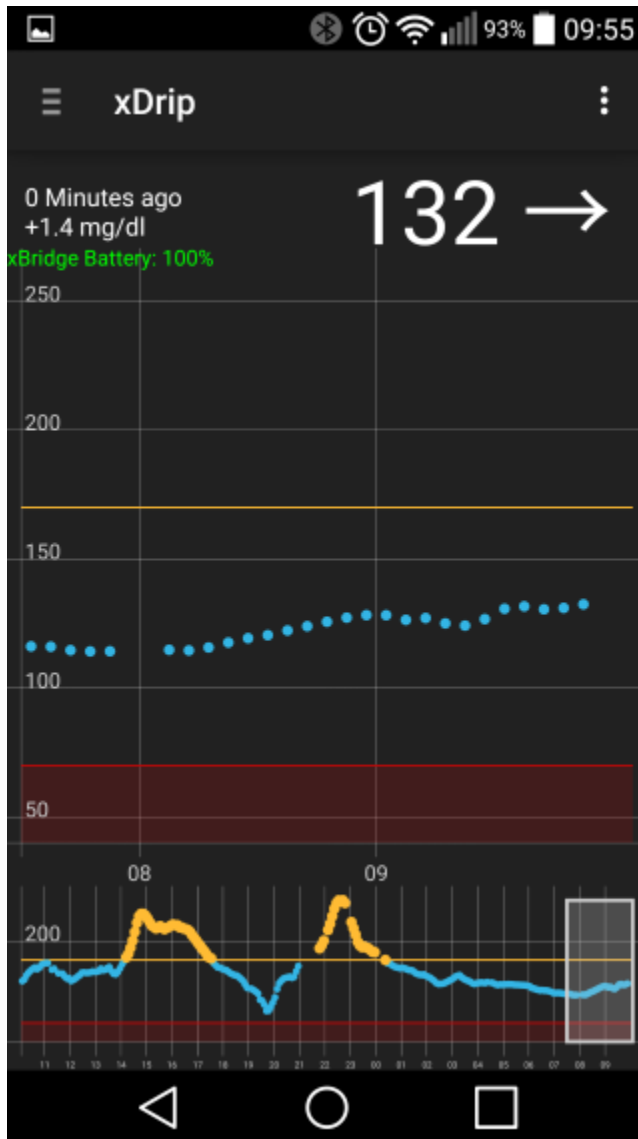
1.5.2.5.1 xDrip Battery Info

—> ist noch zu beschreiben!

Settings

Name	Bedeutung	Empfehlung für Einsteiger
Bg Units	“mg/dl” oder “mmol”	
High Value	Grenze, ab der ein Wert in der Anzeige und Statistik als “zu hoch” gewertet wird.	
Low Value	Grenze, ab der ein Wert in der Anzeige und Statistik als “zu tief” gewertet wird.	
Display Bridge Battery	Sofern ein Spannungsteiler eingebaut wurde, kann eine Batterie-Anzeige gemacht werden	Nur aktivieren, wenn ein Spannungsteiler verbaut wurde.
Run Collection Service in foreground	Klärung nötig: Kann das die BT-Verbindung tatsächlich verbessern? Zeigt in den Statusmeldungen und ggf. im Lockscreen den Wert mit Tendenz an.	aktivieren

1.5.2.6 Screenshot des Hauptbildschirms



oberer Bereich des Hauptbildschirms

- aktueller GZ mit Trendpfeil
- Zeit seit letzter Messung
- Differenz zur vorletzten Messung
- Batterie-Stand von xDrip/xBridge (optional)
- dünne gelbe Gerade: Individuell definierte Grenze für „zu hoch“
- dünne rote Gerade: Individuell definierte Grenze für „zu tief“
- Verlauf der letzten drei Stunden
- Jeder Punkt entspricht einer Messung. Alle 5 Minuten erscheint ein Wert.
- blau: im Zielbereich

- gelb: zu hoch
- rot: zu tief
- rot/weiß: Kalibrierungen

unterer Bereich des Hauptbildschirms

Der Verlauf der letzten 24 Stunden. Der unten rechts weiß markierte Rand zeigt die letzten drei Stunden, die oben größer dargestellt sind.

1.5.2.7 Starten des Sensors

Wenn sichergestellt ist, dass der Sensor liegt und der Transmitter gut sitzt (prüfen, ob an der Seite alles gut eingerastet ist), geht man über das Hauptmenü zum Punkt “Start Sensor”.

Hier folgt neben einem Warnhinweis, frei übersetzt: Erst dann auf “Sensor Start” tippen, wenn der Sensor gelegt und der Transmitter (Sender) auf der Sensorplatte befestigt worden ist, die Bitte, das Datum und die genaue Uhrzeit der Sensorlegezeit auszuwählen, zumindest so nah dran, wie man sich noch erinnert.

Es empfiehlt sich, dieses Datum samt Uhrzeit -inklusive AM oder PM - zu notieren oder einen entsprechenden Screenshot (Foto vom Bildschirm) auf dem Smartphone zu speichern, falls man den Sensor später mit diesem Zeitpunkt nochmal neu starten möchte, z.B. um die bisherigen Kalibrierungen zu löschen oder weil man den Transmitter aus welchen Gründen auch immer kurz abgenommen hatte.

Zur Erklärung der 12-Stunden-Anzeige:

AM = ante meridiem (lateinisch), auf Deutsch: VOR dem Mittag, Eselsbrücke: **Am Morgen**;

PM = post meridiem, auf Deutsch: NACH dem Mittag

Hat man den Sensor gestartet, erscheint auf dem Hauptbildschirm in roten Buchstaben der Hinweis, dass sich der Sensor warmläuft und in 120 Minuten (rückwärtslaufend) eine Doppelkalibrierung benötigt.

Alternativen beim Starten des Sensors

Einige haben die Erfahrung gemacht, dass die CGM-Werte genauer sind, wenn man sich einen neuen Sensor 1 bis 2 Tage vor dem eigentlichen Start dieses Sensors legt, während der alte Sensor noch weiter in Gebrauch ist. Dabei wird bis zum Austausch ein alter Transmitter als Platzhalter auf dem neu gesetzten Sensor verwendet.

1.5.2.8 Stoppen des Sensors

Sobald der Sensor einmal gestartet worden ist, wird der Menüpunkt “Start Sensor” in der App durch den Menüpunkt “**Stop Sensor**” ersetzt.

Wenn man hier antippt, erscheint ein Warnhinweis, man möge den Sensor nur dann stoppen, wenn man auch vorhat, ihn zu entfernen. Darunter befindet sich das Feld mit der Aufschrift “Stop Sensor”. In dieses Feld tippt man, um den Sensor zu stoppen.

Ab diesem Moment wechselt die Anzeige im Menü wieder zu “Sensor Start”.

1.5.2.9 Kalibrieren

Zwei Stunden nach dem Sensor-Start (oder früher, s.o. unter Sensor starten) müssen die vom Sensor gemessenen Werte mit zwei von einem Blutzuckermessgerät direkt hintereinander ermittelten Werten (aus verschiedenen Fingern) abgestimmt, sprich kalibriert werden. Hierfür geht man über das xDrip Menü zum Punkt “**Add double calibration**” und tippt jeweils in die Felder mit der blinkenden Schreibmarke (cursor) und gibt über die auf diese Weise aufgerufene

Tastatur die durch zwei Blutzuckermessungen ermittelten Werte ein. Danach tippt man auf das Feld “done”, auf Deutsch: erledigt. Nach einer kurzen Verzögerung (nicht wundern, das System arbeitet) geht es automatisch zurück zum Hauptbildschirm und die ersten Sensormesswerte sind schon in der Grafik eingetragen.

Für ein paar Minuten erscheint im Auswahlménü die Möglichkeit “**Override Calibration**”, auf Deutsch: Kalibrierung überschreiben. Damit können falsch eingegebene Werte durch die richtigen Werte ersetzt werden.

Anschließend gibt es im Menü nicht mehr die Doppel-, sondern die einfache Kalibrierung “**Add Calibration**”, die wie die Doppelkalibrierung funktioniert, nur daß lediglich ein einzelner mit dem Blutzuckermessgerät gemessene Wert eingetragen werden muss. Wie oft man (einfach) kalibriert, ist Geschmacksache, es sollte jedenfalls nicht zu häufig sein. Man kann sich daran orientieren, dass der Dexcom Receiver alle zwölf Stunden eine Kalibrierung abfragt. Und man kann bei Zweifeln an der Richtigkeit der vom Sensor gemessenen Werte auch mit einem Messgerät gegenmessen, ohne zu kalibrieren. Unstimmigkeiten müssen nicht zwingend am Sensor liegen, sie können auch darin begründet sein, dass der Gewebezuckerwert im Vergleich zum Blutzuckerwert mehrere Minuten verzögert eine Veränderung aufzeigt. Eine Kalibrierung zu einem solchen Zeitpunkt würde sich ungünstig auf die Messgenauigkeit auswirken. Daher der **Merksatz**: *Eine Kalibrierung zu Zeiten durchführen, in denen der Blutzucker möglichst stabil ist, also z.B. vor einer Mahlzeit.*

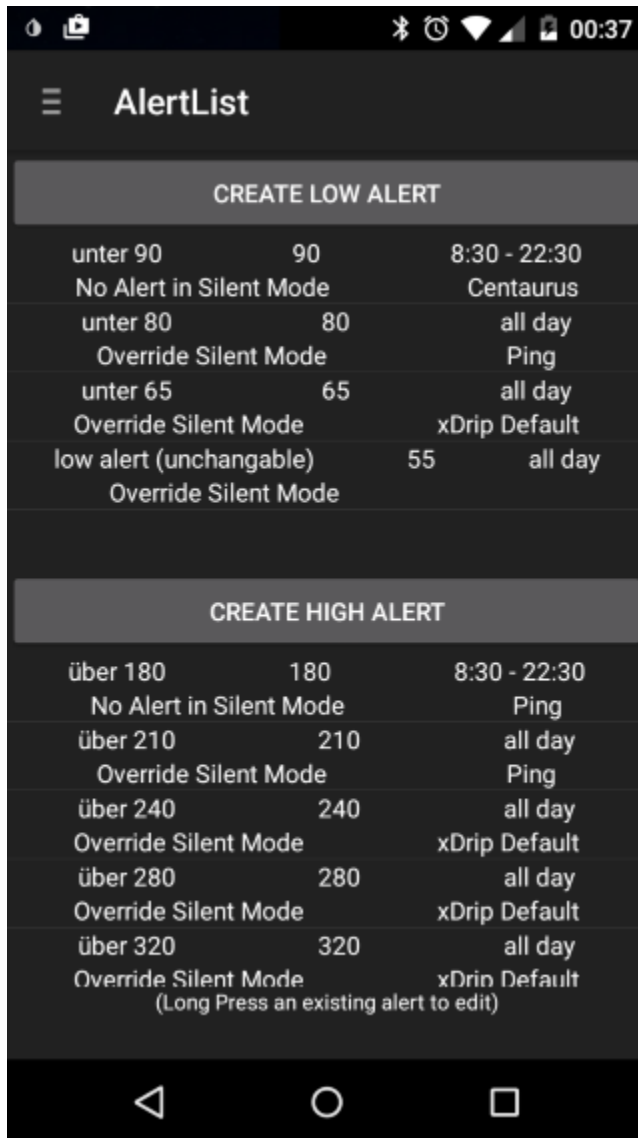
1.5.2.10 Calibration Alerts

Dieser Menüpunkt beschäftigt sich mit dem Kalibrierungserinnerungssignal.

- Calibration Alerts - alert when a calibration is requested: *Hier kannst du durch Antippen ein Häkchen setzen, wenn du an eine erforderliche Kalibrierung erinnert werden möchtest. Das bietet sich z.B. für die Doppelkalibrierung an.*
- Calibration Request Sound: *Hier kannst du dir durch Antippen einen Signalton aus deinem Smartphonerepertoire auswählen.*
- Override Silent Mode on Calibration Alerts: *Wenn du hier ein Häkchen setzt, ertönt das Erinnerungssignal auch im lautlosen Modus des Smartphones.*
- Calibration Request Snooze: *Voreingestellt. Wenn du den Signalton aktiviert hast, wiederholt er sich bei fehlender Kalibrierung alle 20 Minuten.*

1.5.2.11 Alarme

1.5.2.11.1 BG-Level-Alerts



Unter Settings - Alerts and Notifications - BG Level Alerts kann man sich Alarme einstellen für den Fall, dass der BZ einen bestimmten Wert über- (Hochalarm) oder unterschreitet (Tiefalarm). Der Tiefalarm für Werte unter 55mg/dl ist vom System bereits voreingestellt und kann nicht verändert werden.

Drücke “**create low alert**”, um einen Tiefalarm selbst zu definieren. Unter “***Alert Name***” gibst du dem Alarm eine von dir ausgewählte Bezeichnung, z.B. für mg/dl zu niedrig 80. Bei “***Threshold***” tippst du den entsprechenden Wert (hier z.B. 80) ein. Unter “***Default Snooze***” gibst du die Minuten ein, nach denen sich der Alarm erneut melden soll, wenn du ihn gestoppt hast, der Wert aber nach wie vor zu tief ist; unter “***Re-raise***” die Minuten, nach denen sich der Alarm erneut melden soll, wenn du ihn ignoriert hast, der Wert aber nach wie vor zu tief ist. Hier empfiehlt es sich bei Tiefalarmen einen kürzeren Zeitraum auszuwählen. Unter “***Alert Tone***” kannst du dir einen entsprechenden Alarnton bestimmen bzw. mit “choose file” aus deinem Smartphonerepertoire aussuchen.

Unter “***select time for alert***” ist “all day” bereits mit Häkchen versehen, so dass dein Alarm ganztägig funktioniert. Du kannst aber auch auf “all day” tippen, um dann mit “tap to change” einen gewissen Zeitraum (z.B. die Nacht)

festzulegen. Du kannst den Alarm über “***override phone silent mode***” auch im Ruhemodus klingen lassen und das Telefon beim Alarm außerdem vibrieren lassen mit Häkchen an “***vibrate on alert***”. Zuletzt kannst du den Alarm noch ***testen*** und vergiss nicht, ihn zu speichern (“***save alert***”), wenn er dir zusagt!

Entsprechend kannst du über “**create high alert**” eine Alarmkaskade nach oben schaffen.

Bei einmal gespeicherten Alarmen kommst du durch längeres Antippen des gewünschten Alarmes in den Bearbeitungsmodus.

1.5.2.11.2 BG-Alert-Settings

1.5.2.11.3 Profil: High, Ascending, ...

1.5.2.11.4 Smart snoozing/alerting: Empfehlung am Anfang diese zu deaktivieren.

1.5.2.11.5 Rise-Fall-Alerts

—> ist noch zu beschreiben!

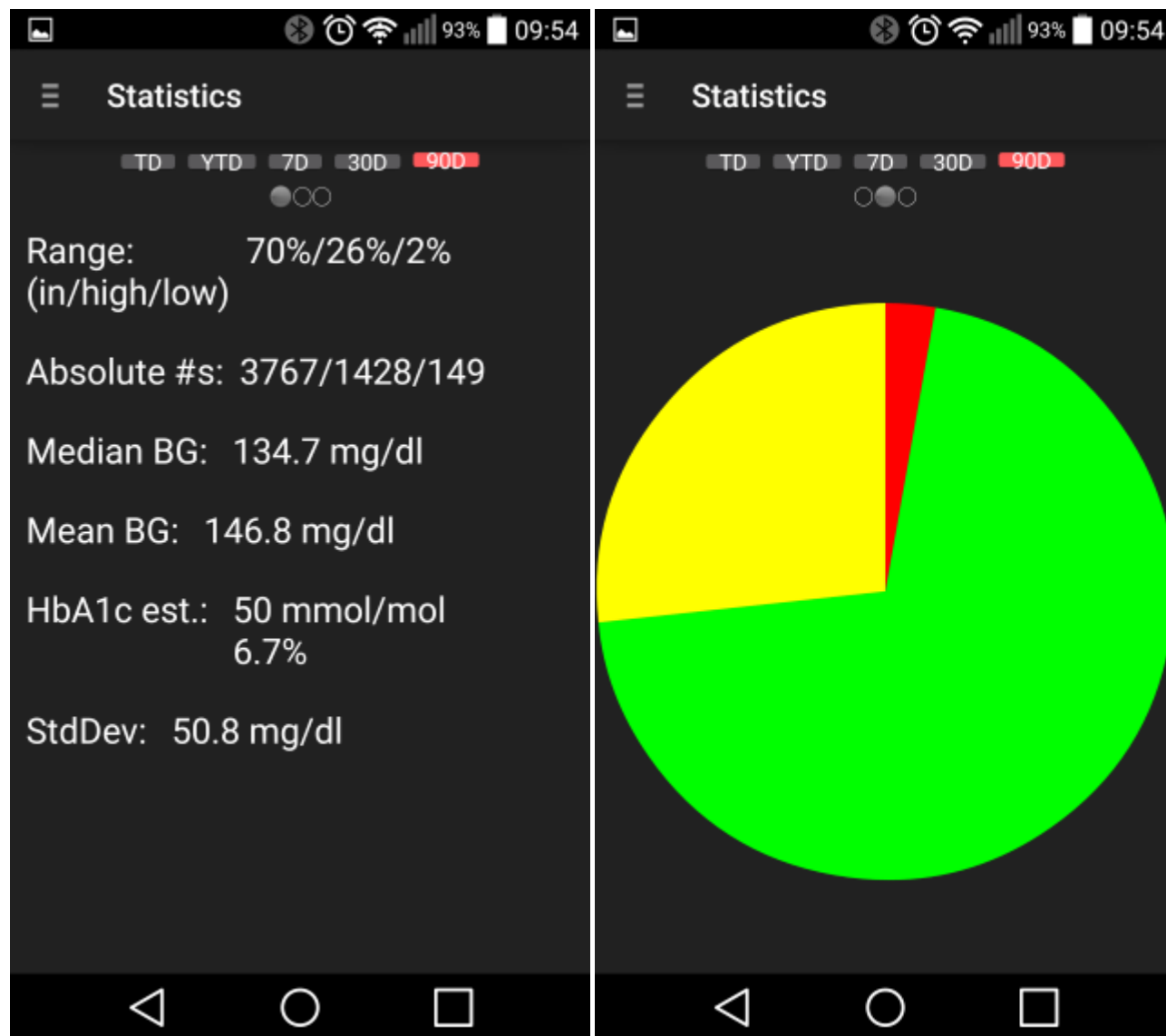
1.5.2.12 Werte vorlesen lassen

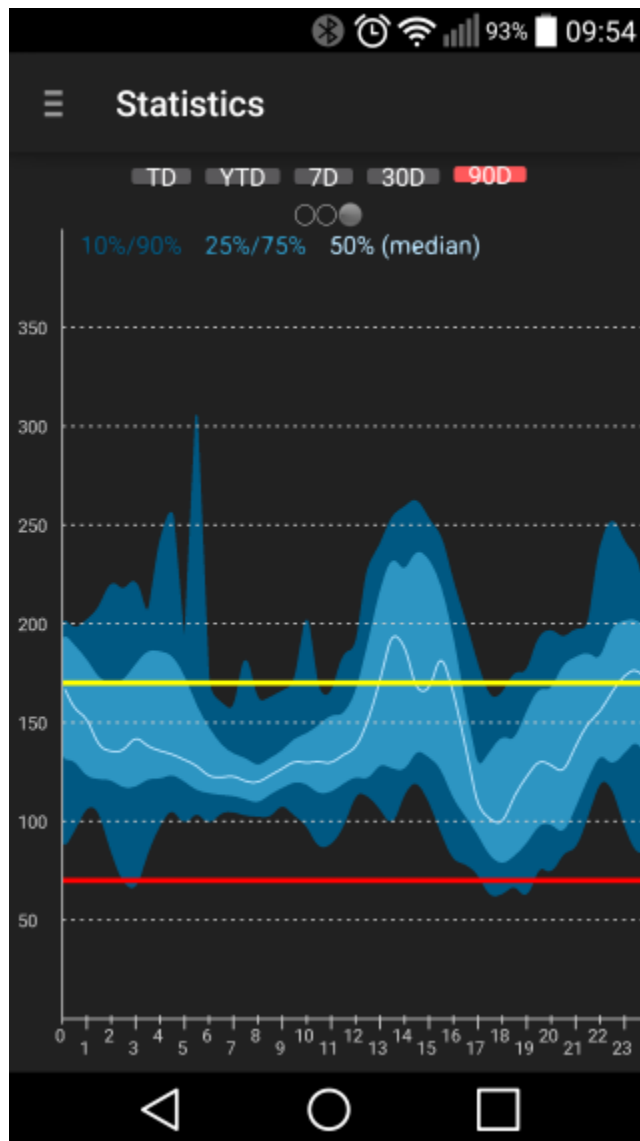
Beim Autofahren oder wenn man aus anderen Gründen die Hände gerade nicht frei hat (z.B. weil man mit den Händen im Spülwasser oder im Kuchenteig steckt) kann man sich die jeweiligen Messwerte laut vorlesen lassen, wenn das Smartphone die sog. “Text-to-speech” Möglichkeit bietet.

Dazu geht man zuerst im xDrip Menü auf Settings (Einstellungen) und verschiebt den Bildschirmtest so lange nach oben, bis man weit unten unter “Other Settings” (Sonstige Einstellungen) den Punkt “**Speak Readings**” findet. Hier setzt man ein Häkchen, woraufhin sich ein Dialogfenster öffnet und fragt, ob man eine passende Text-To-Speech Datei installieren möchte. Bejaht man dies, öffnet sich ein weiteres Fenster, in dem man sich verschiedene Sprachbeispiele (alle mit weiblicher Stimme) anhören (Pfeil antippen) oder gleich die Datei dazu installieren (Download Symbol antippen) kann.

Wenn man nicht jedes Mal diesen etwas längeren Weg gehen möchte, kann man gleich noch ein Häkchen direkt unter Speak Readings bei “**Speak Readings Shortcut**” setzen. Dadurch ist die Vorlesefunktion in das xDrip Menü (je nach Smartphone drei kleine Punkte oben rechts auf dem Hauptbildschirm der App oder bei geöffneter App die Smartphonefläche unten links) aufgenommen und kann gleich dort mit Häkchen aktiviert oder deaktiviert werden.

1.5.2.13 Statistik





Wenn man im xDrip Menü den Punkt Statistics antippt, gelangt man zu den Statistiken für verschiedene Zeiträume, die man wiederum durch Antippen auswählen kann.

TD = today, heute; YTD = yesterday, gestern; 7/30/90d = Tage

Hierzu gibt es drei Seiten, die man durch Wischen nach links bzw. rechts aufrufen kann. Auf welcher der Seiten man sich befindet, erkennt man an der Veränderung der drei kleinen Kreise direkt unter der Leiste mit den Zeiträumen von ungefüllten zu gefüllten Kreisen.

Die linke der drei Seiten enthält Informationstext.

Range, auf Deutsch: Spannweite, gibt die Verteilung der gemessenen Werte an, unterteilt nach in/high/low, auf Deutsch: innerhalb der Grenzwerte/höher/tiefer.

Absolute numbers: Dies sind die absoluten Zahlen der jeweils in den genannten Bereichen gemessenen Werte.

Der Median BG ist der Wert in der Mitte der gemessenen Werte, damit werden Ausreißer nicht berücksichtigt (Bsp: 1,3,5,6,9=5).

Der Mean BG ist das arithmetische Mittel der gemessenen Werte (Bsp: $1+3+5+5+9=23/5=4,6$).

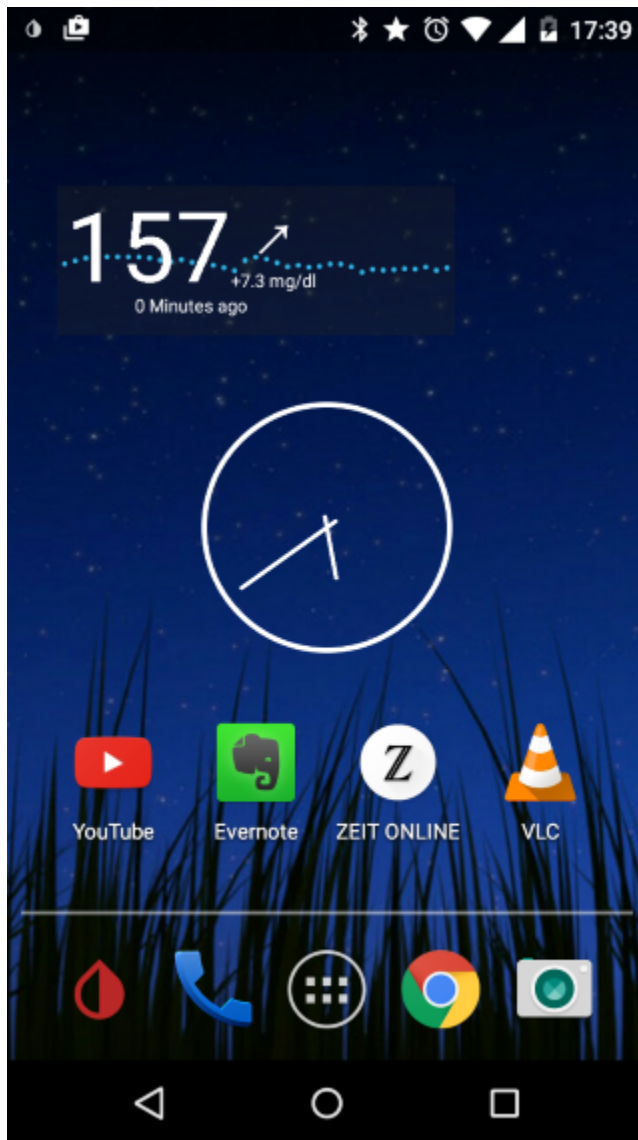
HbA1c est. (estimated, auf Deutsch: geschätzt) Hier wird je nach angetipptem Zeitraum der HbA1c in mmol/mol und in mg/dl überschlägig berechnet.

StdDev steht für standard deviation, auf Deutsch: Standardabweichung. Sie ist ein Maß für die Streuung der Werte.

Auf der mittleren Seite sind die Werte je nach ausgewähltem Zeitraum in ihrer Verteilung als Tortengrafik dargestellt. Grün steht für innerhalb der Grenzwerte, gelb für hohe und rot für tiefe Werte.

Auf der rechten Seite wird für jeden ausgewählten Zeitraum die Werteverteilung in einer Verlaufskurve im Tagesablauf gezeigt.

1.5.2.14 Widget



Die xDrip App kommt nicht allein, sie bringt ein Widget mit, das du dir auf dem üblichen Weg auf deinem Smartphone einrichten kannst. Das Widget enthält den aktuellen Glucosewert mit Trendpfeil, die voraussichtliche Veränderung in der eingestellten Maßeinheit (mmol/l oder mg/dl), darunter die Angabe, von wann der zuletzt gemessene Wert stammt

(z.B. 2 min ago = vor 2 Minuten) und im Hintergrund den letzten Teil der Verlaufskurve. Wird das Widget angetippt, öffnet sich die xDrip App.

Bsp. Samsung S4 mini: links unten durch Antippen der Fläche ins Menü gehen und jeweils antippen - Apps und Widgets - Widgets, dann blättern (wischen) bis zum xDrip Widget, mit dem Finger auf dem Widget dieses nach rechts oder links auf den gewünschten Bildschirm ziehen, loslassen und ggf. durch Ziehen an den äußeren Ecken und/oder Seiten die Größe des Widgets noch verändern, den Bildschirm außerhalb des Widgets antippen, fertig!

1.5.2.15 BG History

(nicht in der jetzigen Beta-Version verfügbar)

Hinter diesem Menüpunkt verbergen sich sämtliche bisherigen Verlaufskurven, die man Tag für Tag über die Doppelpfeiltasten aufrufen kann. Man kann auch vom ausgewählten Datum aus die letzten 1 bis 14 Tage in einer einzigen Anzeige über das Feld direkt rechts vom Datum aufrufen.

Für eine genauere Betrachtung lassen sich die Kurven mit einem Doppelantippen oder mit einer auseinanderlaufenden, nicht zwingend waagerechten, Bewegung zweier Finger in die Breite ziehen. Wenn man beliebige einzelne Punkte auf der Verlaufskurve antippt, erhält man unten kurz eingeblendet die Information über die Uhrzeit und den gemessenen Wert. Diese Einzelbetrachtung ist übrigens ebenfalls in der aktuellen Verlaufskurve des Hauptbildschirmes möglich.

1.5.2.16 Weiteres

1.5.2.16.1 Verbinden mit einer Smartwatch

An dieser Stelle wäre es möglich, die Daten direkt an die Smartwatch zu senden. Damit könnte man die Installationsschritte in "Nightscout einrichten" umgehen. Damit wiederherum umgeht man aber auch die Möglichkeit, die Daten weiter zu nutzen (als Überwachungsfunktion, als Tagebuch oder für das OpenAPS).

1.5.2.16.2 Android Wear

1.5.2.16.3 Pebble

1.5.2.17 Upload zu NightScout

—> ist noch zu beschreiben!

1.5.2.18 Fehlerbehebung

1. **Frage:** In der xDrip-App wird das Bluetooth-Modul gefunden als "hm-soft", nicht als „xBridge“ oder „xDrip“. Was muss ich tun, damit das neue xDrip (richtig) funktioniert? **Antwort:** Sobald der BT-Chip Strom hat, kann sich das Handy damit verbinden. Das heißt leider noch nicht, dass die Daten-Leitung Wixel -> BT-Chip passt. Klicke im System Status auf "forget device", Wixel kurz stromlos machen und danach neu verbinden. Wenn danach der Chip immer noch "hm-soft" heißt, dann stimmt irgendetwas mit der Kommunikation vom Wixel zum BT-Chip nicht.
 - Verkabelung zwischen Wixel und Bluetooth-Modul prüfen
 - Baud-Rate prüfen. Bei einigen HM11-Modulen ist die Baudrate falsch gesetzt, ggf. also die Baudrate neu einstellen
1. **Frage:** Die xDrip App meldet einen Fehler "Unclear Sensor Readings". Was kann man tun, damit die Daten wieder plausibler werden?

Antwort:

Die Meldung wird angezeigt, wenn unter Settings/Other Alerts/Noisy Readings die Checkbox für “Bad (noisy) Value Alerts” aktiviert ist.

Was hilft ist: Warten bzw. den Sensor ersetzen. (Falls jemand eine bessere Antwort hat: Bitte die Antwort löschen und passend umformulieren. Danke!)

2. **Frage:** Nächste Frage kommt hierher.

Antwort:

Nächster Antwort kommt hierher.

—> Erweiterbar!

1.6 640G

Bevor man das Messgerät Countour Next Link 2.4 vorbereitet oder den 640gAndroidUploader installiert sollte man zuerst das Kapitel *Nightscout* durchgearbeitet und die Webseite entsprechend eingerichtet haben.

1.6.1 Konfiguration des Android Smartphones mit dem Contour Next Link Messgerät

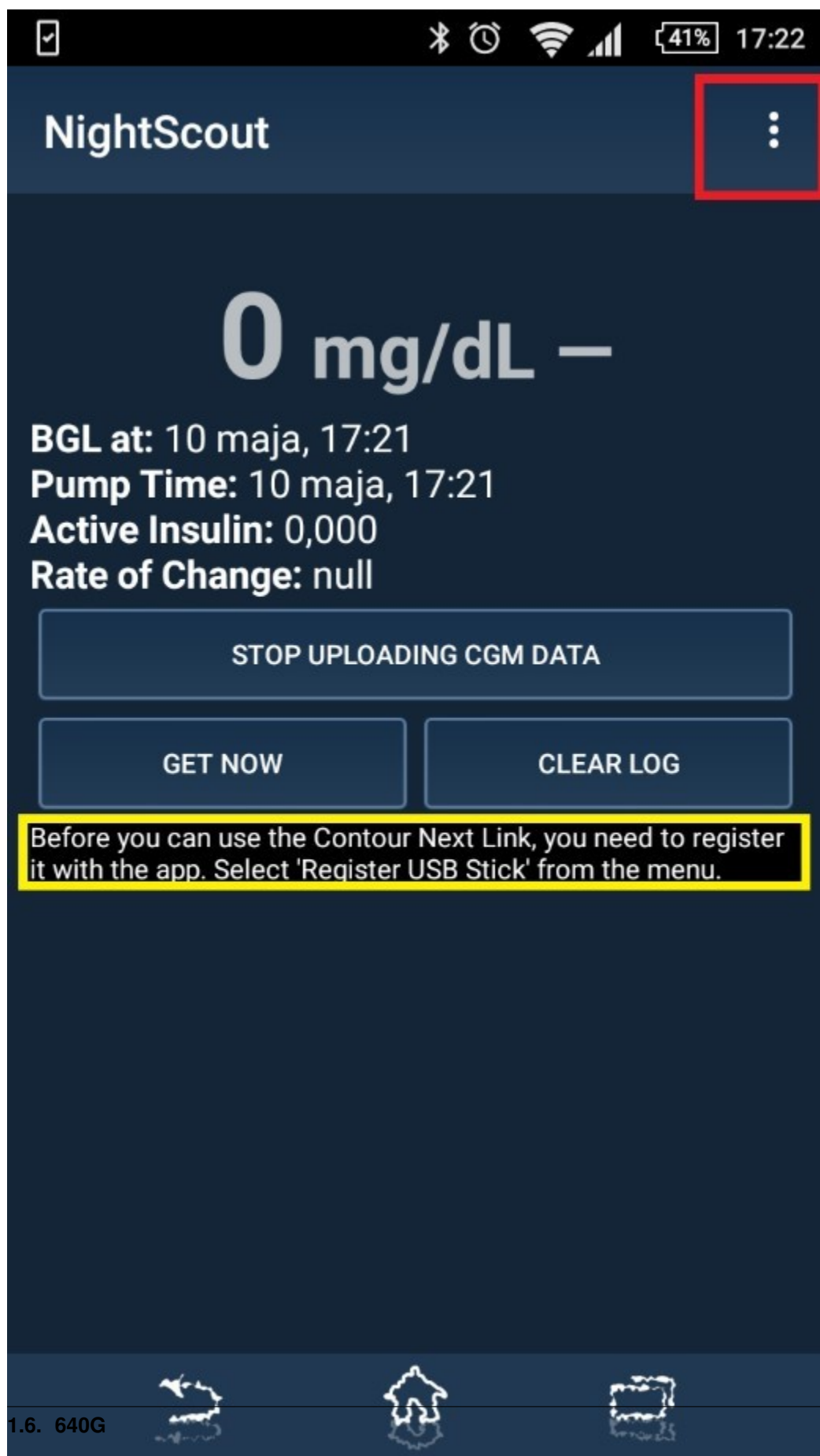
Lade die aktuellste Version der Nightscout Uploader App auf dein Android Smartphone und installiere sie (<https://github.com/pazaan/640gAndroidUploader/releases>).

Nachdem du die App installiert hast, verbinde dein Smartphone mit dem Contour Next Link 2.4 Messgerät über ein USB-OTG (On-The-Go) Kabel und starte die App:





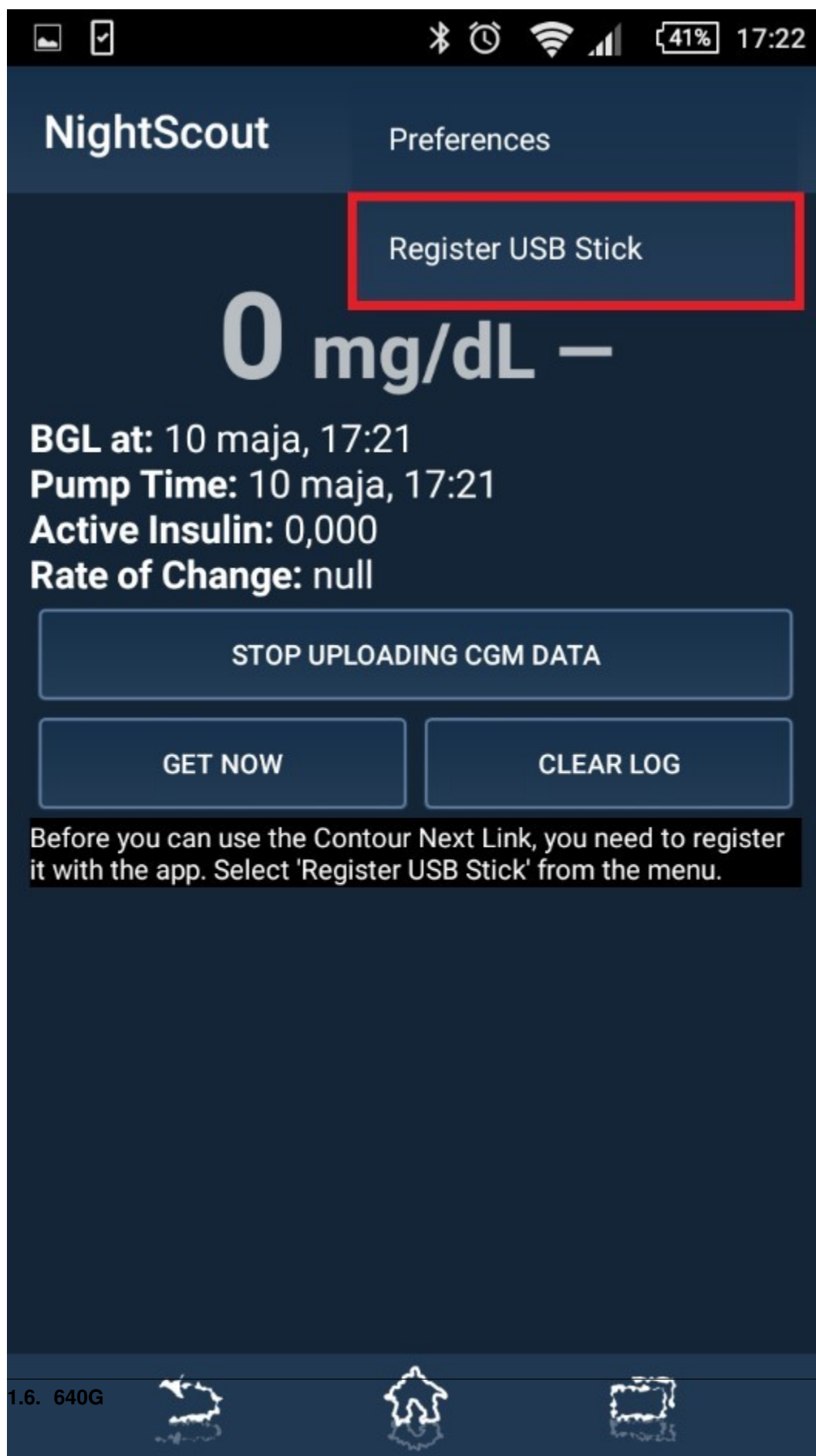
Im ersten Schritt musst du die Nutzungsbedingungen lesen und akzeptieren. Als nächstes musst du dein Contour Next Link Gerät registrieren. Dazu klicke auf die drei kleinen Menü-Punkte in der rechten oberen Ecke:



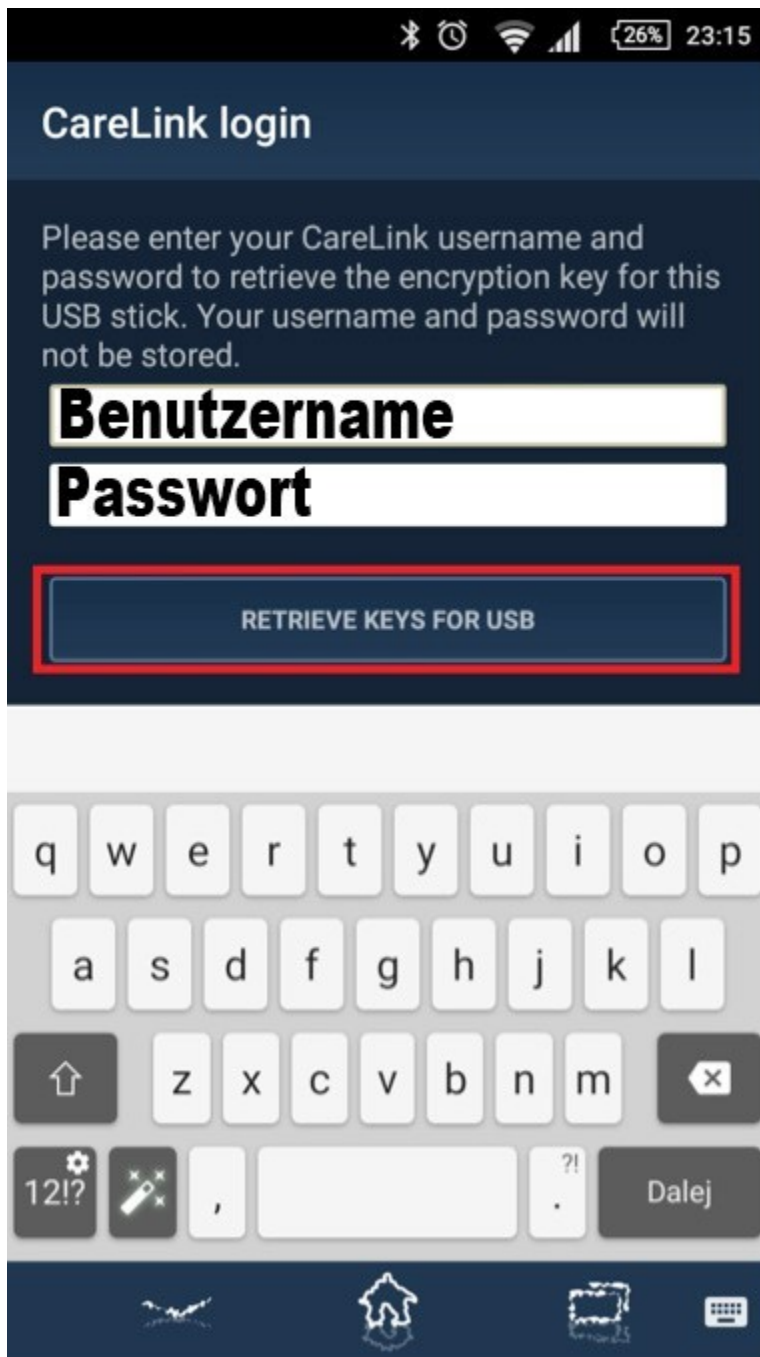
Solltest du keine drei Menü-Punkte sehen, musst du das Menü über deine Menü-Taste des Smartphones öffnen:



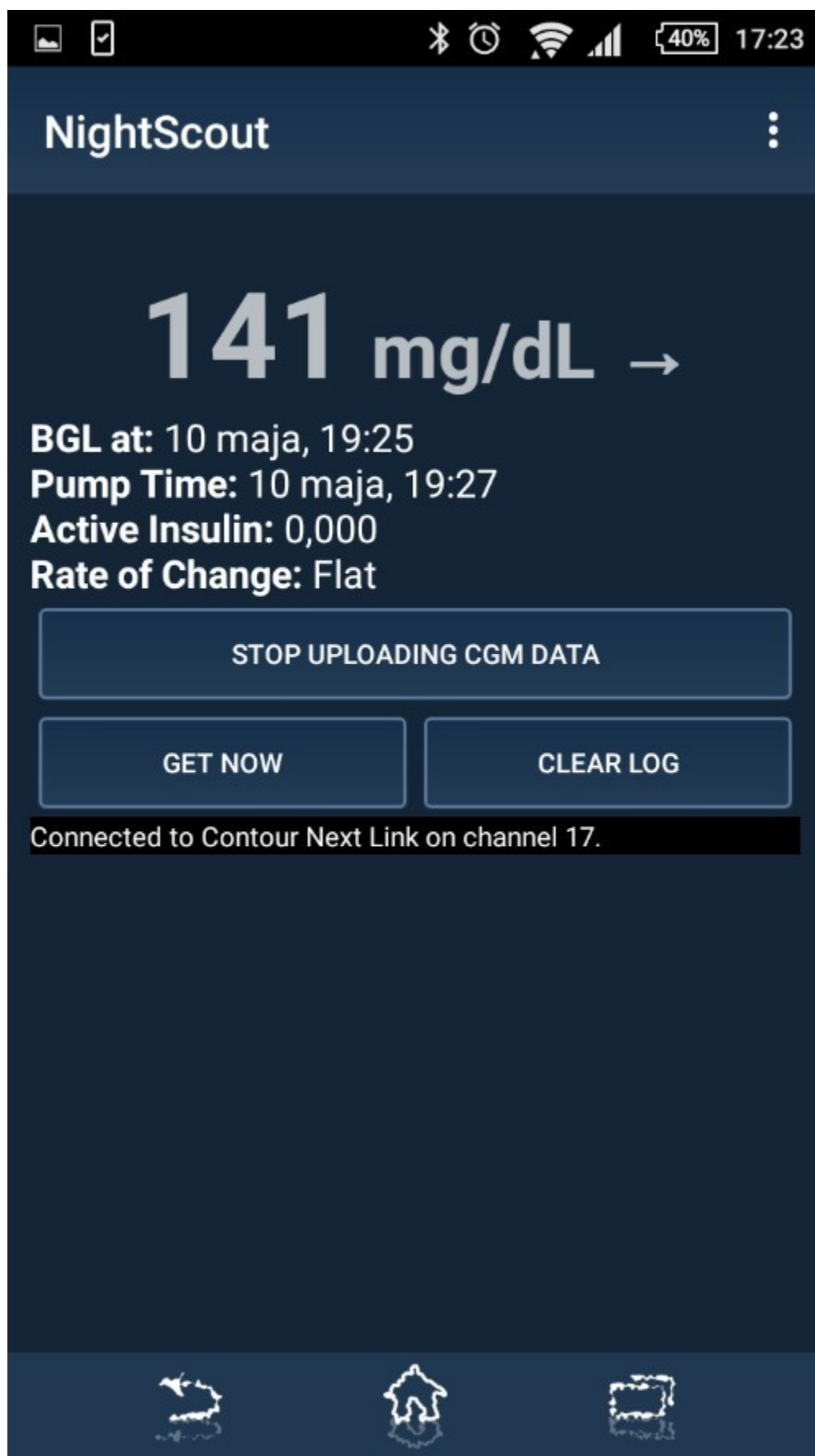
Klicke nun auf „Register USB Stick“:



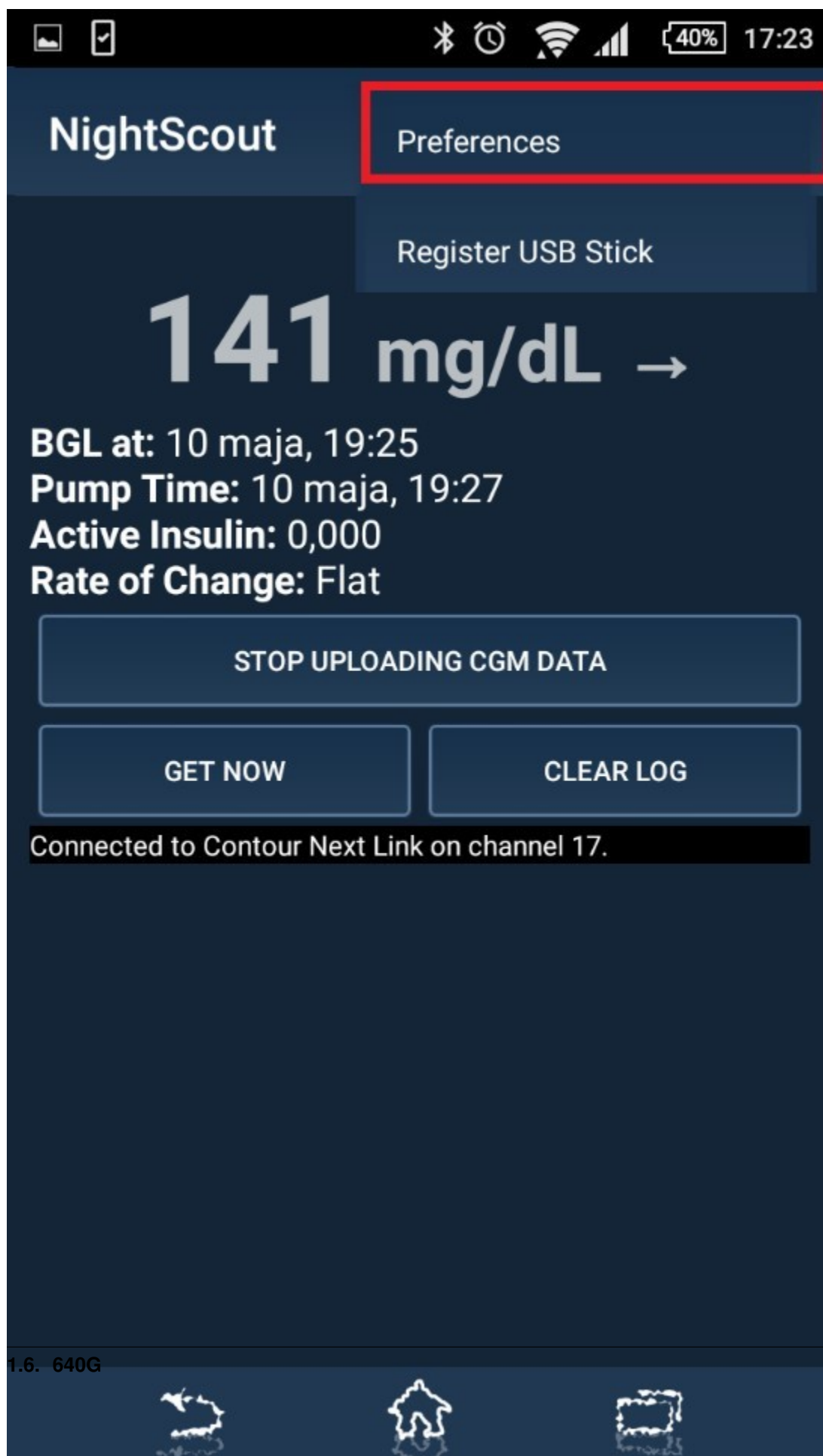
Gib nun deine Login Informationen von carelink.minimed.eu ein und drücke „RETRIEVE KEYS FOR USB“:



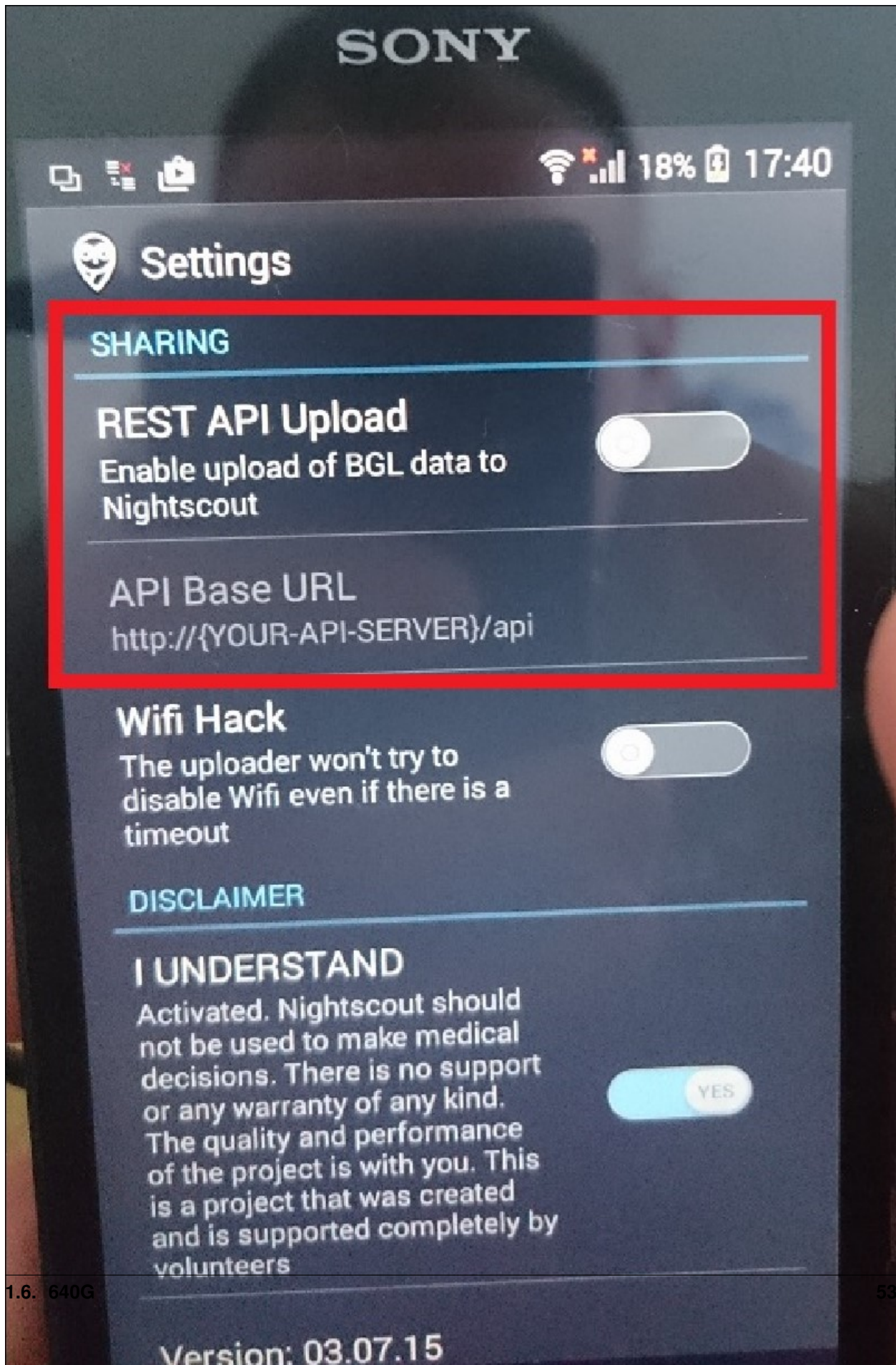
Wenn die App den richtigen Schlüssel von CareLink bekommt, siehst du nun auf dem Bildschirm den ersten CGM-Wert: Gratuliere!



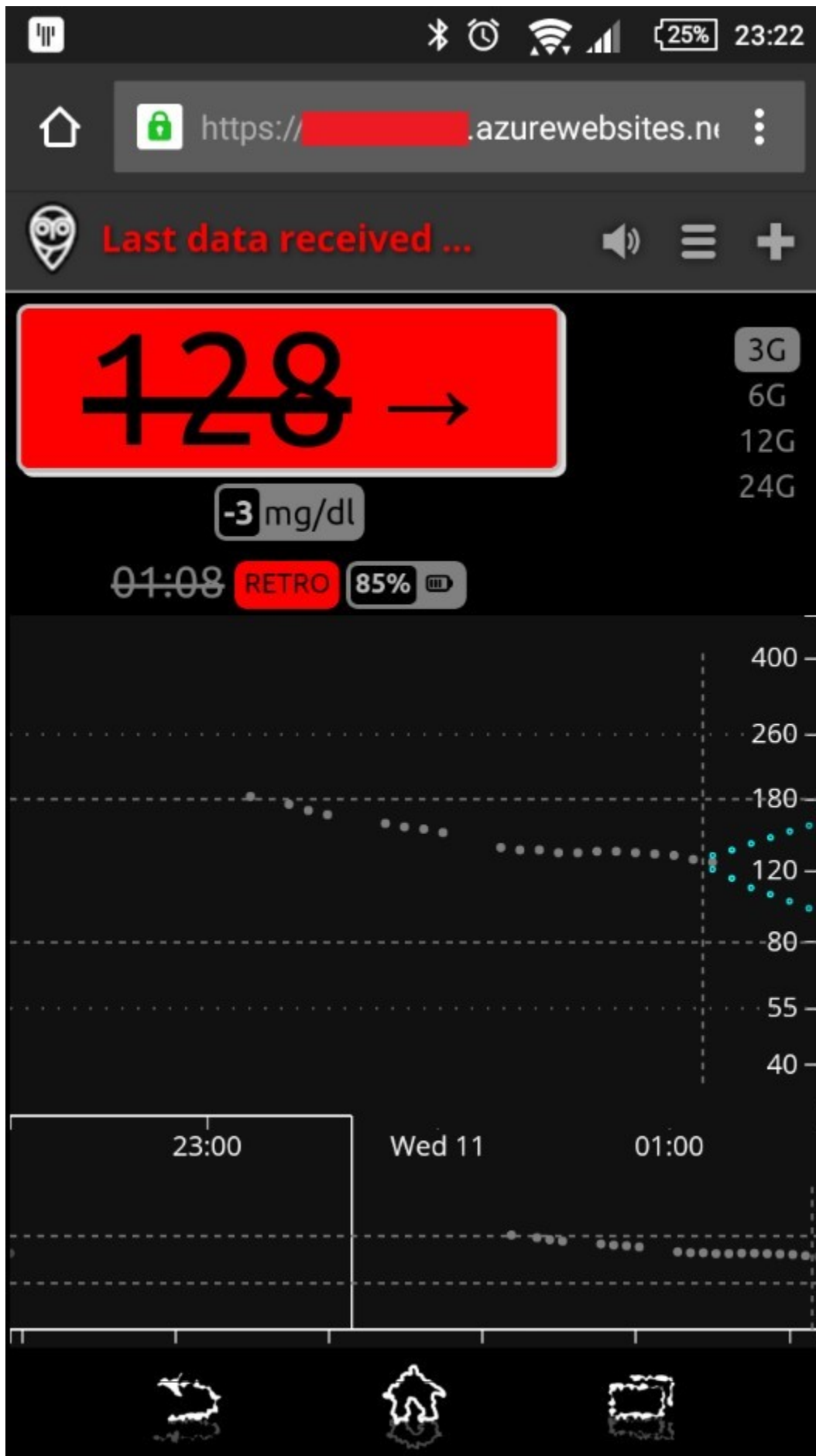
Damit die CGM-Werte nun auch auf anderen mobilen Geräten abgelesen werden können, müssen wir noch einen Schritt machen. Gehe wiederum ins Menü der App und drücke auf „Preferences“:



Aktiviere nun „REST API Upload“. Je nach Version der App musst du noch deine API Base URL eingeben, die so aussieht: `https://deinAPI_SECRET@deinewebsite.azurewebsites.net/api/v1/` Es kann sein, dass dein API_SECRET Passwort und deine Nightscout Website (deinewebsite.azurewebsites.net) separat verlangt werden.



Nun kannst du auf deiner Website (deinewebsite.azurewebsites.net) die CGM-Werte ablesen (es kann einige Minuten in Anspruch nehmen, bis die Daten angezeigt werden):



Als App zum Lesen der Kurve für Android empfehle ich „NightWatch“. Für iPhones empfehle ich die Apps „Nightscouter“ oder „Nightscout“.

Sobald die Verbindung zwischen der Pumpe und der Android-Messgerät-Verbindung unterbrochen wird (Distanz, Wackelkontakt des USB-OTG Kabels), werden keine CGM-Werte mehr aufgezeichnet. Sie werden zum jetzigen Zeitpunkt auch nicht nachgetragen, sobald die Verbindung wieder steht.

Bei Problemen kannst du jederzeit die Community auf Facebook um Rat bitten (Gruppe für Deutschland heißt „Nightscout Deutschland (Germany)“).

Nun viel Spaß mit Nightscout - #WeAreNotWaiting !

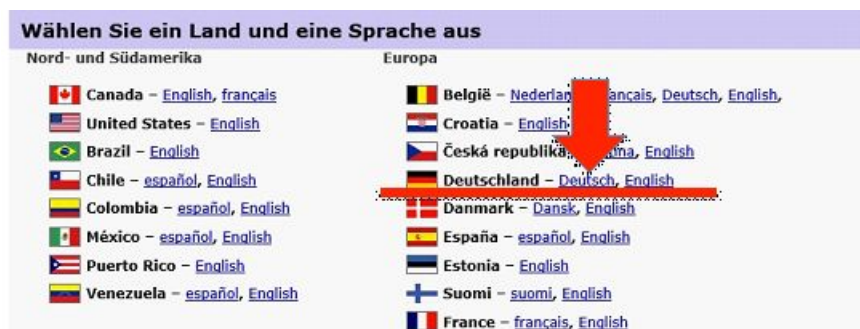
1.6.2 Anlegen eines Kontos auf <https://carelink.minimed.eu>

Wir benötigen ein Konto auf CareLink und müssen mindestens einmal die Pumpendaten über das Contour Next Link 2.4 Messgerät auf Carelink hochladen.

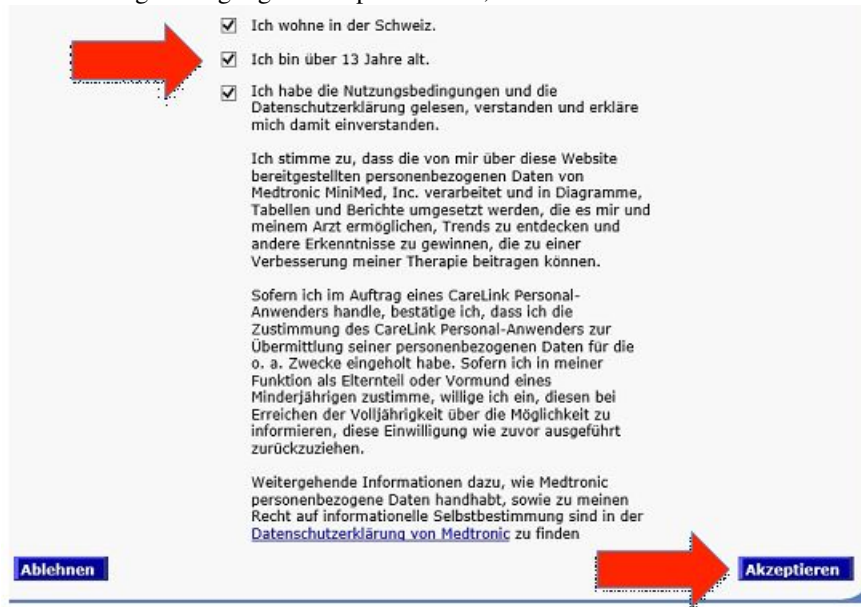
Gehe auf <https://carelink.minimed.eu> und klicke auf „Jetzt anmelden“:



Wähle dein Land und deine Sprache aus:



Nach dem Durchlesen der Nutzungsbedingungen akzeptiere diese, indem du diese drei Checkboxes klickst und



☒ Ich wohne in der Schweiz.

☒ Ich bin über 13 Jahre alt.

☒ Ich habe die Nutzungsbedingungen und die Datenschutzerklärung gelesen, verstanden und erkläre mich damit einverstanden.

Ich stimme zu, dass die von mir über diese Website bereitgestellten personenbezogenen Daten von Medtronic MiniMed, Inc. verarbeitet und in Diagramme, Tabellen und Berichte umgesetzt werden, die es mir und meinem Arzt ermöglichen, Trends zu entdecken und andere Erkenntnisse zu gewinnen, die zu einer Verbesserung meiner Therapie beitragen können.

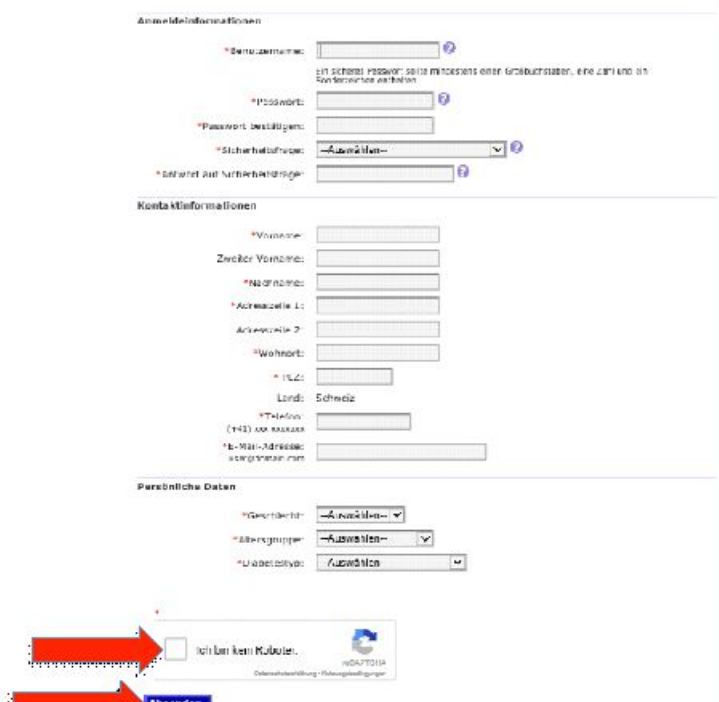
Sofern ich im Auftrag eines CareLink Personal-Anwenders handle, bestätige ich, dass ich die Zustimmung des CareLink Personal-Anwenders zur Übermittlung seiner personenbezogenen Daten für die o. a. Zwecke eingeholt habe. Sofern ich in meiner Funktion als Elternteil oder Vormund eines Minderjährigen zustimme, willige ich ein, diesen bei Erreichen der Volljährigkeit über die Möglichkeit zu informieren, diese Einwilligung wie zuvor ausgeführt zurückzuziehen.

Weitergehende Informationen dazu, wie Medtronic personenbezogene Daten handhabt, sowie zu meinen Recht auf informationelle Selbstbestimmung sind in der [Datenschutzerklärung von Medtronic](#) zu finden

Ablehnen **Akzeptieren**

„Akzeptieren“ klickst:

Fülle nun das Formular mit deinen Angaben aus. Bestätige, dass du kein Roboter bist, und klicke auf „Absenden“:



Anmeldeinformationen

* Benutzername:

Ein sicherer Passwort sollte mindestens einen Großbuchstaben, eine Zahl und ein Sonderzeichen enthalten

* Passwort:

* Passwort bestätigen:

* Sicherheitsfrage:

* Antwort auf Sicherheitsfrage:

Kontaktinformationen

* Vorname:

Zweiter Vorname:

* Nachname:

* Adressezeile 1:

Adressezeile 2:

* Wohnort:

* PLZ:

Land: Schweiz

* Telefon (inkl. Vorzeichen):

* E-Mail-Adresse:

Persönliche Daten

* Geschlecht:

* Altersgruppe:

* U-Identifizierung:

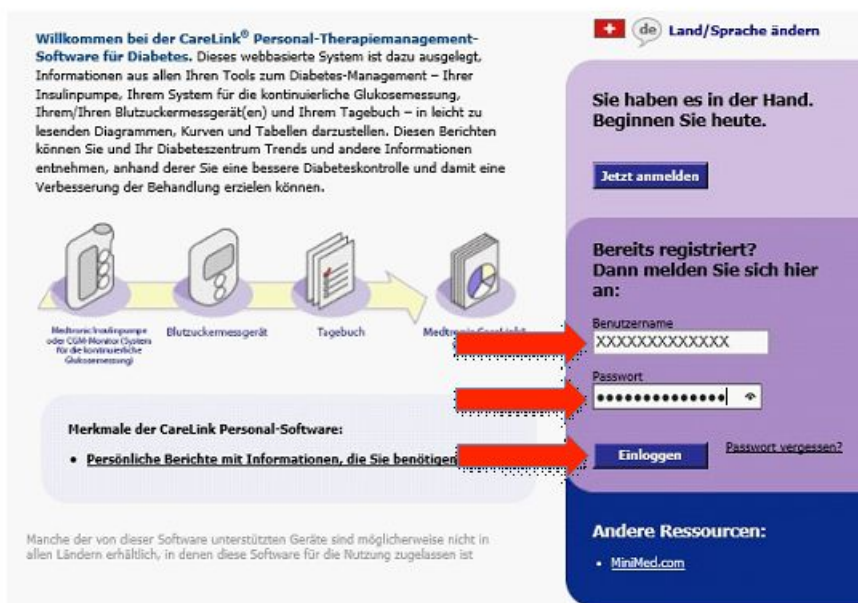
☒ Ich bin kein Roboter.

Absenden

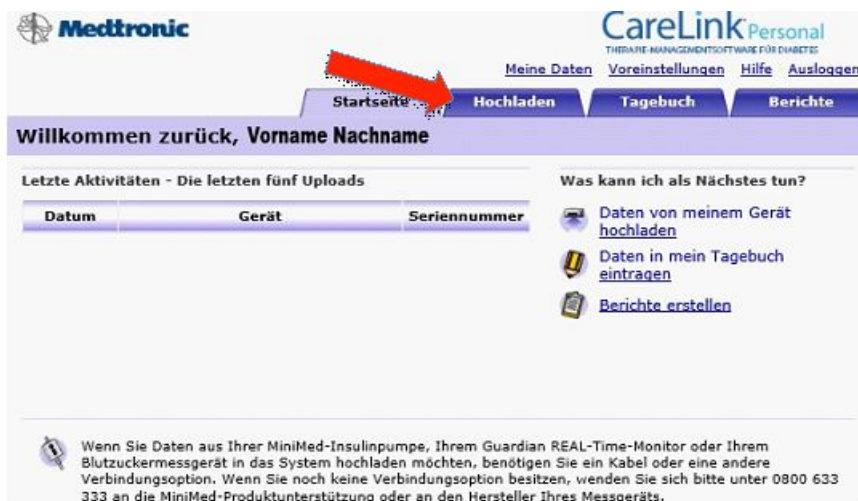
Nach dem „Absenden“ erscheint eine Bestätigungsseite:



Nachdem du auf „Fertig stellen“ klickst, kommst du auf die Startseite zum Login. Gib deine vorhin erstellten Benutzernamen und Passwort ein und klicke auf „Einloggen“:



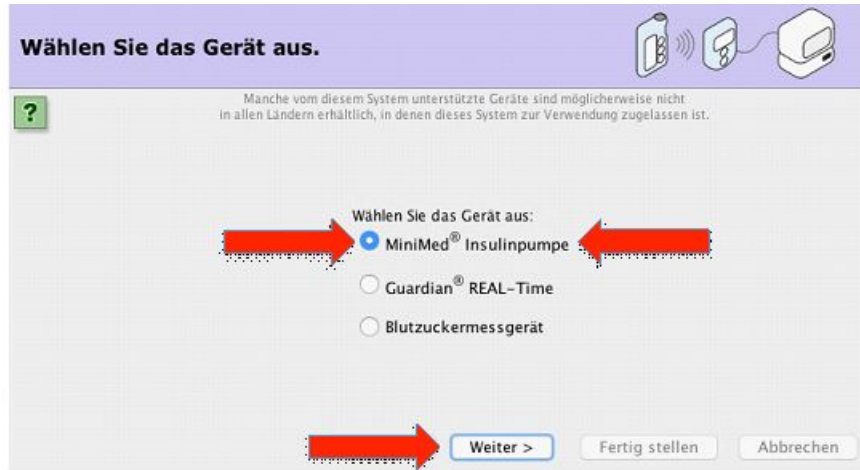
Wechsle nun auf den Reiter „Hochladen“:



Stecke nun dein Contour Next Link 2.4 Messgerät in einen USB-Stecker deines Computers. **WICHTIG** ist, dass dieses Messgerät mit der Minimed 640G Pumpe bereits verbunden/gepaart wurde. (Eine Anleitung dazu findest du [hier](#).) **Die Pumpe** bzw. der/die Pumpenträger_in muss sich nun in der Nähe des Computers bzw. des angeschlossenen Messgeräts befinden.

(Java muss auf dem Computer installiert sein und die Ausführung von Java erlaubt werden).

WICHTIG ist, als Gerät „Minimed Insulinpumpe“ auszuwählen. Klicke nun auf „Weiter“



Wähle nun „Minimed Serie 600“ und klicke auf „weiter“:

Du wirst nun gebeten

- eine laufende Bolusabgabe abzuwarten/abzubrechen
- eine laufende temporäre Basalrate abzuwarten/abzubrechen
- zu kontrollieren, dass die Pumpenbatterie nicht schwach ist
- etwaige Fehlermeldungen auf der Pumpe zu löschen

Sofern alles stimmt, klicke auf „Weiter“



Nun musst du die Seriennummer der Pumpe eingeben und auf „Weiter“ klicken:



Wähle nun das Contour Next Link Messgerät aus und klicke auf „Weiter“:



Stelle sicher, dass alle aufgeführten Punkte zutreffen und klicke auf „Fertigstellen“:



Die Pumpendaten werden nun ausgelesen:



Wichtig:

Sollte die Verbindung zwischen Pumpe und Messgerät unterbrochen werden (z.B. durch das Auslesen der Pumpendaten in CareLink Pro bei einer 3-Monatskontrolle) kann es sein, dass das Auslesen auf CareLink (wie eben durchgeführt) erneut notwendig wird. Damit das nicht passiert empfiehlt es sich, ein zusätzliches Contour Next Link 2.4 Messgerät anzuschaffen, welches nur in Verbindung mit dem Android Smartphone steht und nicht für etwaiges manuelles Hochladen von Pumpendaten gebraucht wird (muss allerdings ebenfalls einmalig zum Auslesen der Pumpendaten gebraucht worden sein!).

1.7 Libre

Das Freestyle-Lesegerät selbst ist nicht Nightscout-fähig, jedoch die [Android-App Glimp](#), die auch Sensoren auch auslesen kann. Dadurch ist eine Fernüberwachung möglich - z.B. durch die Eltern, die dann auf dem selben Stand wie das Kind sind und sehen ob und was gemessen wurde. Mit der Android-App werden allerdings z.T. leicht andere Werte ausgelesen als vom Lesegerät, da diese einen eigenen Algorithmus verwendet.

1.7.1 Nightscout mit FreeStyle Libre

Um Nightscout mit dem FreeStyle Libre nutzen zu können, braucht man:

- ein funktionierendes FreeStyle Libre System
- eine eingerichtete Nightscout Seite
- ein Android Smartphone, das NFC (Nahfeldkommunikation) unterstützt
- die [Glimp App](#) aus dem Google Play Store

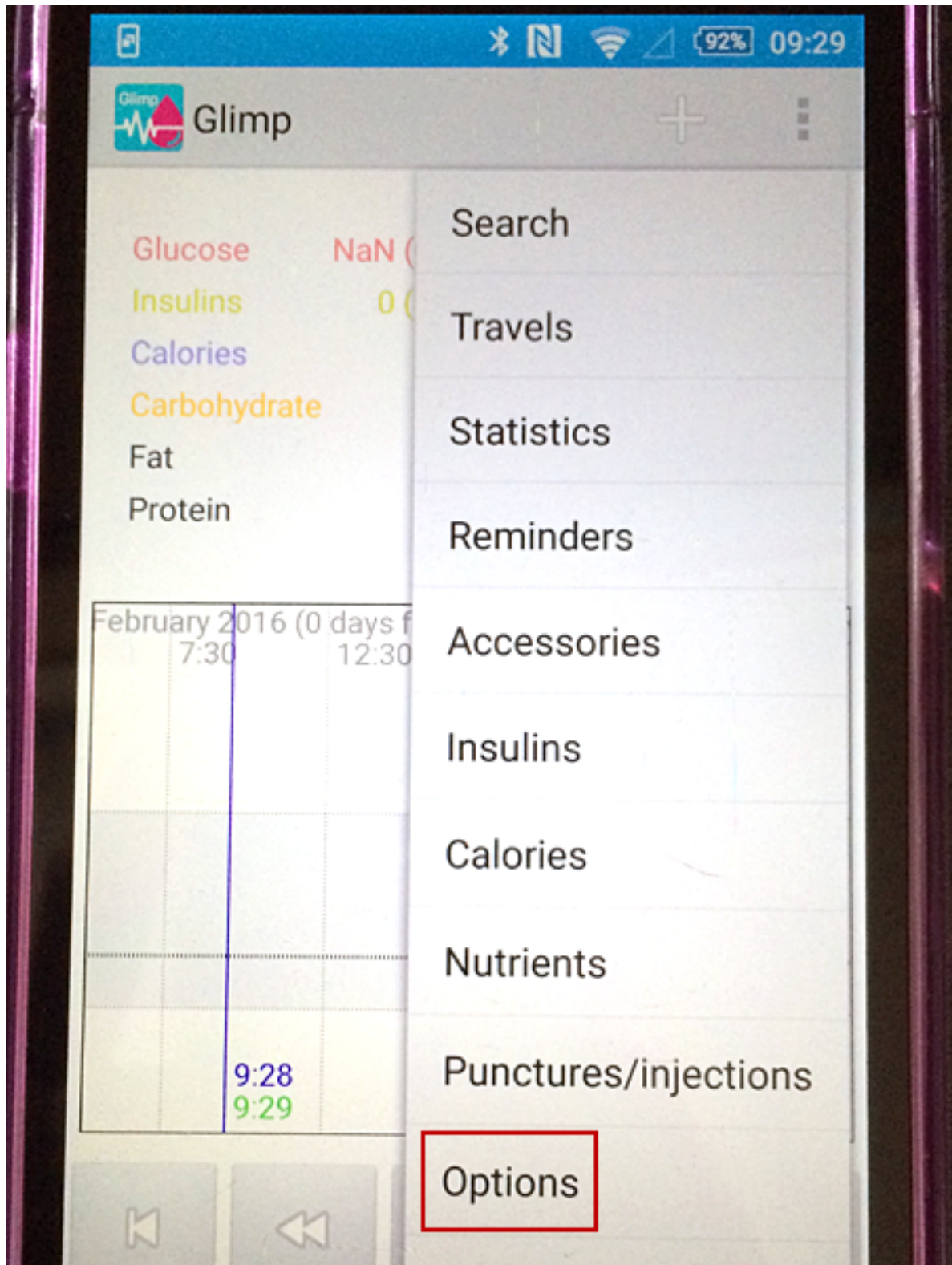
Falls man bisher noch keine Nightscout Seite eingerichtet hat, findet man alle erforderlichen Schritte im Kapitel [Nightscout einrichten](#). Wichtig ist dabei darauf zu achten, dass in Azure bzw. Heroku das CarePortal eingeschaltet wurde (dies geschieht, in dem man “careportal” bei der “ENABLE” Variable eingefügt. Auch muss in Azure bzw. Heroku ein “API_SECRET” gesetzt worden sein. Dies klingt komplizierter als, es in Wirklichkeit ist. Am besten Schritt für Schritt der Anleitung folgen ;-)

Viele aktuelle Android Smartphones unterstützen die NFC Technologie. Eine sehr gute, englischsprachige Liste findet man [hier](#).

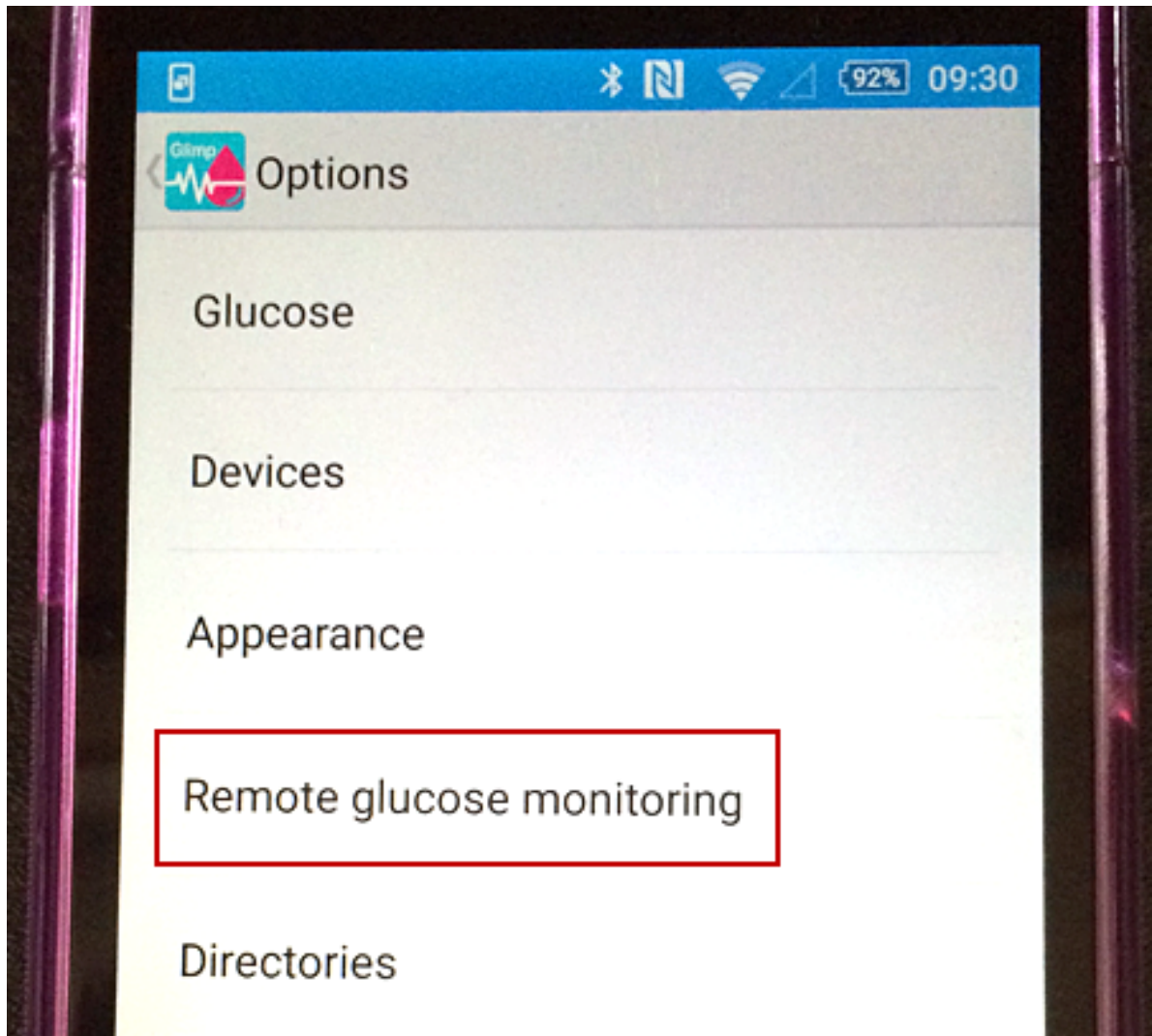
1.7.2 Daten des FreeStyle Libre mit Glimp hochladen

Um die Daten auf der Nightscout Seite angezeigt zu bekommen, müssen diese mit der Glimp App über das Smartphone vom Sensor gelesen werden und die App sendet die Daten dann zu Nightscout. Folgende Schritte sind dazu nötig:

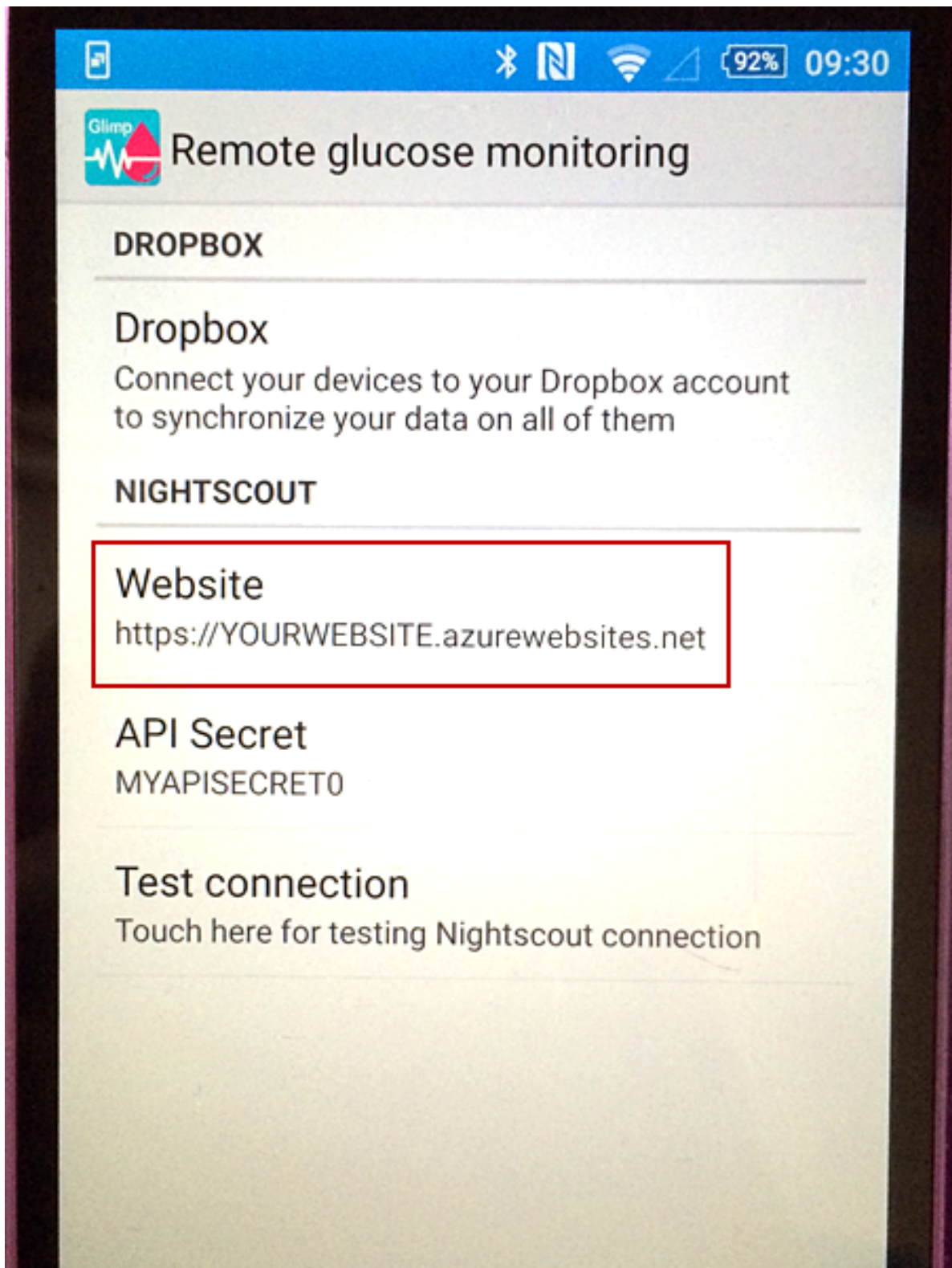
1. die [Glimp App](#) aus dem Google Play Store laden
2. Glimp konfigurieren damit es mit dem FreeStyle Libre zusammen arbeitet - das bedeutet am Telefon NFC einschalten und mit dem Smartphone den bereits gestarteten Sensor scannen. Wenn nötig, dann kann man in der App auch noch eine Kalibrierung veranlassen, in dem man auf das “+” Zeichen klickt und einen BZ Wert eingibt (eventuell mehrmals wiederholen).
3. Glimp für die Zusammenarbeit mit Nightscout konfigurieren. Dazu auf die 3 Punkte in der rechten, oberen Ecke klicken und danach auf “**Optionen**” klicken.



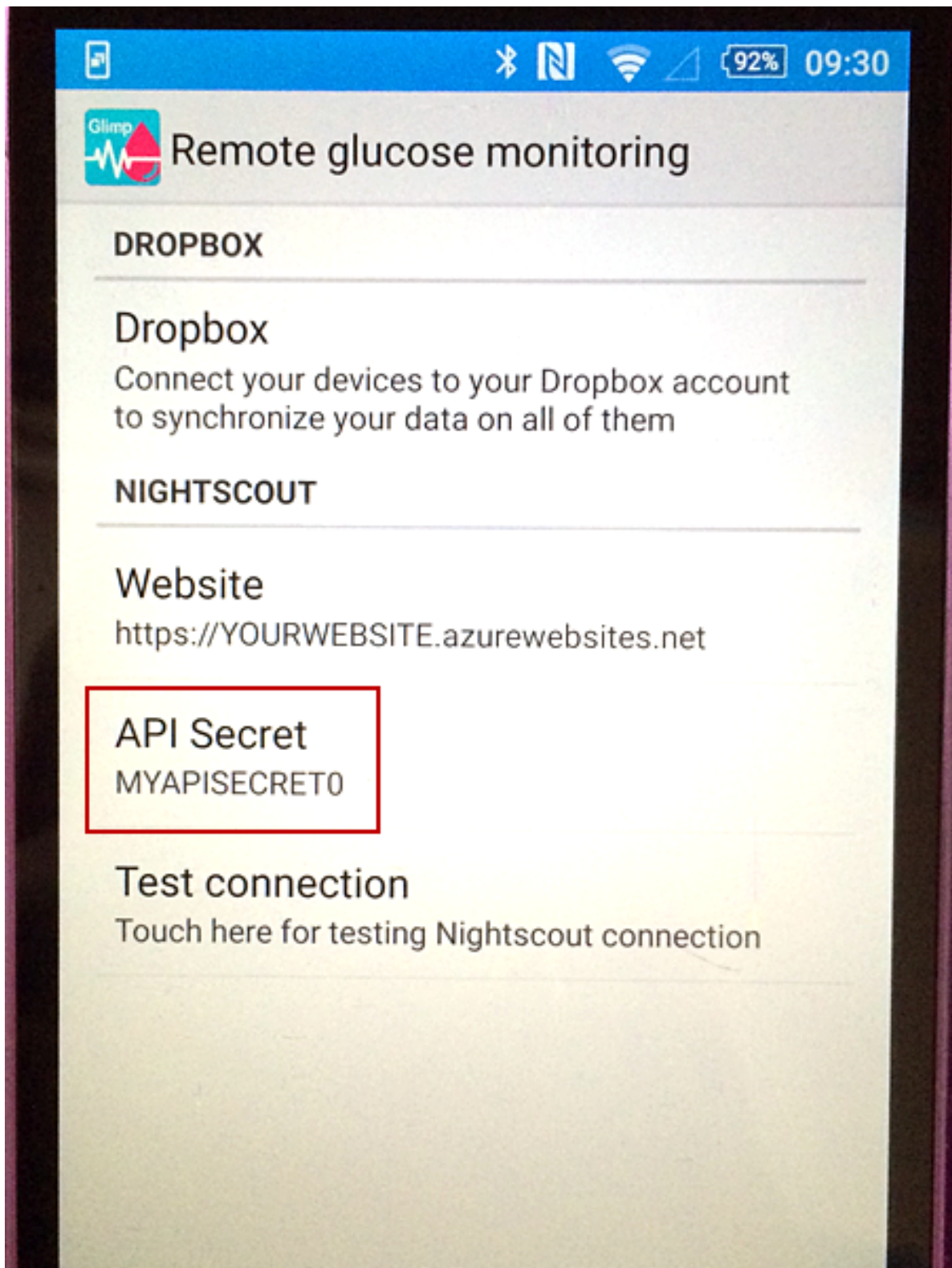
4. Jetzt “**entferntes Glucose-Monitoring**” wählen.



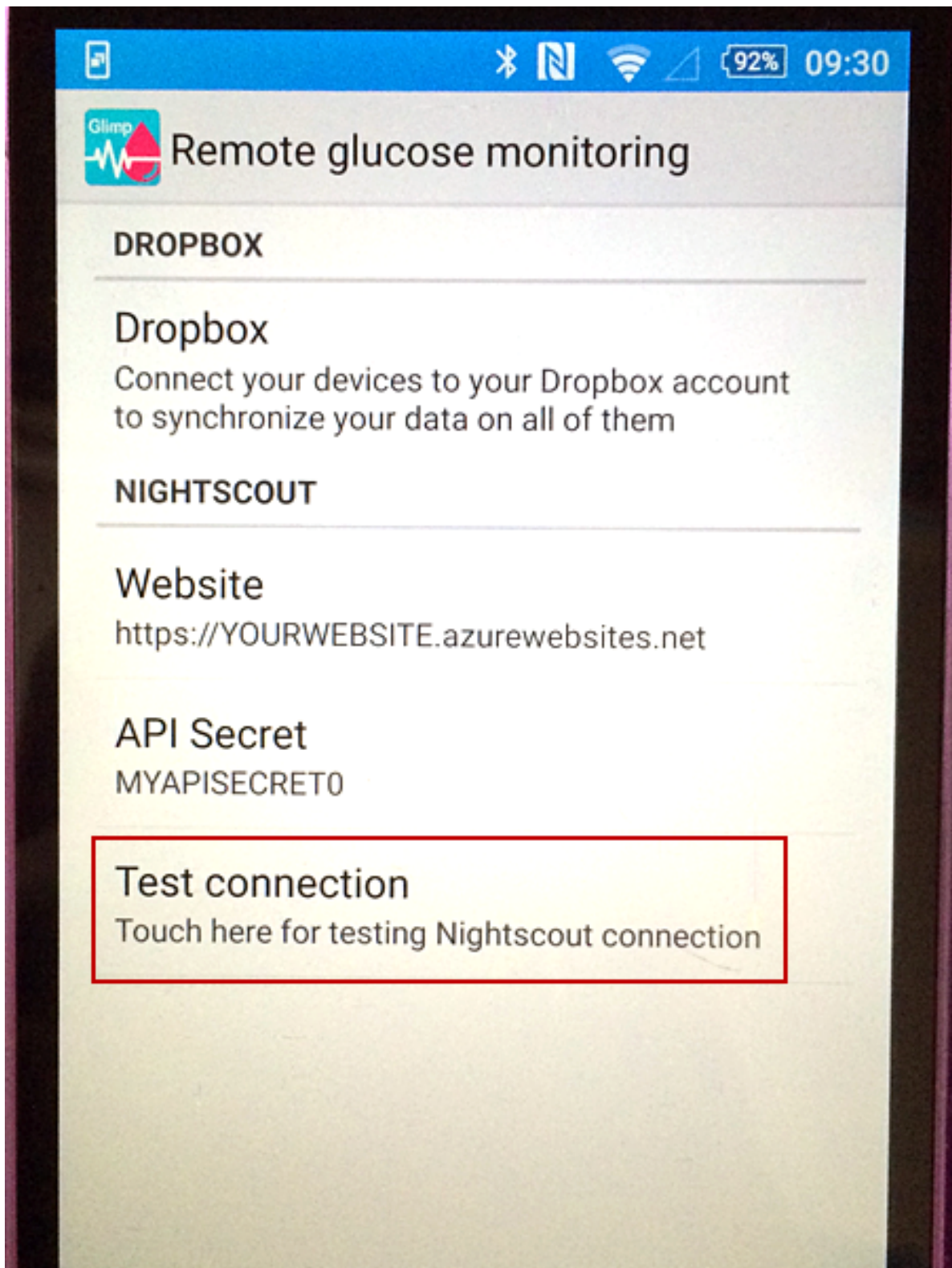
5. Auf der “entferntes Glucose-Monitoring”-Seite, gibt man jetzt die vorher erstellte Nightscout URL ein.



6. Jetzt das API SECRET eingeben (Anmerkung: das API SECRET *muss* identisch sein, mit das man bei der Konfiguration der Nightscout Seite bei Azure bzw. Heroku gesetzt hat.)



7. Nun auf **“Test connection”** klicken.



8. Wenn der Test erfolgreich war, dann ist Glimp nun fertig konfiguriert und lädt ab jetzt die mit der Glimp App gescannten Werte des Libre auf die Nightscout Seite.

1.7.3 Libre als CGM

LimiTter

Dank eines DIY Projektes ist es bereits heute möglich, den Libre Sensor mit einem selbst zu bastelnden Transmitter auszustatten, der dann automatisch alle 5 Minuten Werte scannt und an ein Android Smartphone überträgt. Auf diesem muss die XDrip App installiert sein, die die Daten sowohl anzeigen kann, als auch an Nightscout oder eine Smartwatch senden. Wie das geht steht [hier](#) auf Englisch.

blueReader

Mittlerweile gibt es auch ein Projekt, das sich an alle wendet, die nicht so gerne mit dem Lötkolben arbeiten und trotzdem den Libre mit dem Funktionsumfang eines CGM-Systems nutzen möchten. Der blueReader soll wasserdicht und klein sein. Wer sich über den aktuellen Stand informieren möchte, kann dies in dem Blog [Sandras Place](#) machen. Damit dieses Projekt allerdings realisiert werden kann, wurde ein [crowdfunding Projekt](#) ins Leben gerufen, welches auf finanzielle Unterstützung angewiesen ist.

Nightscout besteht aus **3 Komponenten**:

- Einer Dokumentations-Datenbank (MongoDB) zur Speicherung von Blutglukose, Kohlenhydraten, Insulin, Aktivitäten usw. MongoDB ist eine Cloudanwendung und kann über das Internet erreicht werden.
- Einer **Web-Applikation**, welche über einen beliebigen Browser oder eine App aufgerufen werden kann. Die Anwendung wird in einem Verzeichnis aktualisiert, welches über einen webbasierten Filehosting Dienst [GitHub](#) verwaltet wird. Dieser Dienst ist grob vergleichbar mit Webdiensten wie z.B. Google Docs. Die Grundidee dabei ist, anderen ein Verzeichnis mit Dateien zur Verfügung zu stellen, damit man gemeinsam daran arbeiten kann. Nightscout besteht aus vielen Plugins zur individuellen Konfiguration. Die Plugins mit Parametern sind im Kapitel *Nightscout Plugins* beschrieben.
- Einem Cloud-Service Anbieter, welcher die Web-Applikation über Webserver zur Verfügung stellt. Microsoft [Azure](#) oder [Heroku](#) können hierfür verwendet werden. Cloud-Service Anbieter betreiben weltweit Rechenzentren, man konfiguriert webbasiert wie hier Nightscout diese Dienste. Das Hosting ist hier nicht zwingend notwendig, eigene standalone - Lösungen bei einem entspr. Provider sind auch möglich.

WIR EMPFEHLEN MITTLERWEILE DIE EINRICHTUNG ÜBER HEROKU. An der deutschen Anleitung dafür wird noch gearbeitet. Bis dahin: https://loopkit.github.io/loopdocs/nightscout/new_user/

Wir können Azure auf verschiedene Weisen einrichten:

Geführte Installation:

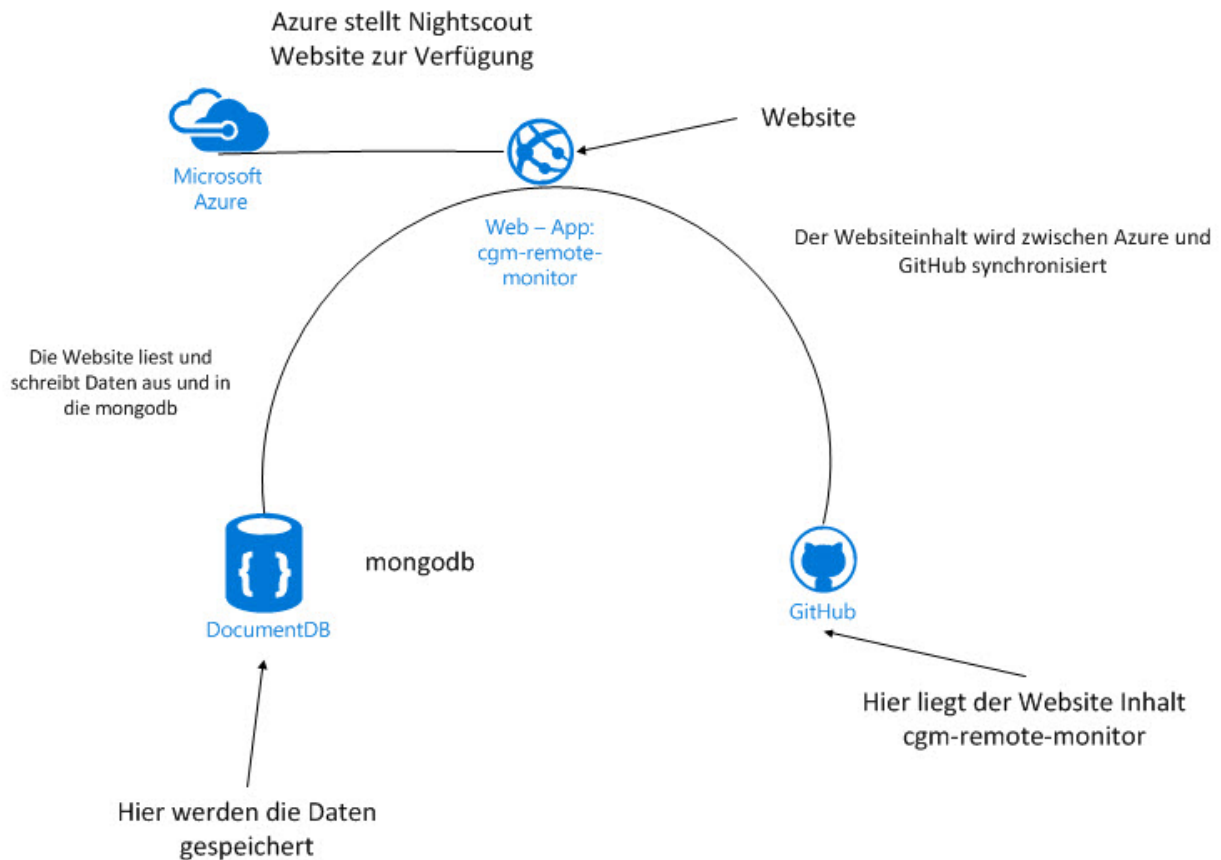
Die Nightscout Website lässt sich fast vollkommen automatisiert einrichten. Dieses sowohl mit **Azure** als auch mit **Heroku** als Provider. Dieses ist in Kapitel *Der Deploy Azure Button* beschrieben.

Lokale Installation:

Nightscout lässt sich auch lokal, z.B. auf einem Rasberry PI installieren. Dieses hat den Vorteil, das keine Internetverbindung zum Auslesen von Daten erforderlich ist. Die mongodb, ein Webservice und die Website müssen dazu installiert und konfiguriert werden. Dieses ist in Kapitel *Installation ohne Cloud Service Provider* beschrieben. Es steht am Anfang und wird weiter bearbeitet.

Klassisch: Die klassische Einrichtung erfordert das manuelle Anlegen der Azure Website mit allen Einstellungen. Dieses ist in Kapitel *Azure* beschrieben.

Ein Schaubild verdeutlicht die Zusammenhänge:



Contents:

2.1 Account-Information (Arbeitsblatt)

Bevor wir mit der Einrichtung von Nightscout beginnen, ist es sinnvoll, das Arbeitsblatt auszudrucken. Wir notieren alle wichtigen Benutzernamen, Passwörter und Zugangsdaten, die bei der ersten Einrichtung festgelegt werden. Am besten füllt man es direkt während der Einrichtung gewissenhaft aus und heftet es ab, um später darauf zugreifen zu können.

[Download Arbeitsblatt \(deutsch\)](#)

[Download Original Arbeitsblatt \(english\)](#)

Wir starten mit dem Einrichten der [Mongo Datenbank](#).

2.2 Einrichten der Mongo-Datenbank

Die [Mongo DB Datenbank](#) ist eine Dokumentations - Datenbank. Hier werden die Daten der Nightscout Website gespeichert. Es handelt sich um eine Cloud - Datenbank in Rechenzentren der USA und Europa.

Mongo DB gibt es als **kostenlose** Sandbox - Variante.

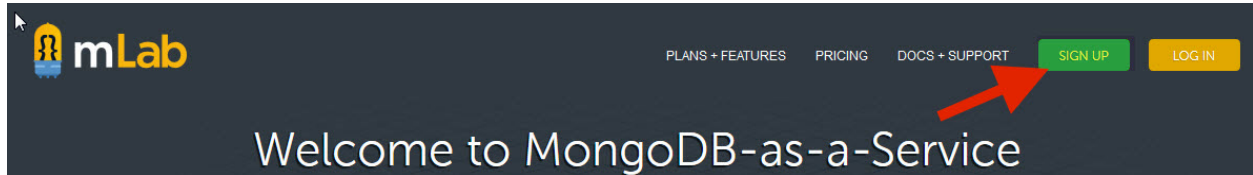
Bevor wir mit der Konfiguration loslegen, ist es wichtig, das [Arbeitsblatt](#) auszudrucken und auszufüllen.

Es ist für weitere Konfigurationen wichtig, dieses Datenblatt sicher aufzubewahren.

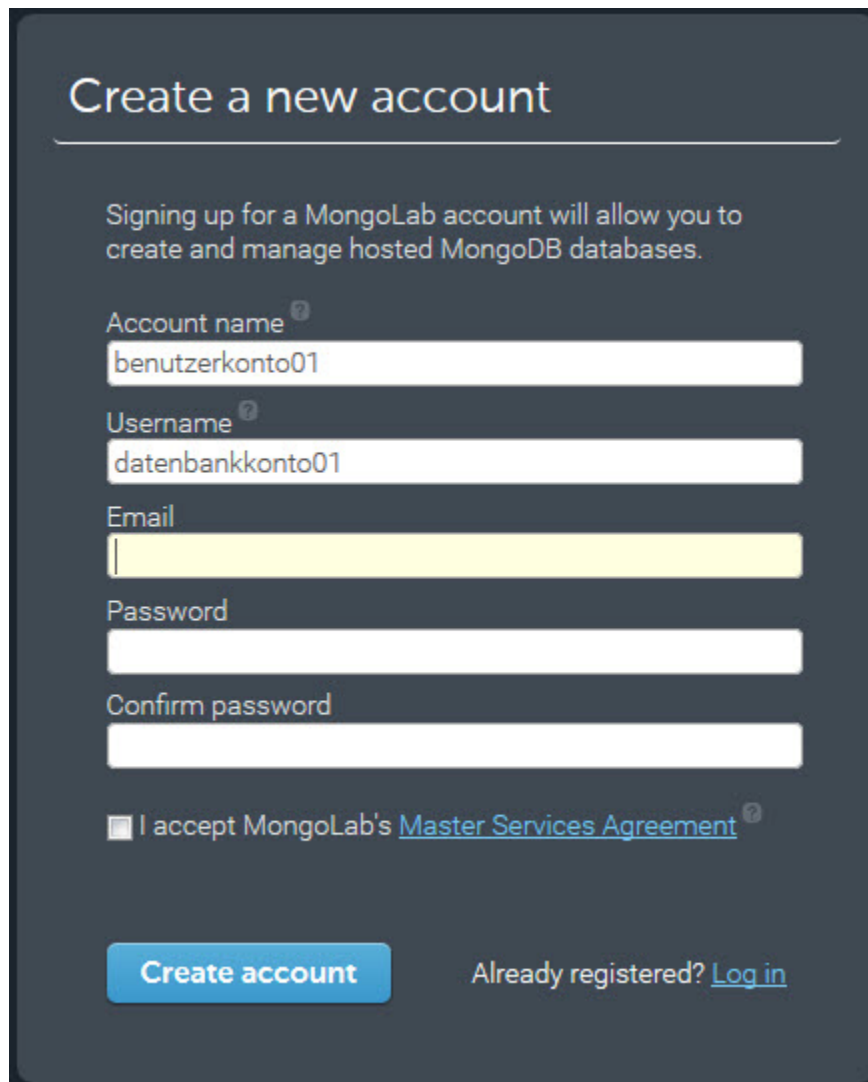
Die URL Adresse zum Start gibt es hier: <https://mlab.com>

Wir starten mit der Einrichtung der Benutzer - Konten auf der mlab - Startseite:

SIGN UP:



Weiter gehts mit dem **Anlegen** der benötigten **Konten:** zur Verwaltung



Create a new account

Signing up for a MongoLab account will allow you to create and manage hosted MongoDB databases.

Account name [?]

Username [?]

Email

Password

Confirm password

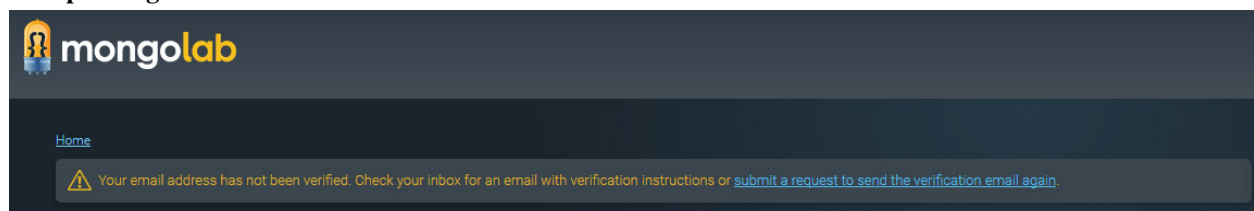
☐ I accept MongoLab's [Master Services Agreement](#) [?]

[Create account](#) Already registered? [Log in](#)

Passwörter

Der Einfachheit halber kann man denselben Benutzernamen (Account name) für die Verwaltung von mongolab als auch für den Datenbanknutzer (Username) nutzen, muss es aber nicht. Hauptsache: **Alles notieren**. Auch sind eigene Passwörter nur für mlab und Nightscout sinnvoll, es sollten auf keinen Fall bereits bekannte Passwörter, welche man für andere Anlässe wie Online-Banking nutzt, verwendet werden.

Überprüfung Email:

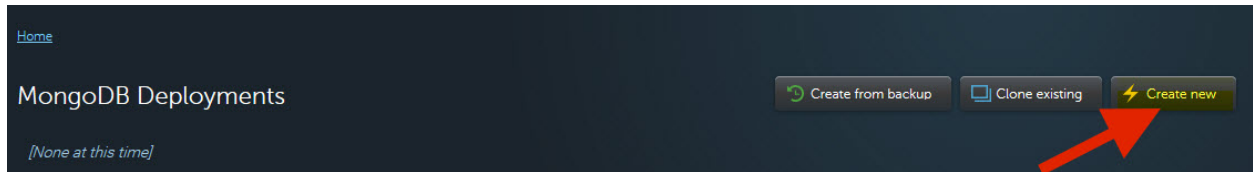


Nachdem der Benutzer - Account eingerichtet wurde, bitte einmal im Postfach der angegebenen Email - Adresse die Bestätigungs - Email suchen. Durch Klick auf den Link wird man auf die Seite zur Einrichtung einer Datenbank

geleitet.

Anlegen einer Datenbank:

Wir starten mit dem Klick auf “Create New”:



Alle gelb markierten Punkte sind auszufüllen, der Datenbankname ist frei wählbar:

[Home](#)

Create new subscription

Fill out this form to create a brand-new MongoDB deployment in the cloud location of your choice.
Alternatively, you can [clone an existing deployment](#).

Cloud provider:



Location: Azure's North Europe (Dublin) Location (North Europe) ▼

Plan ([view pricing page](#)) :

Single-nodeReplica set cluster

These plan(s) are perfect for development/testing/staging environments as well as for utility instances that do not require high-availability.

● Sandbox (shared, 0.5 GB)FREE

MongoDB version: 3.0.x ▼

Database name:

mycgmdatabase

Price:

\$0 / month

 Create new MongoDB deployment

Wichtig ist er Eintrag FREE, damit wird die Datenbank kostenlos betrieben.

Nachdem der Button **Create new MongoDB deployment** geklickt wurde, erscheint eine Erfolgsmeldung:

Home

Success! Your new database, nscgmdatabase, is listed below.

MongoDB Deployments

Create from backup Clone existing Create new

Development and Utility Single-node deployments intended for environments that do not require high-availability.

NAME	PLAN	RAM	SIZE	FILE SIZE
ds040888/nscgmdatabase	Sandbox	shared	0.00 KB	0.00 KB

Durch Doppelklick auf den Namen, hier: **ds040888/nscgmdatabase** fügt man u.a. den Datenbankbenutzer ein. Das ist derjenige, welcher in Azure oder xdrip konfiguriert wird, um auf die mongodb zuzugreifen:

Datenbankbenutzer hinzufügen:

Home

Database: nscgmdatabase Delete database

To connect using the mongo shell:

```
% mongo ds040888.mongo1ab.com:40888/nscgmdatabase -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds040888.mongo1ab.com:40888/nscgmdatabase
```

mongod version: 3.0.7

A database user is required to connect to this database. [Click here](#) to create a new one.

Collections Users Stats Backups Tools

Database Users

Add database user

..und die vorher notierten Daten eingeben:

WICHTIG:

Dabei darauf achten, dass bei “Make read-only” kein Haken gesetzt ist, da man Schreibrechte auf die Datenbank braucht.

Hinzufügen einer “Collection”:

Unter Collections auf “add collection” gehen, einen Namen wie z.B. “entries” eintragen und auf “create” tippen. Dieser Name wird später bei azure unter Anwendungseinstellungen als mongo collection eingefügt.

WICHTIG:

Am Ende bitte unbedingt die MongoDB [URI](#) notieren. Diese wird benötigt, um auf die Daten über das Internet zugreifen zu können.

Die URI Syntax:

Zum besseren Verständnis ausführlich erklärt:

mongodb://dbuser:dbpassword@ds040888.mlab.com:40888/nscgmdatabase

dbuser: ist der vorher eingerichtete Datenbankbenutzer

dbpassword: ist das eingerichtete Passwort für den Datenbankbenutzer

@: ist ein Verbindungszeichen, muss angegeben werden

ds040888.mlab.com: ist ein einzigartiger Name, über den die mongodb angesprochen wird

: ist ein Verbindungszeichen, muss angegeben werden

40888: ist ein sogenannter [Kommunikationsport](#), der für den Zugriff benötigt wird

nscgmdatabase: Name der mongodb

Diese Daten bitte unbedingt im Datenblatt notieren. Bei evtl. Verbindungsproblemen ist eine der häufigsten Ursachen ein falsche URI, es kommt hier auf Details, auf jedes Zeichen an!

Es gibt **optionale** Komponenten zum Management der MongoDB im nächsten Kapitel, diese sind nicht Bestandteil der Implementierung.

Ansonsten geht es jetzt weiter mit der Einrichtung von [GitHub](#)

2.2.1 Mongo DB Administration

Dieser Teil ist optional

Mongo DB ist eine Dokumentations - Datenbank, welche aus Collections (Sammlungen von Dokumenten) und Dokumenten besteht. Mongo DB ist eine NoSQLDB und schemafrei. Die Dokumente werden u.a. im JSON Format abgespeichert. <http://www.json.org/json-de.html> Die Dokumente können direkt in die DB geladen werden. Um eine Dokumenten DB zu administrieren, ist eine Admin - GUI: https://de.wikipedia.org/wiki/Grafische_Benutzeroberfl%C3%A4che sinnvoll.

Ein Anbieter ist z.B. 3T Software Labs aus Berlin.

Der 3T MongoChef <http://3t.io/mongochef> ist für den Privatgebrauch kostenlos.

Unter: <http://3t.io/mongochef/download> muss man sich registrieren und kann dann die Software für das jeweilige Betriebssystem herunterladen.

Mit dieser Admin GUI ist folgendes möglich:

- **Kopieren von Datenbanken und Collections**

Dieses ist hilfreich, wenn man die mongodb umziehen möchte, z.B. von einem Standort aus den USA zu einem Standort nach Europa. Man kann es auch als Backup Möglichkeit nutzen

- **Im- und Export von Dokumenten**

Man kann sowohl csv-, als auch json - Dokumente im- bzw. exportieren. Die Daten aus der Cloud können lokal auf einen Rechner gespeichert werden, als Backup - Möglichkeit, oder zur Weiterverarbeitung im csv - Format

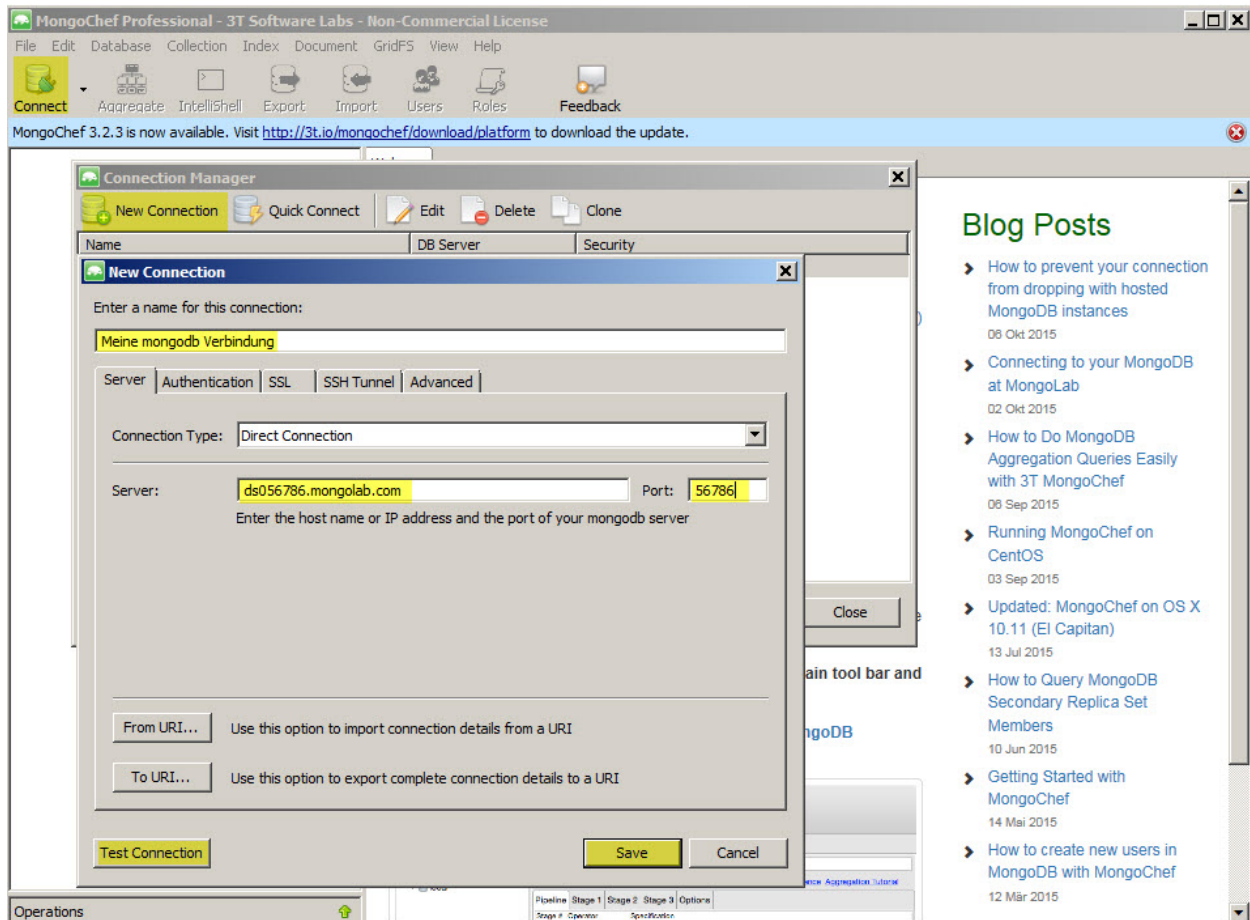
- **Erstellung von JSON Queries um BZ und andere Daten auszuwerten**

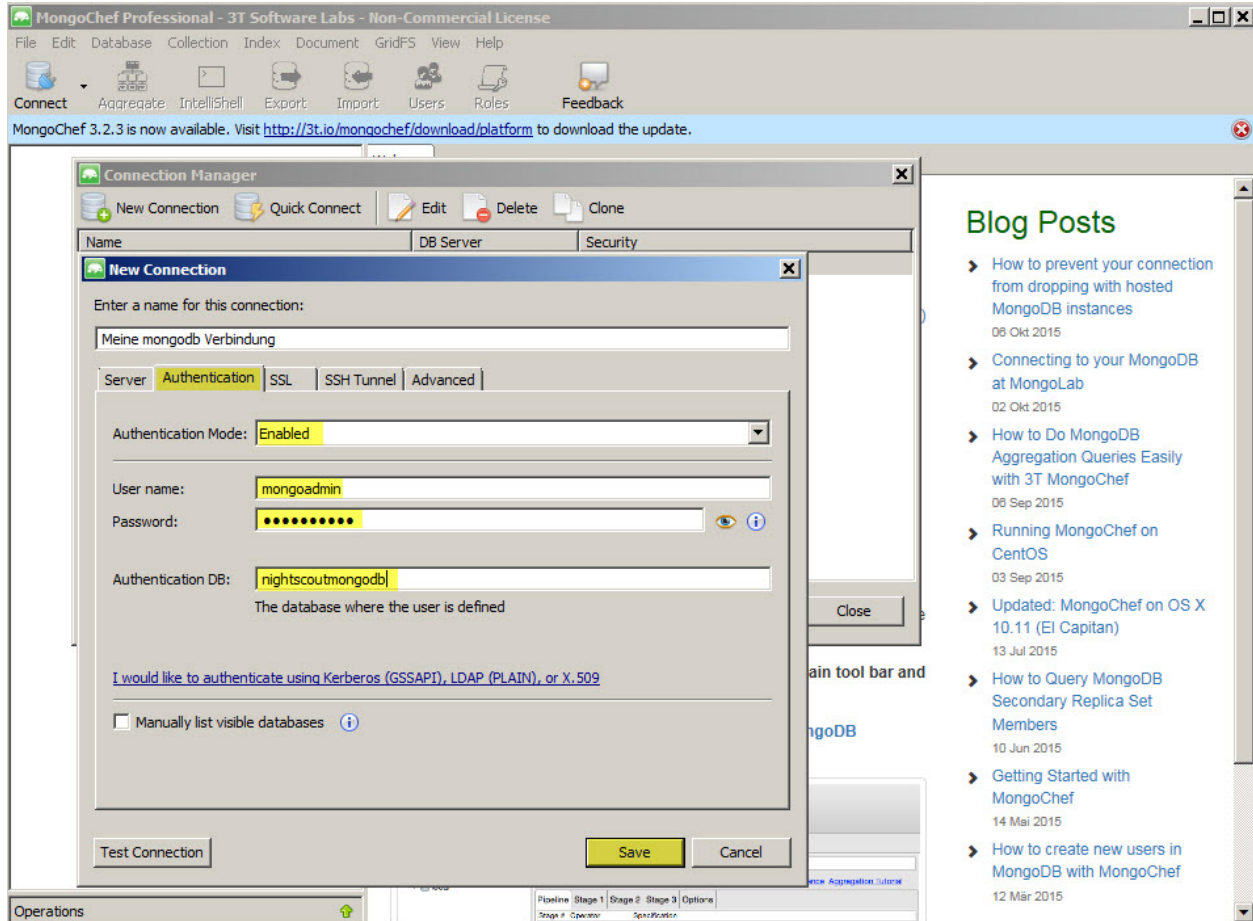
Für weitergehende Reporting Abfragen, welche nicht über die Nightscout - Reporting Seite zur Verfügung gestellt werden, können JSON - queries gebildet werden. Die Ergebnisse können wiederum im json- oder csv Format abgespeichert werden, um sie z.B. dem Doktor zukommen zu lassen.

Die **Installation** z.B. unter Windows ist eine Standard - Installation ohne speziell benötigte Parameter.

Beim Aufruf wird man aufgefordert, eine **Connection** = Verbindung zur Mongo DB zu konfigurieren. Dieses ist die URI aus vorhergehendem Kapitel.

Connection: Klicke **Connect**, dann öffnet sich folgendes Fenster:





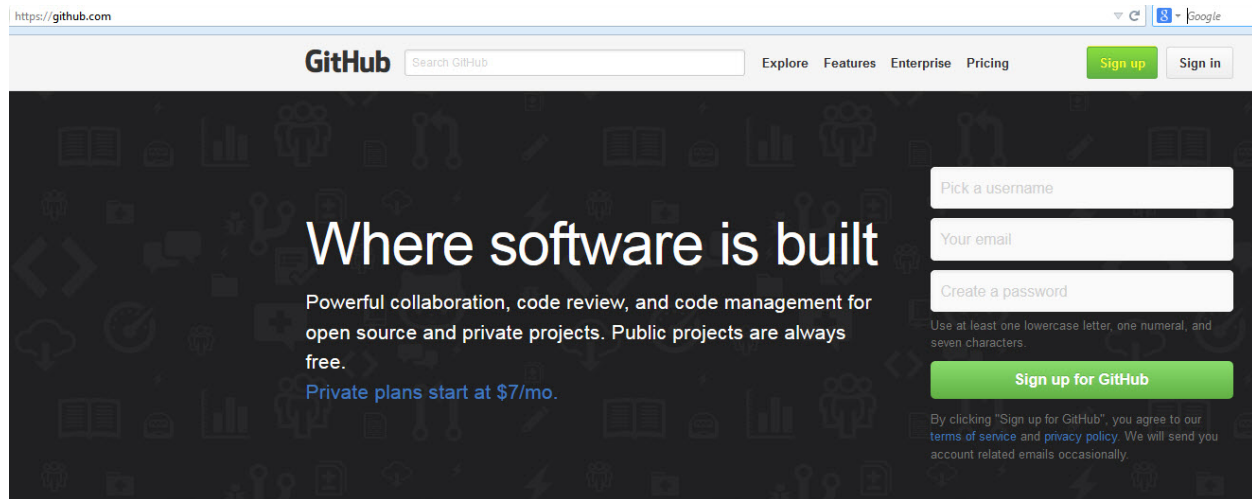
2.3 GitHub

GitHub ist ein webbasierter Filehosting-Dienst für Software-Entwicklungsprojekte. Software besteht aus vielen Files, die in einer Verzeichnisstruktur liegen. Diese Files enthalten den eigentlichen Programmiercode. Das `cgm-remote-monitor` repository enthält den Programmcode zum Aufbau der Nightscout Website.

Stellt ein Software - Entwickler etwas zur Verfügung, legt er die files in einem **branch** ab. Der **branch** kennzeichnet Umgebungen. Im **dev - branch** wird normalerweise entwickelt und getestet. Verlieft alles erfolgreich, werden die geänderten files in den **master branch** verschoben (committed). Nightscout wird weiterentwickelt, mit der Einteilung in **branches** lassen sich verschiedene Versionen einer Software managen.

Die Files eines branches kopieren wir uns in unser eigenes Verzeichnis (Fork).

Wir starten mit dem **Sign Up** auf der [GitHub Site](#):



Wichtig ist auch hier, Benutzername und Passwort in das Datenblatt einzutragen.

Mit **Edit Profile** können wir uns unsere Einstellungen ansehen.

Billing:

Unter diesem Menüpunkt sind die Kosten aufgelistet, wichtig ist hier, das hier **Free** angezeigt wird:

The image shows the GitHub Billing overview page. On the left, there's a sidebar with 'Personal settings' and various sub-links. The 'Billing' link is highlighted. The main content area is titled 'Billing overview'. It shows the current plan as 'Free' with '0 private repositories'. There are buttons for 'Change plan' and 'Purchase more'. Below this, it shows 'Git LFS Data' as '\$0 per month for 0 data packs'. It also displays the 'Billing cycle' as '30 days left, billed monthly'. There are progress bars for 'Bandwidth' (Using 0.0 of 1 GB/month) and 'Storage' (Using 0.0 of 1 GB). At the bottom, it shows 'Payment' as 'No payment method on file.' with a button to 'Add payment method'. There's also a 'Coupon' section stating 'You don't have an active coupon.' with a button to 'Redeem a coupon'. A 'Payment history' section at the bottom states 'You have not made any payments.' and 'Amounts shown in USD'.

Wir suchen jetzt den **Programmcode**:

Über die Suchfunktion jetzt `cgm-remote-monitor` eingeben:

Search

cgm-remote-monitor

Search

Repositories39

Code130

Issues345

Users

We've found 39 repository results

Sort: Best match

nightscout/cgm-remote-monitor

JavaScript ★ 145 3,948

nightscout web monitor

Updated 23 hours ago

Den **branch** wie master oder dev auswählen und mit **Fork** kopieren wir den Code in unser eigenes Verzeichnis:

nightscout / cgm-remote-monitor

Watch77

Star145

Fork3,953

Code

Issues71

Pull requests8

Wiki

Pulse

Graphs

nightscout web monitor

2,639 commits

8 branches

23 releases

40 contributors

Branch: master

New pull request

New file

Find file

HTTPS

https://github.com/nightscout/cgm-remote-monitor

Download ZIP

Das Ergebnis mit unserem eigenen Branch:

nightscout / cgm-remote-monitor

forked from nightscout/cgm-remote-monitor

Unwatch1

Star0

Fork3,981

Code

Pull requests0

Wiki

Pulse

Graphs

Settings

nightscout web monitor — Edit

2,644 commits

12 branches

18 releases

41 contributors

Branch: master

New pull request

New file

Find file

HTTPS

https://github.com/baembe/cgm-remote-monitor

Download ZIP

This branch is 5 commits ahead of nightscout:master.

Pull request

Compare

Merge branch 'master' into master

Latest commit e7922a6 2 days ago

bin

really basic template to post sgvs

a year ago

bundle

admin tools initial release

4 months ago

lib

use the d3 require trick again

a month ago

static

version bump for 0.8.3 release

a month ago

testing

replace the all the scattered time constants with a little module

5 months ago

tests

temporary removed client.test.js

a month ago

Aktualisierung des eigenen cgm-remote-monitor repositories

In den vergangenen Monaten sind Beta Versionen (0.9-beta) erschienen. Für die Versionen gibt es ein Web - Update-tool.

Die Links hierzu werden über Facebook (CGM in the Cloud) und Gitterchatgruppen veröffentlicht. Es empfiehlt sich eine separate Website anzulegen, um Beta - Versionen testen zu können. Durch Angabe des mongodb - Namens wird die Datenquelle mitgenommen.

2.3. GitHub

81

Beim Aufruf des Update-Links geben wir den GitHub Account - Namen ein und Klicken **I'm Ready**. Anschließend werden wir nach dem **github** Benutzernamen gefragt. Mit **Check for Updates** geht es weiter. Liegt eine neuere Version vor, dann erscheint der Button

Create Pull Request jetzt erfolgt der Vergleich mit den file - Versionen und es wird angezeigt, was aktualisiert wurde.

Mit **Confirm Merge** bestätigen wir die ermittelten Änderungen und bringen damit den cgm-remote-monitor auf den aktuellen Stand.

Jetzt installieren wir die Nightscout Website [Deploy to Azure](#).

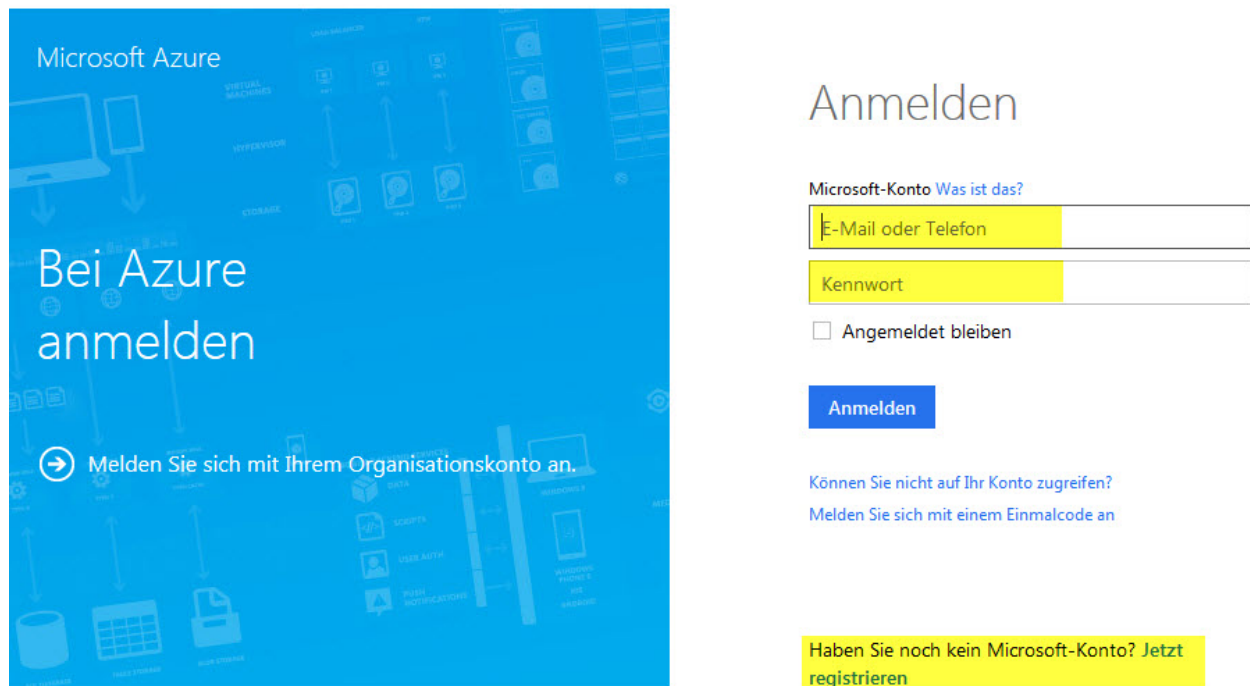
2.4 Azure

[Azure](#) ist ein Cloud - Computing Dienst, welcher umfangreiche Services für Anwender zur Verfügung stellt.

2.4.1 Einstieg

Den Einstieg in deutscher Sprache findet man hier: [Azure](#)

Mit dem zweimaligen Klicken des Buttons **Kostenlos testen** gelangen wir zur Registrierungsseite:



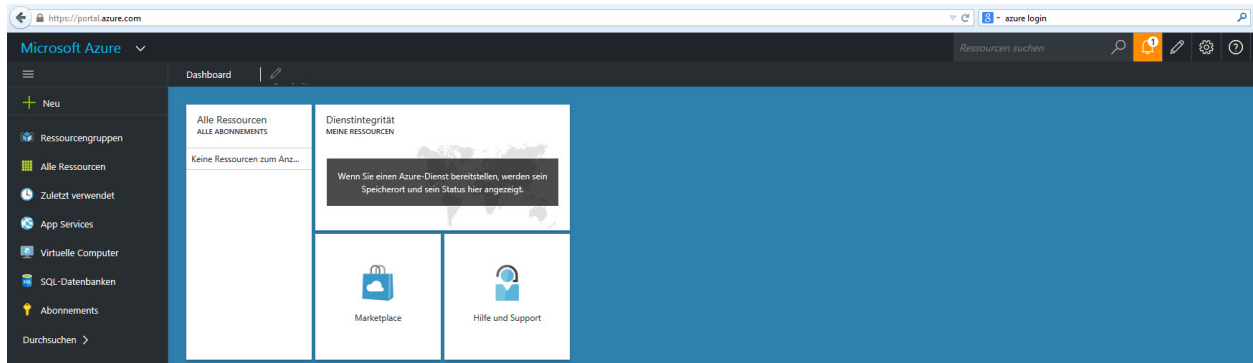
Sollte man schon ein Konto haben, kann man sich einfach anmelden, ansonsten geht es mit der **Registrierung** weiter.

Wir werden aufgefordert, die Kreditkartendaten einzugeben, die Karte wird nicht belastet.

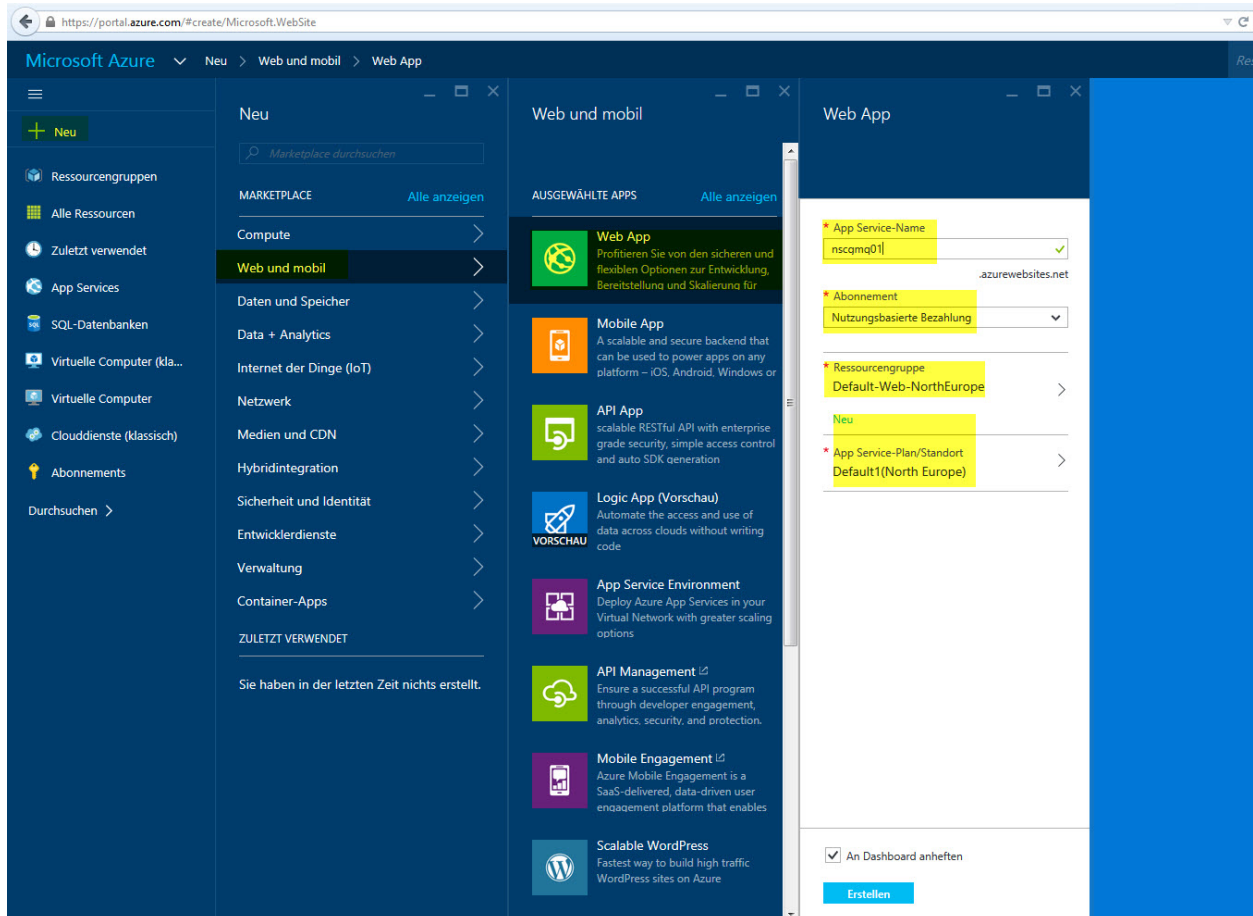
Nachdem alles abgeschlossen und gespeichert wurde, können wir uns im Azure Portal anmelden.

Über den Klick auf **Portal** befinden wir uns jetzt im Azure Portal und können die Web - Applikation konfigurieren.

Azure Portal Startseite:



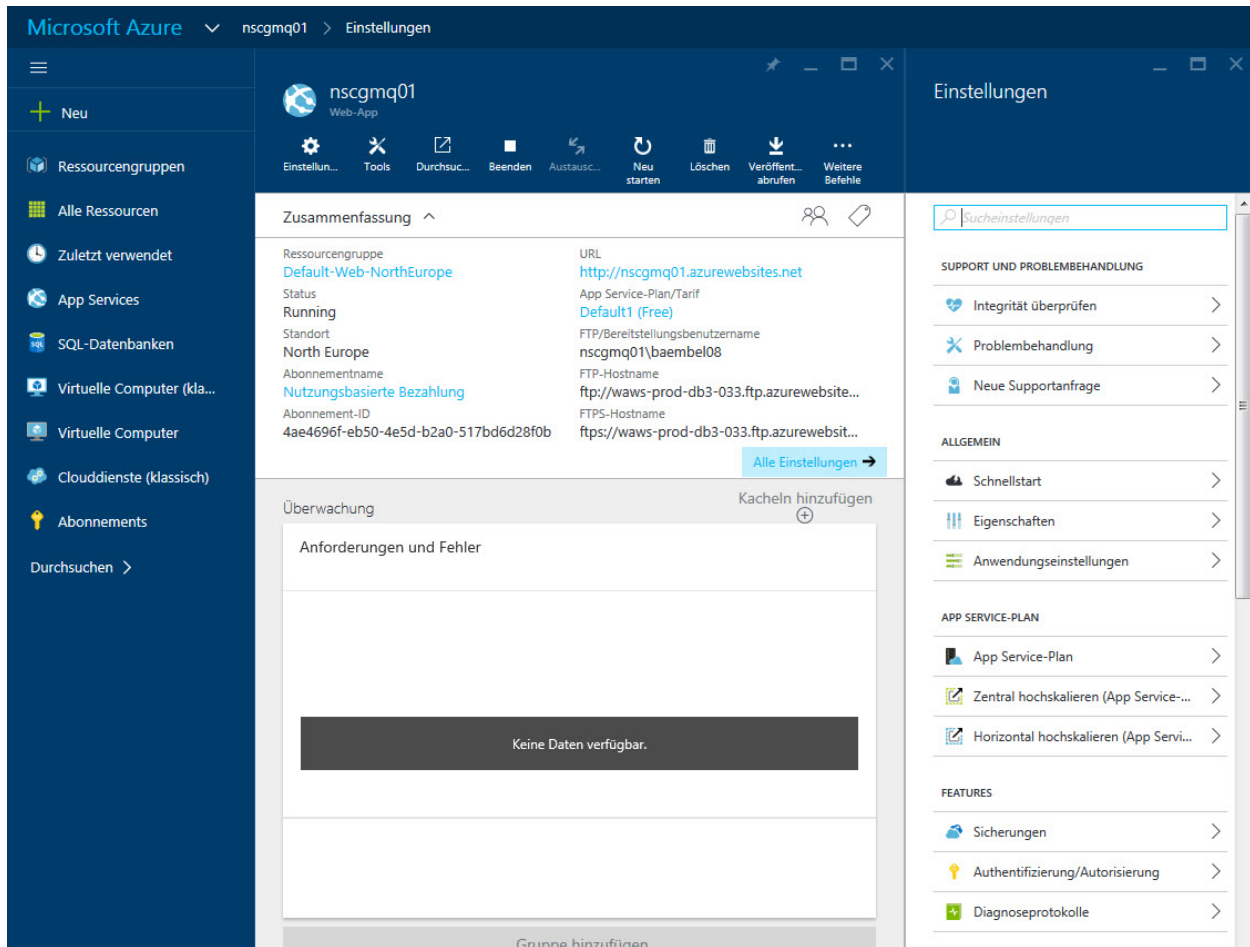
Mit dem Button “+ Neu” richten wir eine neue **Resource** ein. Die Menüführung ist kaskadiert, alle gelb markierten Punkte sind auszuwählen. Bei Auswahlpunkten mit > kann man weitere Einstellungen auswählen:



Mit dem Button **Erstellen** speichern wir die Einstellungen. Dieses kann einige Zeit in Anspruch nehmen.

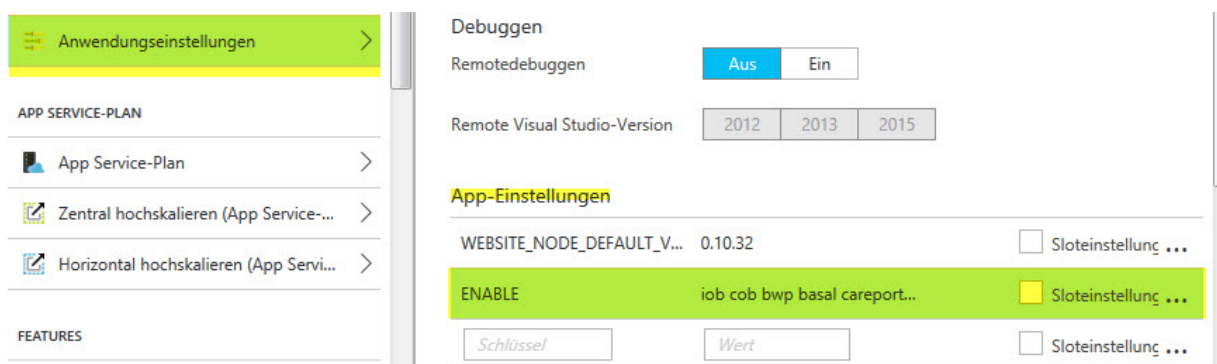
Der Name der Nightscout - Website ist jetzt .azurewebsites.net, hier im Beispiel: **nscgmq01.azurewebsites.net**. Diesen Website - Namen bitte auch notieren.

Wenn die Web-App erstellt wurde, erscheint diese Ansicht:



Die Web-App wird jetzt in Nordeuropa/Irland gehostet. Wichtig ist der **Status**. Hier wird angezeigt, ob die App **Running** oder **Stopped** ist.

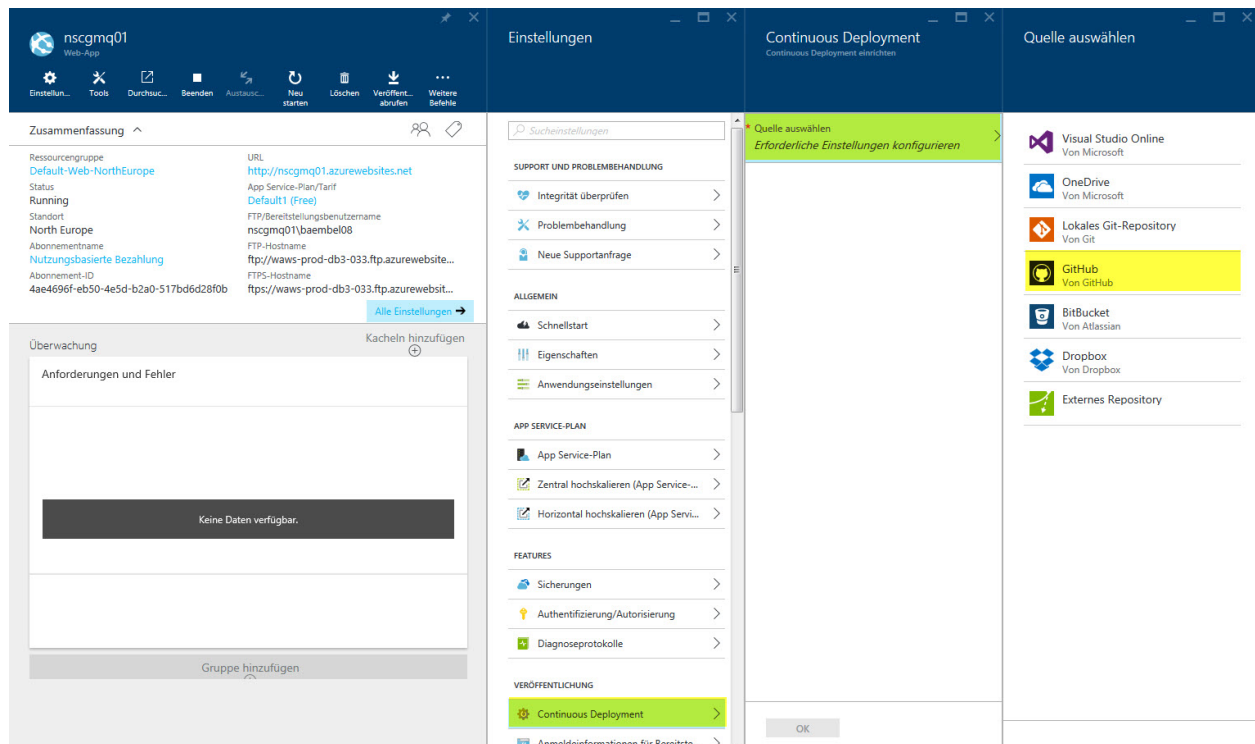
Weiter geht es mit der Konfiguration der **Plugin** Module. Diese sind in Kapitel [Konfiguration der Nightscout Plugins](#)



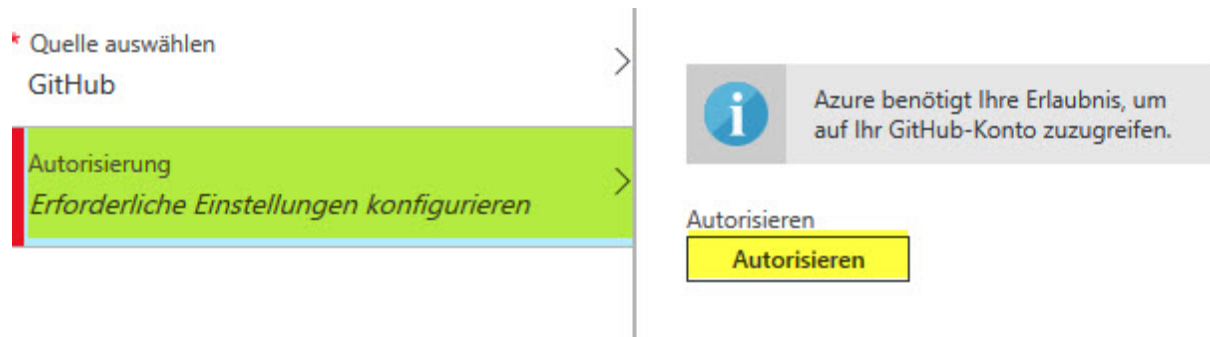
2.4.2 Einbindung des GitHub Source Codes

Zu Bereitstellung der Nightscout-Website fehlt uns noch der `cgm-remote-monitor`, den wir über GitHub laden.

Wir steigen hier ein, um die Verbindung herzustellen:



Jetzt folgt die **GitHub Autorisierung**:



Nachdem man gitHub Benutzernamen und Passwort eingegeben hat, erscheint diese Meldung:

Authorize application

Azure Management Portal by @Azure would like permission to access your account



Review permissions



Repositories

Public and private



Authorize application

Azure Management Portal
No description
Visit application's website
 ⓘ Learn more about OAuth

Mit **OK** bestätigen.

Jetzt wählt man den `cgm-remote-monitor` aus:

* Quelle auswählen GitHub	>	nightscout_handbuch
* Autorisierung [REDACTED]	>	cqm-remote-monitor
Projekt auswählen <i>Erforderliche Einstellungen konfigurieren</i>	>	
Verzweigung auswählen <i>Erforderliche Einstellungen konfigurieren</i>		

Mit **OK** bestätigen. Die Web-app wird bereitgestellt. Dieses kann ein wenig dauern.

Am Ende erscheint in der Zusammenfassung die Einbindung von GitHub hier:

Sollte es zu einer Fehlermeldung kommen, bitte die Bereitstellung noch einmal starten.

Die Nightscout Webseite steht jetzt im Internet zur Verfügung und kann über einen Browser mit dem Aufruf <http://YOURWEBSITENAME.azurewebsites.net> aufgerufen werden.

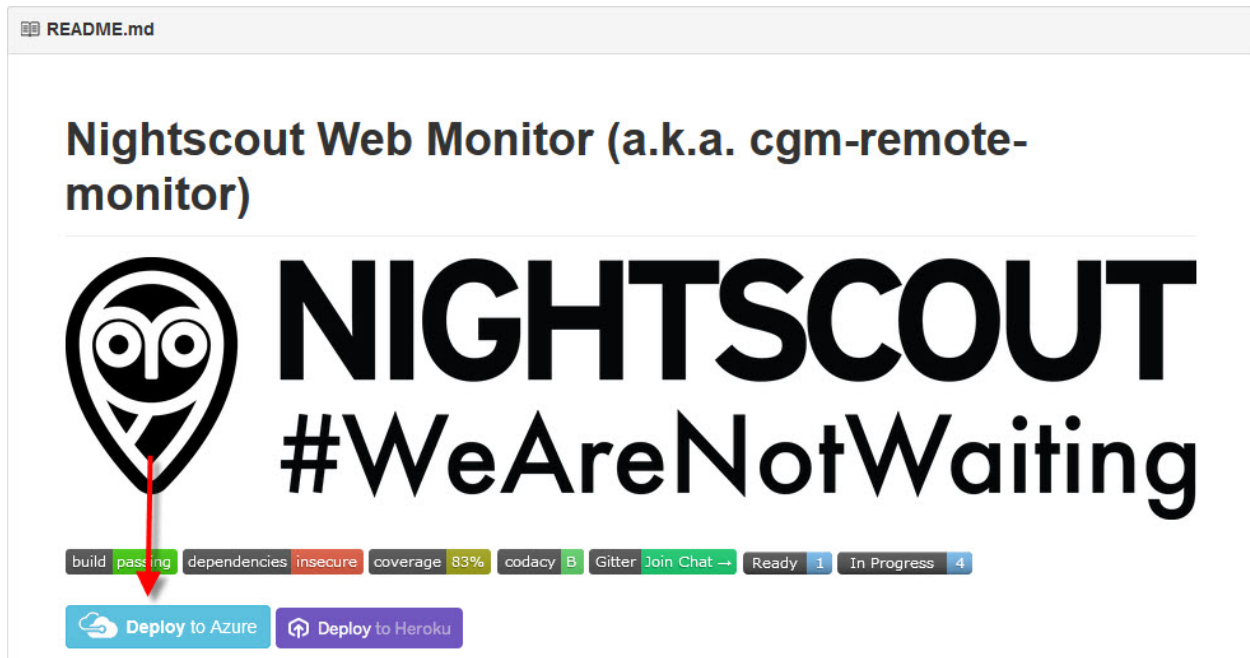
Die Einrichtung ist fertig.

2.4.3 Kosten

Web-Applikationen sind grundsätzlich **kostenfrei**. Auf dem Azure - Portal gibt es eine **Preisübersicht**:

Wir Starten mit der Anmeldung in in Github an, navigieren in unserer Repository und den Branch, welchen wir installieren wollen.

Weiter Klicken auf das **Deploy to Azure** Symbol:



Wir loggen uns bei Azure ein (folgt den angezeigten Anweisungen, um eingeloggt zu werden). (Wenn wir noch kein Azure-Konto haben, sollten wir das Azure-Konto zunächst separat hinzufügen und dann wieder in diese Sequenz von Schritten zurückkehren.)

Es ist möglich, das Konto an dieser Stelle einzurichten, aber es kann weniger verwirrend sein, wenn wir das Konto vorher erstellen.)

Der Bildschirm **Deploy to Azure** wird mit allen Optionen geladen, die wir (am Anfang) für unsere Website konfigurieren können. (Wir können dieses später bei Bedarf ändern.)

Es erscheint diese Web - Maske:

Deploy to Azure

1
2
3

SETUP PREVIEW DEPLOY

Repository Url - <https://github.com/nightscout/cgm-remote-monitor>
 Branch - dev

Directory Default Directory ▾ Resource Group Create New ▾ Site Name - Name is available cgm-remote-monitorb27d Site Location South Central US ▾ Skus Free ▾ Display Units mg/dl ▾ Theme colors ▾ Language en ▾ Alarm_high off ▾ Alarm_timeago_urgent off ▾ Alarm_urgent_high off ▾ Basal_render none ▾ Enable basal bwp cage careportal cob rawbg sage iage treatment ▾ Show_plugins careportal ▾ Devicestatus_advanced false ▾	Subscription Pay-As-You-Go ▾ Resource Group Name cgm-remote-monitorb27d Mongo Connection Api Secret Enter an API secret. Must be at least 12 characters Time_format 24 ▾ Custom_title Nightscout Alarm_low off ▾ Alarm_timeago_warn off ▾ Alarm_urgent_low off ▾ Scale_y log ▾ Night_mode off ▾ Show_rawbg never ▾ Profile_multiple off ▾
--	---

Next ➞

Wir stellen sicher, dass das Abonnement Pay-as-you-go zeigt. (Wenn das Konto neu angelegt ist, kann es stattdessen die kostenlose Testversion zeigen, und wir müssen dieses ändern, bevor der Testphase endet.)

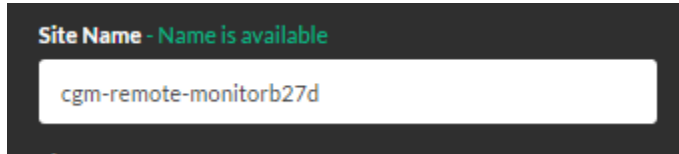
Wenn eine Ressourcengruppe im Dropdown-Menü aufgelistet ist, wählen wir sie aus. Ansonsten wählen wir Create New und akzeptieren oder ersetzen den “neuen” Ressourcengruppennamen im Feld Resource Group Name.

Wir wählen unseren Site-Namen. Ein Standardname wird bereits im Feld angezeigt. Wir können ihn mit dem gewünschten Namen ersetzen. Was wir hier eingeben, wird der “Name” unserer Nightscout-Website (und wird der Schlüs-

selteil Ihrer Nightscout-URL) sein. Der Name unserer Website kann nicht geändert werden. Wenn wir später einen anderen Namen verwenden möchten, müssen wir eine neue Website erstellen.

Api Secret: API Secret Wert, bleibt der Wert unter 12 Zeichen, bricht die Installation ab.

Language: Standart Wert ist **en**, bitte in **de** wechseln.



Wir wählen im Dropdown-Menü den Standort aus. (Dieser Standort muss nicht mit unserem tatsächlichen geografischen Standort übereinstimmen. North Europe ist für Deutschland zu empfehlen.)

Wir stellen sicher, dass **Free** im Sku-Dropdown-Menü ausgewählt ist. (Dies ist die “kostenlose Variante” von Azure.)

Wir geben jetzt die mongo (mLab) URI in das Feld **Mongo-Connection** ein.

Die mLab URI verwendet die Syntax, die in diesem Beispiel gezeigt wird, wir müssen unsere “EIGEN-Informationen” ausfüllen, “nicht” die Beispielwerte, die hier gezeigt werden:

```
Mongodb://databaseUserName:databaseUserPassword@ds012345.mlab.com:12345/  
databaseName
```

Beispielsweise:

```
Mongodb://sallyuser:sallypassword@ds012345.mlab.com:12345/mycgm
```

Wir wählen **mg / dl** oder **mmol** aus dem Dropdown-Menü **Display Units**.

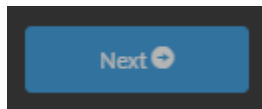
In das **Api Secret**-Feld geben wir das minimale **API-SECRET** mit “mindestens 12 Zeichen” ein. Zeichen sind frei wählbar, aber es muss mindestens 12 Zeichen lang sein.

Viele der anderen Variablen auf dem Bereitstellungsbildschirm sind optional. Wenn wir nicht wissen, was eine Variable bedeutet, können wir im Kapitel **Azure Plugins** Hilfe finden.

Die **ENABLE**-Variable wird auf dieser einstufigen Implementierung mit vielen häufig verwendeten Features vorbelegt. Wenn wir der Werte-Zeichenfolge etwas hinzufügen möchten, setzen wir ein Leerzeichen nach dem letzten Zeichen im String und geben einen neuen Wert ein (z. B. Bridge für Benutzer von Dexcom G4 Share / G5-Bridge-Sites oder Pushover für Pushover-Benachrichtigungen).

Im Feld **Show_plugins** wird “careportal” vorgefüllt. In diesem Feld aufgelistete Plugins sind immer auf der Website sichtbar, nn sie müssten ansonsten innerhalb des Einstellungsbereichs des einzelnen Browsers aktiviert werden. Wenn wir möchten, dass **iage**, **cage**, **sage**, **bwp** und **rawbg** “immer sichtbar” sind, dann listen wir sie in diesem Feld in einer durch Leerzeichen getrennten Liste auf. (Hier können auch andere Werte eingegeben werden, ua **bwp**, **boluscalc** usw.)

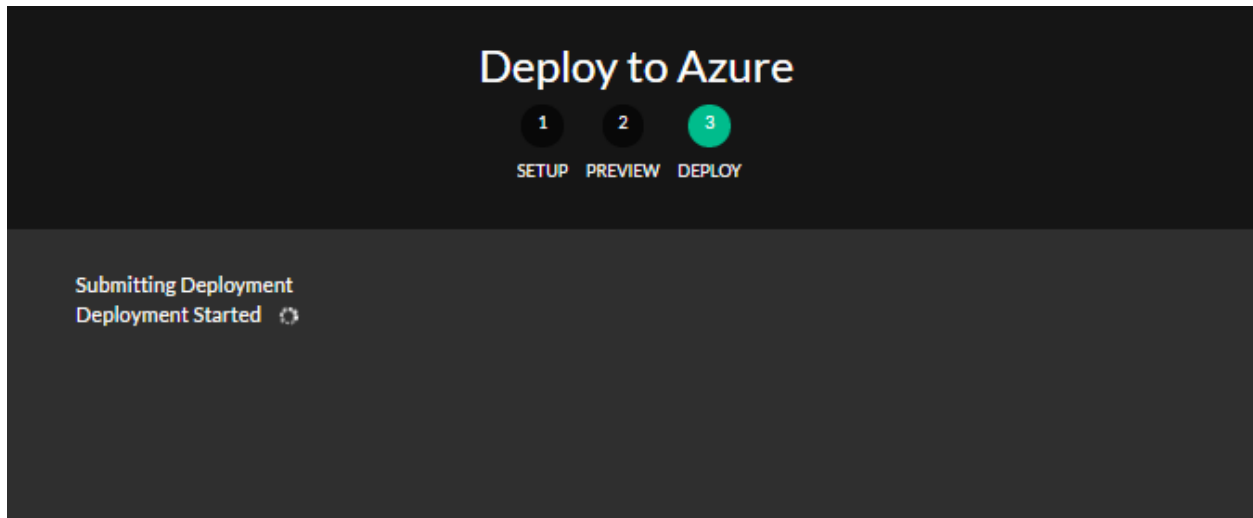
Wir klicken auf die Schaltfläche **Next**:



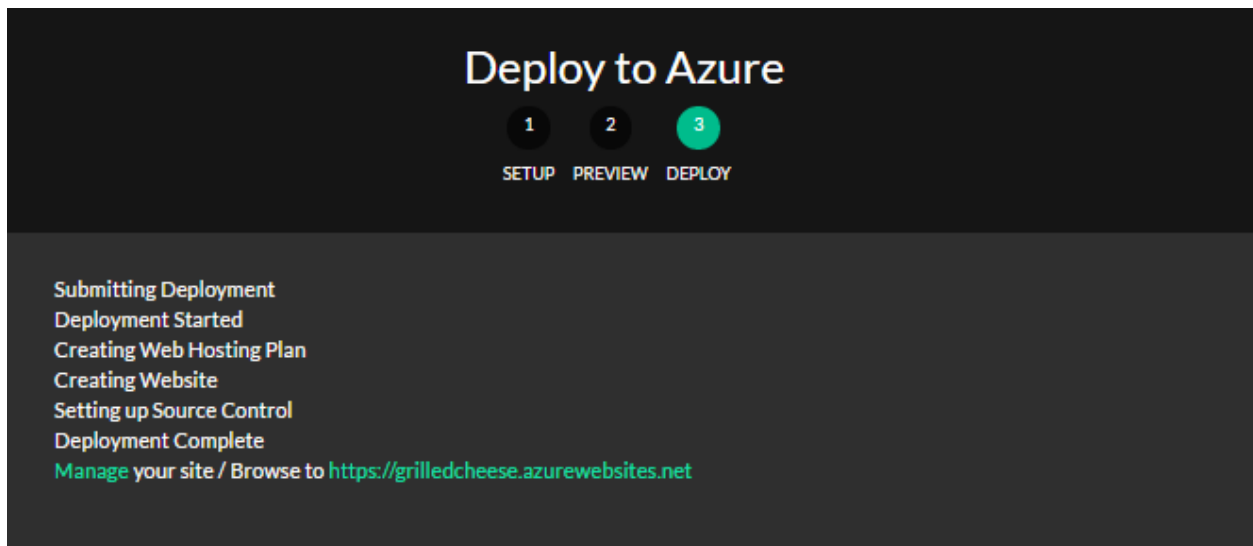
Der nächste Bildschirm bestätigt, dass eine “Website” bereitgestellt wird. Wir klicken Sie auf **Deploy**. (Wenn wir zum vorherigen Bildschirm zurückkehren müssen, klicken wir auf “Setup”).



Beim nächsten Schritt sehen wir den Fortschritt. **Sei geduldig** :).



Wenn die Bereitstellung abgeschlossen ist (und erfolgreich), sehen wir eine abschließende Zeile, die über Links verfügt, die uns erlauben, unsere Website zu verwalten oder nach `yoursiteURL` zu browsen. Unsere Nightscout-URL verwendet dieses Format: <http://YOURWEBSITENAME.azurewebsites.net>.



Wir Klicken auf unsere Website-URL, um die Website zu öffnen. Wenn die Bereitstellung erfolgreich war und unsere **Mlab-Informationen** korrekt eingegeben wurden, sollten wir eine schwarze Nightscout-Website sehen. (Wenn unsere Website noch keine Daten enthält, wird es eine schwarze Website mit Zeit und Bindestrichen oder Zeit und das Wort **undefiniert**.

Wenn wir unsere Uploader-App noch nicht installiert und konfiguriert haben, können wir dieses jetzt tun. (Wenn wir ein kabelgebundenes Rig verwenden, bedeutet dies, dass wir die Nightscout App installieren müssen. Wenn wir einen anderen Uploader verwenden oder die Bridge konfigurieren, kehren wir zu den Grundeinstellungen für unser spezielles Setup zurück und fahren mit dem nächsten Schritt fort.) Wir werden keine Daten sehen Auf der Nightscout-Website, bis die Uploader-App (oder Bridge) konfiguriert ist.

Wir bestätigen, das unsere Website ist als kostenlose Website konfiguriert ist.

Wenn unser Azure Konto neu ist und wir während des Setups keine Pay-As-You-Go gewählt haben, haben wir 30 Tage Zeit, die Site in Pay-As-You-Go zu konvertieren. Wir empfehlen, die Website-Einstellungen zu überprüfen, indem wir Folgendes überprüfen:

Bitte Bestätigen, dass der Abonnementtyp auf Pay-**as**-you-go festgelegt ist.
Bitte Bestätigen, dass das Pricing Tier auf Free festgelegt ist.

Essentials ^

Resource group
MyResourceGroup

Status
Running

Location
North Central US

Subscription name
Pay-As-You-Go

Subscription ID
77-15511-001-400-000-54511-76101

URL
[Redacted]

App Service plan/pricing tier
MyPlan (Free)

FTP/Deployment username
[Redacted]

FTP hostname
[Redacted]

FTPS hostname
ftps://auaws-prod-ch1-000-ftp.azurewebsites.net

All settings →

2.6 Nightscout Plugins

Wir befassen uns hier mit der Beschreibung der Programm - Module, den sogenannten **Plugins**. Die Nightscout Website besteht aus vielen kleinen **Plugins**. Diese Plugins können konfiguriert werden. Die Konfigurationseinstellungen können wir in den **Azure** Anwendungseinstellungen speichern.

2.6.1 Pflichtwerte

- MONGO_CONNECTION - Deine Mongo uri, zum Beispiel: `mongodb://sally:sallypass@ds099999.mongolab.com:99999/nightscout`
- DISPLAY_UNITS (mg / dl) - Auswahl: mg / dl und mmol. Die Einstellung auf mmol setzt den gesamten Server auf mmol Modus als Standard, weitere Einstellungen sind nicht erforderlich.
- BASE_URL - Wird verwendet für den Aufbau von Verbindungen zu deinem Websiten api, dh pushover callbacks, in der Regel die URL deiner Nightscout Website, statt http möchtest du vielleicht lieber https verwenden

2.6.2 Eigenschaften

- ENABLE - Wird verwendet, um optionale Funktionen zu aktivieren, erwartet eine durch Leerzeichen getrennte Liste, wie zum Beispiel: `careportal rawbg iob`
- DISABLE wird genutzt, um Standardfunktionen zu deaktivieren, erwartet eine durch Leerzeichen getrennte Liste
- API_SECRET - Ein geheimes Passwort zur Authentifizierung. Es muss mindestens **12 Zeichen** lang sein und ist frei wählbar. Das API_SECRET wird für Careportal Eingaben, Pushover Notifications und REST API Upload benötigt.
- AUTH_DEFAULT_ROLES (readable) - mögliche Werte readable, denied, oder ein beliebiger Rollenname. Wenn **lesend** gesetzt, kann jeder ohne einen Zugriffstoken auf die Website zugreifen. Die Einstellung denied bewirkt, dass man sich bei jedem Besuch der Website authentifizieren muss, die Einstellung

`status-only` bewirkt ein Anmelden über das **API Secret**. Wenn auf den Wert `off` gesetzt ist, wird die Careportal Rolle zu `AUTH_DEFAULT_ROLES` hinzugefügt.

- `TREATMENTS_AUTH` (`off`) - mögliche Werte `on` oder `off`. Bei Einstellung auf `on` muss man sich bei Eingaben auf dem jeweiligen Gerät mit dem `API_SECRET` authentisieren.

2.6.3 Alarmer

Diese Alarmerinstellungen wirken sich auf alle Benachrichtigungsmethoden aus (Browser, Pushover, IFTTT Maker, etc.), einige Einstellungen können durch Browsereinstellungen außer Kraft gesetzt werden.

- `ALARM_TYPES` (`simple` falls `BG_* ENV` gesetzt sind, sonst `predict`) - derzeit werden zwei Alarmtypen unterstützt und können unabhängig voneinander oder in Kombination verwendet werden. Der `simple` Alarmtyp vergleicht nur die aktuelle BG mit `BG_` Schwellen, der `predict` Alarmtyp verwendet eine optimierte Formel, die die BG aufgrund ihres bisherigen Verlaufs prognostiziert. Derzeit verwendet `Predict` **NICHT** ****** eines der `BG_* ENV`.
- `BG_HIGH` (260) - muss auf mg / dl Einheiten eingestellt werden; das ist eine hohe BG außerhalb des Zielbereichs, die als dringlich erachtet wird
- `BG_TARGET_TOP` (180) - muss auf mg / dl Einheiten eingestellt werden; das obere Ende des Zielbereichs, wird auch verwendet, um die obere Linie in der Grafik zu zeichnen
- `BG_TARGET_BOTTOM` (80) - muss auf mg / dl Einheiten eingestellt werden; das untere Ende des Zielbereichs, wird als untere Linie in der Grafik verwendet
- `BG_LOW` (55) - muss auf mg / dl Einheiten eingestellt werden; eine niedrige BG außerhalb des Zielbereichs, die als dringlich erachtet wird
- `ALARM_URGENT_HIGH` (`on`) - mögliche Werte `on` oder `off`
- `ALARM_URGENT_HIGH_MINS` (30 60 90 120) - Anzahl der Minuten, in denen ein dringender Hochalarm schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet
- `ALARM_HIGH` (`on`) - mögliche Werte `on` oder `off`
- `ALARM_HIGH_MINS` (30 60 90 120) - Anzahl der Minuten, in denen ein (einfacher) Hochalarm schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet
- `ALARM_LOW` (`on`) - mögliche Werte `on` oder `off`
- `ALARM_LOW_MINS` (15 30 45 60) - Anzahl der Minuten, in denen ein (einfacher) Tiefalarm schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet
- `ALARM_URGENT_LOW` (`on`) - mögliche Werte `on` oder `off`
- `ALARM_URGENT_LOW_MINS` (15 30 45) - Anzahl der Minuten, in denen ein dringender Tiefalarm schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet
- `ALARM_URGENT_MINS` (30 60 90 120) - Anzahl der Minuten, in denen ein dringender Alarm (der nicht als hoch oder niedrig markiert ist) schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet
- `ALARM_WARN_MINS` (30 60 90 120) - Anzahl der Minuten, in denen ein Alarm (der nicht als hoch oder niedrig markiert ist) schlafen gelegt wird, bevor er erneut ertönt; die einzelnen Minuten werden durch Leerzeichen voneinander getrennt, die erste Zahl wird für pushover verwendet

2.6.4 Basiseinstellungen

Die Basiseinstellungen sind Standardwerte, die beim Implementieren einer Nightscout Website gesetzt werden.

- `MONGO_COLLECTION` (`entries`) - Diese Sammlung speichert SGV, MBG und CAL Aufzeichnungen von deinem CGM-Gerät
- `MONGO_TREATMENTS_COLLECTION` (`treatments`) - Diese Sammlung speichert sog. `treatments` (eingegebene Therapiemaßnahmen oder kurz: Eingaben), die im Care Portal eingegeben worden sind, s.o. unter `ENABLE` env var
- `MONGO_DEVICESTATUS_COLLECTION` (`devicestatus`) - Diese Sammlung speichert Gerätestatusinformationen wie z.B. die Uploader Batterie
- `MONGO_PROFILE_COLLECTION` (`Profile`) - Diese Sammlung speichert deine Profile
- `MONGO_FOOD_COLLECTION` (`food`) - Die Sammlung speichert deine Lebensmittel-Datenbank
- `PORT` (1337) - Der Port, auf den die node.js Anwendung zugreift
- `Hostname` - Der Hostname, den die node.js-Anwendung verwendet, standardmäßig null für jeden Hostnamen, für IPv6 evtl. `::` nicht vergessen.
- `SSL_KEY` - Pfad zu deiner SSL-Schlüsseldatei, so dass ssl (https) direkt in node.js aktiviert werden kann
- `SSL_CERT` - Pfad zu deiner SSL-Zertifikat-Datei, so dass ssl (https) direkt in node.js aktiviert werden kann
- `SSL_CA` - Pfad zur ssl CA-Datei, so dass ssl (https) direkt in node.js aktiviert werden kann
- `HEARTBEAT` (60) - Anzahl der Sekunden zwischen den einzelnen Datenbanküberprüfungen

2.6.5 Vordefinierte Werte für Browser-Einstellungen (optional)

Wir können **Browsers-einstellungen** zentral in den Azure App. Einstellungen speichern. Das hat den Vorteil, dass man egal von welchem Browser oder einer Smartphone App aus immer die gleichen Einstellungen hat. Machen wir dies nicht, werden die Einstellungen nur im Browser Cache gespeichert und müssen ggf. immer wieder neu eingegeben werden. Jeder Parameter mit Wert steht in einer extra Zeile.

- `TIME_FORMAT` (12) - mögliche Werte 12 oder 24
- `NIGHT_MODE` (off) - mögliche Werte on oder off
- `SHOW_RAWBG` (never) - mögliche Werte always, never oder noise
- `CUSTOM_TITLE` (Nightscout) - angepasste/r Titel/Überschrift
- `THEME` (default) - mögliche Werte default oder colors
- `ALARM_TIMEAGO_WARN` (on) - mögliche Werte on oder off
- `ALARM_TIMEAGO_WARN_MINS` (15) - Minuten seit dem letzten Messwert, um eine Warnung auszulösen
- `ALARM_TIMEAGO_URGENT` (on) - mögliche Werte on oder off
- `ALARM_TIMEAGO_URGENT_MINS` (30) - Minuten seit dem letzten Wert, um einen dringenden Alarm auszulösen
- `SHOW_PLUGINS` - Anzeige der Plugins in der Website, Standardwerte für alle freigegeben
- `SHOW_FORECAST` (ar2) - Plugin Prognosen, die standardmäßig angezeigt werden sollen, unterstützt räumlich begrenzte Werte wie " ar2 openaps "
- `LANGUAGE` (de) - Sprache der Website. Falls nicht verfügbar, wird Englisch verwendet,
- `SCALE_Y` (log) - Die Art der Skalierung , welche für die Y-Achse des Charts System verwendet wird.

- Die Standard `log` (logarithmisch) Option lässt mehr Details im unteren Bereich zu , während immer noch der volle CGM Bereich angezeigt wird.
- Die `linear` Option hat im gleichen Abstand Markierungen , der verwendete Bereich ist dynamisch, so dass kein Platz an der Spitze des Diagramms verschwendet wird.
- Die `log-dynamic` ist der Standard `log` Option ähnlich und verwendet den gleichen dynamischen Bereich, aber die `linear` Skala.
- `EDIT_MODE` (`on`) - mögliche Werte `on` oder `off`. Aktivieren oder deaktivieren des Symbols, das den Bearbeitungsmodus der treatments (Behandlungsmaßnahmen) ermöglicht.

Nach Eingabe der Daten das **Speichern** nicht vergessen.

Die Einstellungen sehen dann z.B. so aus:

App-Einstellungen

WEBSITE_NODE_DEFAULT_VERSION	0.10.32
ENABLE	iob cob bwp basal careportal maker treatmentnotify
TREATMENT_AUTH	ON
TIME_FORMAT	24
NIGHT_MODE	off
SHOW_RAWBG	never
THEME	colors
ALARM_URGENT_HIGH	off
ALARM_HIGH	off
ALARM_LOW	off
ALARM_URGENT_LOW	off
ALARM_TIMEAGO_WARN	off
ALARM_TIMEAGO_URGENT	off
LANGUAGE	de
SHOW_PLUGINS	iob cob bwp basal
BASAL_RENDER	icicle

2.6.6 Plugins

Plugins werden verwendet, um die Art und Weise Informationen anzuzeigen, wie Benachrichtigungen gesendet werden, Alarmer ausgelöst werden, und vieles mehr.

Die eingebauten / Beispiel-Plugins, die standardmäßig verfügbar sind, sind nachfolgend aufgelistet. Die Plugins müssen möglicherweise noch aktiviert werden, indem sie der Umgebungsvariable `ENABLE` hinzugefügt werden.

2.6.6.1 Standard Plugins

Diese können durch Einstellen der `DISABLE` env var deaktiviert werden, zum Beispieldisable="Direction upbat "

2.6.6.1.1 Delta (BG Delta)

Berechnet und zeigt die Veränderung zwischen den letzten 2 BG-Werten.

2.6.6.1.2 Direction (BG Direction)

Zeigt die Trendrichtung an.

2.6.6.1.3 Upbat (Uploader Batterie)

Zeigt den aktuellsten Batteriestatus vom Uploader Telefon.

2.6.6.1.4 Timeago (Time Ago)

Zeigt die Zeit seit dem letzten CGM-Eintrag an. Unterstützt die `TIMEAGO_ENABLE_ALERTS` [Erweiterte Einstellung] (# extended-Einstellungen) für die Serverseite bei Alarmen wegen veralteter Daten via Pushover und IFTTT.

2.6.6.1.5 Devicestatus (Gerätestatus)

Wird von `upbat` und anderen Plugins verwendet, um Gerätestatusinformationen anzuzeigen. Unterstützt die `DEVICESTATUS_ADVANCED = " true "` [Erweiterte Einstellung] (# extended-Einstellungen), um jeden Gerätestatus an den Client für die retrospektive Anwendung zu senden und um andere Plugins zu unterstützen.

2.6.6.1.6 Errorcodes (CGM-Fehlercodes)

Erzeugt Alarme für CGM-Codes 9 (Sanduhr) und 10 (???).

- Verwende [erweiterte Einstellungen] (# extended-Einstellungen) um zu sehen, welche Fehlercodes Benachrichtigungen und Alarme auslösen:
 - `ERRORCODES_INFO` (1 2 3 4 5 6 7 8) - Standardmäßig erzeugen die “Needs Calibration” (Blutstropfen) und andere Fehlercodes unter 9 eine Info-Ebene Benachrichtigung; die Auswahlmöglichkeiten sind eine durch Leerzeichen getrennte Zahlenreihe oder `off`, um diese Benachrichtigung auszuschalten
 - `ERRORCODES_WARN` (`off`) - Standardmäßig ist keine Warnung konfiguriert, die Auswahlmöglichkeiten sind eine durch Leerzeichen getrennte Zahlenreihe oder `off`, um diese Benachrichtigung auszuschalten
 - `ERRORCODES_URGENT` (9 10) - standardmäßig generieren die Sanduhr und ??? einen dringenden Alarm, die Auswahlmöglichkeiten sind eine durch Leerzeichen getrennte Zahlenreihe oder `off`, um diese Benachrichtigung auszuschalten

2.6.6.1.7 Ar2 (AR2 Forecasting)

Erzeugt Alarme auf Basis von prognostizierten Werten. Siehe [Forecasting mit AR2 Algorithmus] (<https://github.com/nightscout/nightscout.github.io/wiki/Forecasting>)

- Standardmäßig aktiviert, wenn keine Schwellenwerte festgelegt werden `** OR ** ALARM_TYPES umfasst predict`.
- Verwende [erweiterte Einstellungen] (# extended-Einstellungen), um das AR2 Verhalten anzupassen:

- `AR2_USE_RAW (false)` - um für Prognosen `rawbg` Werte zu verwenden, wenn Standardwerte keinen Alarm auslösen
- `AR2_CONE_FACTOR ( '2' )` - um die Größe der Kegel anzupassen, benutze '0' für eine einzelne Zeile

2.6.6.1.8 Simplealarms (Simple BG Alarme)

Verwendet `BG_HIGH`, `BG_TARGET_TOP`, `BG_TARGET_BOTTOM`, `BG_LOW` -Schwellenwerte, um Alarme abzugeben.

Ist standardmäßig aktiviert, wenn einer dieser Schwellenwerte gesetzt ist `** OR ** ALARM_TYPES` umfasst `simple`.

2.6.6.2 profile (Treatment Profile)

Fügen Sie Link zum Profil-Editor hinzu, und geben Sie die Einstellungen für das Behandlungsprofil ein. Verwendet auch die erweiterte Einstellung:

- `PROFILE_HISTORY (off)` - mögliche Werte `on` oder `off`. Aktivieren / Deaktivieren der NS-Fähigkeit zur Protokollierung Ihrer Profile (noch experimentell)
- `PROFILE_MULTIPLE (off)` - mögliche Werte `on` oder `off`. Aktivieren / Deaktivieren der NS-Fähigkeit zum Handhaben und Umschalten zwischen mehreren Behandlungsprofilen

2.6.6.3 Erweiterte Plugins:

2.6.6.3.1 Careportal (Careportal)

Eine optionale Form, Therapiemaßnahmen einzugeben.

2.6.6.3.2 Boluscalc (de: Bolus Rechner)

Nightscout berechnet anhand bestimmter Vorgaben die zu verabreichende Bolusmenge.

2.6.6.3.3 Food (de: Nahrungsmittel)

Wir können über den `Nahrungsmittel - Editor` siehe () Mahlzeiten mit Angabe von Kohlenhydraten eingeben

2.6.6.3.4 rawbg (Roh Blutzuckerwerte)

Berechnet Roh Blutzuckerwerte mit Sensor und Kalibrierungsaufzeichnungen und zeigt alternative Werte mit Verzerrungspegel an.

2.6.6.3.5 Iob (en: Insulin-on-Board, de: Wirksames Insulin)

Fügt die IOB Pillbox Visualisierung im Browser ein und berechnet das noch wirksame Bolus Insulin, welches von anderen Plugins verwendet werden kann. Verwendet Eingaben mit `Insulindosen` und der `dia` und `sens` Felder aus dem [Behandlungsprofil] (# behandlungs Profil).

2.6.6.3.6 Cob (Carbs-on-Board)

Fügt die COB Pillbox Visualisierung im Browser ein und berechnet Werte, die von anderen Plugins verwendet werden können. Verwendet Eingaben (treatments) mit carb Angaben und der `carbs_hr`, `carbratio` und `sens` Felder aus dem [Behandlungsprofil] (# Behandlungs Profil).

2.6.6.3.7 Bwp (BolusExpert Vorschau)

Dieses Plugin hat erstens den Zweck, Alarmer automatisch schlafen zu legen, wenn das CGM zwar hohe Blutzuckerwerte anzeigt, es aber noch ausreichend Insulin an Bord (IOB) gibt und zweitens, um Benutzer darauf hinzuweisen, dass es günstig sein könnte, den Blutzucker mit einem Blutzuckermessgerät zu bestimmen und Insulin laut Berechnung durch die Insulinpumpe oder wie von medizinischem Fachpersonal empfohlen abzugeben. *** Die vom Plugin zur Verfügung gestellten Werte sind als Hinweis gedacht basierend auf CGM-Daten und der Insulinempfindlichkeit, die du konfiguriert hast, sie sind nicht als Empfehlung für die Bolus-Berechnung gedacht. *** Das Plugin berechnet die Bolus-Menge, wenn der Wert über dem von dir angegebenen Zielwert liegt, erzeugt Alarmer, wenn du Prüfung und die Abgabe eines Bolus in Betracht ziehen solltest, und Alarmer snoozes, wenn genug IOB vorliegt, um eine hohe BG zu decken. Verwendet die Ergebnisse des `iob` Plugin und `sens`, `target_high` und `target_low` Felder aus dem [Behandlungsprofil] (# behandlungs Profil). Defaults, die mit angepasst werden können [erweiterte Einstellung] (# extended-Einstellungen)

- `BWP_WARN (0.50)` - Wenn BWP größer ist als der `BWP_WARN` wird ein Warnalarm ausgelöst.
- `BWP_URGENT (1.00)` - Wenn BWP größer als dieser Wert ist, wird ein dringender Alarm `BWP_URGENT` ausgelöst.
- `BWP_SNOOZE_MINS (10)` - Die Anzahl der Minuten, in der kein Alarm ausgelöst wird, wenn noch genügend IOB vorhanden ist, um eine hohe BG abzudecken.
- `BWP_SNOOZE - (0.10)` Wenn BG höher als die `target_high` und `BWP < BWP_SNOOZE` Alarmer für `BWP_SNOOZE_MINS` werden snoozed ist.
-

2.6.6.3.8 Cage (en: Cannula Age, de: Pumpenkatheter Setz-Alter)

Berechnet die Anzahl der Stunden seit der letzten `Site Change` Eingabe/Bearbeitung, die aufgezeichnet wurde.

- `CAGE_ENABLE_ALERTS (false)` - Wenn du stattdessen auf `true` einstellst, kannst du dich an einen Katheterwechsel erinnern lassen.
- `CAGE_INFO (44)` - Wenn die hinter `CAGE_INFO` eingestellte Stundenanzahl mit der Zahl seit dem letzten `Website Change` übereinstimmt, wirst du an einen demnächst fälligen Katheterwechsel erinnert.
- `CAGE_WARN (48)` - Wenn die hinter `CAGE_WARN` eingestellte Stundenanzahl mit der Zeit seit dem letzten `Website Change` übereinstimmt, wirst du an den nun fälligen Katheterwechsel erinnert.
- `CAGE_URGENT (72)` - Wenn die hinter `CAGE_URGENT` eingestellte Stundenanzahl die Zeit seit dem letzten `Website Change` erreicht, wird von da ab eine dauerhafte Warnung ausgegeben, dass der Katheterwechsel überfällig ist.
- `CAGE_DISPLAY (hours)` - Mögliche Werte sind “Stunden” oder “Tage”. Wenn “Tage” ausgewählt ist und das Alter des Katheters größer ist als 24-Stunden, wird in Tagen und Stunden angezeigt.

2.6.6.3.9 Sage (Sensor-Alter)

Berechnet die Anzahl der Tage und Stunden seit der letzten `Sensor Starten` und `Sensor Change` Eingabe/Bearbeitung, die aufgezeichnet wurde.

- `SAGE_ENABLE_ALERTS` (`false`) - Stattdessen auf `true` eingestellt, kannst du dich an einen Sensorwechsel erinnern lassen.
- `SAGE_INFO` (144) - Wenn die hinter `SAGE_INFO` eingestellte Stundenanzahl mit der Zeit seit dem letzten dokumentierten Sensorwechsel bzw. Sensorstart übereinstimmt, wirst du an einen demnächst fälligen Sensorwechsel erinnert.
- `SAGE_WARN` (164) - Wenn die hinter `SAGE_WARN` eingestellte Stundenanzahl mit der Zeit seit dem letzten Sensorwechsel bzw. -start übereinstimmt, wirst du an den nun fälligen Sensorwechsel oder -neustart erinnert.
- `SAGE_URGENT` (166) - Wenn die hinter `SAGE_URGENT` eingestellte Stundenanzahl erreicht ist, wird der Benutzer dauerhaft an einen überfälligen Wechsel/Neustart erinnert.

2.6.6.3.10 Iage (Insulin-Alter)

Berechnet die Anzahl der Tage und Stunden seit der letzten `Insulin Change` Eingabe, die aufgezeichnet wurde.

- `IAGE_ENABLE_ALERTS` (`false`) - Stattdessen auf `true` eingestellt, kannst du dich an einen Insulinreservoirwechsel erinnern lassen.
- `IAGE_INFO` (44) - Wenn die hinter `IAGE_INFO` eingestellte Stundenanzahl mit der Zeit seit dem letzten dokumentierten `Insulin Change` übereinstimmt, wirst du an einen demnächst fälligen Reservoirwechsel erinnert.
- `IAGE_WARN` (48) - Wenn die hinter `IAGE_WARN` eingestellte Stundenanzahl mit der Zeit seit dem letzten `Insulin Change` übereinstimmt, wirst du an den nun fälligen Reservoirwechsel erinnert.
- `IAGE_URGENT` (72) - Wenn die hinter `IAGE_URGENT` eingestellte Stundenanzahl mit der dokumentierten `Insulin Change` Stundenanzahl übereinstimmt, wirst du dauerhaft an einen überfälligen Reservoirwechsel erinnert.

2.6.6.3.11 Treatmentnotify (Behandlung Benachrichtigungen)

Erzeugt Benachrichtigungen, wenn eine Therapiemaßnahme eingegeben wurde, und legt Alarmer für bestimmte Minuten nach einer Eingabe schlafen. Die Standardeinstellung ist 10 Minuten und kann über `TREATMENTNOTIFY_SNOOZE_MINS` [Erweiterte Einstellung] (# extended-Einstellungen) verändert werden.

2.6.6.3.12 Basal (Basal-Profil)

Fügt ein Textfeld, eine sog. Pillbox, für das Basalinsulin ein, in dem du die aktuelle Basalrate ablesen kannst. Außerdem ermöglicht es dem `bwp` Plugin, Vorschläge für eine temporäre Basalkorrektur zu berechnen. Verwendet das `basal` Feld aus dem [Behandlungsprofil] (# behandlungs Profil). Es nutzt auch die erweiterte Einstellung:

- `BASAL_RENDER` (`none`) - Mögliche Werte sind `none`, `default` oder `icicle` (invertiert, auf den Kopf gestellt)

2.6.6.3.13 Bridge (Share2Nightscout Brücke)

Die Share2Nightscout Bridge Funktionalität ist derzeit nur für Dexcom Share Benutzer interessant. Die Verbindung zum Dexcom G5 System wird über diesen Weg konfiguriert. Um die Bridge Funktionalität nutzen zu können, müssen wir **bridge** zu **ENABLE** in den App-Einstellungen hinzufügen. Die Einstellungen erfolgen über die `Connection Strings`.

- `BRIDGE_USER_NAME` - Dein Benutzername für den Share-Dienst
- `BRIDGE_PASSWORD` - Dein Passwort für den Share-Dienst
- `BRIDGE_INTERVAL` (150000 * 2,5 Minuten *) - Die Wartezeit zwischen den Updates
- `BRIDGE_MAX_COUNT` ('1') - Die maximale Anzahl der Datensätze pro Update
- `BRIDGE_FIRST_FETCH_COUNT` (3) - Ändert `Bridge_Max_Count` nur während des ersten Updates
- `BRIDGE_MAX_FAILURES` (3) - Die Anzahl der Fehlversuche, bevor aufgegeben wird
- `BRIDGE_MINUTES` (1400) - Das Zeitfenster für die Suche nach neuen Updates (Standard ist ein Tag in Minuten)

2.6.6.3.14 Mmconnect (MiniMed Connect-Brücke)

Überträgt Echtzeit MiniMed Connect Daten vom Medtronic Carelink-Server an Nightscout ([mehr lesen] (<https://github.com/mddub/minimed-connect-to-nightscout>))

- `MMCONNECT_USER_NAME` - Dein Benutzername für CareLink® Connect
- `MMCONNECT_PASSWORD` - Dein Passwort für CareLink® Connect
- `MMCONNECT_INTERVAL` (60000 * 1 Minute *) - Anzahl der Millisekunden zwischen den einzelnen Anfragen an den CareLink® Server
- `MMCONNECT_MAX_RETRY_DURATION` (32) - Maximale Gesamtanzahl der Sekunden, nach denen bei fehlgeschlagenen Anfragen aufgegeben wird
- `MMCONNECT_SGV_LIMIT` (24) - Maximale Anzahl der letzten Sensorwerte, die bei jeder Anfrage gesendet werden
- `MMCONNECT_VERBOSE` - Setze dies auf "true", um Informationen über CareLink® Anfragen auf der Konsole wiederzugeben.
- `MMCONNECT_STORE_RAW_DATA` - Setze dies auf "true", um Rohdaten zu speichern, die von CareLink® als Typ: `"carelink_raw"` Datenbankeinträge zurück kommen (nützlich für die Entwicklung).

2.6.6.3.15 Pump (Pumpenüberwachung)

Allgemeingültige Pumpenüberwachung für OpenAPS, MiniMed Connect, RileyLink, t:slim, mit mehr auf dem Weg

- Erfordert `DEVICESTATUS_ADVANCED = " true "`
- `PUMP_ENABLE_ALERTS` (false) - Auf `true` werden Anzeigen über die Pumpenbatterie und das -reservoir bereitgestellt.
- `PUMP_FIELDS` (reservoir battery) - Diese Felder werden standardmäßig angezeigt. Wähle aus: `reservoir,battery,clock,status` und `device`
- `PUMP_RETRO_FIELDS` (reservoir battery clock) - Die Textfelder, die im Retro-Modus angezeigt werden. Jede der oben genannten Möglichkeiten.

- `PUMP_WARN_CLOCK (30)` - Die Anzahl der Minuten, die verstrichen sein müssen, bevor ein Alarm ausgelöst wird
- `PUMP_URGENT_CLOCK (60)` - Die Anzahl der Minuten, die verstrichen sein müssen, bevor ein dringender Alarm ausgelöst wird
- `PUMP_WARN_RES (10)` - Bei Unterschreiten der ausgewählten Insulinrestmenge im Reservoir wird eine Warnung ausgelöst.
- `PUMP_URGENT_RES ( '5' )` - Bei Unterschreiten der ausgewählten Insulinrestmenge wird ein dringender Alarm ausgelöst.
- `PUMP_WARN_BATT_P (30)` - Bei Unterschreiten der ausgewählten Restleistung der Pumpenbatterie in % wird eine Warnung ausgelöst.
- `PUMP_URGENT_BATT_P (20)` - Bei Unterschreiten der ausgewählten Restleistung der Batterie in % wird ein dringender Alarm ausgelöst.
- `PUMP_WARN_BATT_V (1.35)` - Bei Unterschreiten der ausgewählten Restleistung der Pumpenbatterie in Voltzahl (falls Prozent nicht verfügbar ist) wird eine Warnung ausgelöst.
- `PUMP_URGENT_BATT_V (1.30)` - Bei Unterschreiten der ausgewählten Restleistung der Batterie in Voltzahl (falls Prozent nicht verfügbar ist) wird ein dringender Alarm ausgelöst.

2.6.6.3.16 Openaps (OpenAPS)

Integrierte OpenAPS Loopüberwachung nutzt diese erweiterten Einstellungen:

- `DEVICESTATUS_ADVANCED = " true "` muss gesetzt werden
- `OPENAPS_ENABLE_ALERTS (false)` - Auf `true` setzen, um benachrichtigt zu werden, wenn OpenAPS nicht loopt. Wenn du mit OpenAPS für einen bestimmten Zeitraum offline gehst, kannst du ein OpenAPS Offline Ereignis für die erwartete Dauer vom Careportal aus hinzufügen, um Warnmeldungen zu vermeiden.
* `OPENAPS_WARN (30)` - Die Anzahl der Minuten seit dem letzten Loop, die überschritten werden muss, bevor ein Alarm ausgelöst wird
* `OPENAPS_URGENT (60)` - Die Anzahl der Minuten seit dem letzten Loop, die überschritten werden muss, bevor ein dringender Alarm ausgelöst wird
- `OPENAPS_FIELDS (status-symbol status-label iob meal-assist rssi)` - Die standardmäßig angezeigten Textfelder. Jedes der folgenden Felder ist möglich: `status-symbol`, `status-label`, `iob`, `meal-assist`, `freq` und `rssi`
- `OPENAPS_RETRO_FIELDS (status-symbol status-label iob meal-assist rssi)` - Die Felder, die im Retro-Modus angezeigt werden sollen. Jede der oben genannten Möglichkeiten.

2.6.6.3.17 loop (Loop)

IOS Loop App-Überwachung, verwendet diese erweiterten Einstellungen:

- * `DEVICESTATUS_ADVANCED(true)` - als gesetzten Wert
- * `LOOP_ENABLE_ALERTS (false)` - Auf `true` setzen, um Benachrichtigungen zu aktivieren, wenn der Loop keine Werte liefert
- * `LOOP_WARN (30)` - Die Anzahl der Minuten seit des letzten Loops, die überschritten werden muss, bevor ein Alarm ausgelöst wird
- * `LOOP_URGENT (60)` - Die Anzahl der Minuten seit des letzten Loops, die überschritten werden muss, bevor ein dringender Alarm ausgelöst wird
- * Füge `loop` zu `SHOW_FORECAST` hinzu, um prognostizierten BZ anzuzeigen.

2.6.7 Extended Settings

Einige Plugins unterstützen zusätzliche Konfigurationen mit zusätzlichen Umgebungsvariablen. Diese werden mit dem Namen des Plugins und einem `_` vorangestellt. Beispielsweise würde die Einstellung `MYPLUGIN_EXAMPLE_VALUE = 1234` dem MYPLUGIN-Plugin den Befehl `extendedSettings.exampleValue` zur Verfügung stellen.

Plugins haben nur Zugriff auf ihre eigenen erweiterten Einstellungen, alle erweiterten Einstellungen von Client-Plugins werden an den Browser gesendet.

2.6.7.1 Pushover

Zusätzlich zu den normalen web-basierten Alarmen können auch [Pushover] Alarme (<https://pushover.net/>) eingerichtet werden.

Zuerst installierst du die Pushover-Anwendung auf deinem iOS oder Android-Gerät und erstellst ein Konto .

Beim Konto Login in [Pushover] (<https://pushover.net/>) findest du oben links deinen User Key, du benötigst diesen Key und einen Anwendungs API Token / Schlüssel, um dieses Setup abzuschließen.

Gehe auf [Erstellen eines Pushover Application] (<https://pushover.net/apps/build>). Du benötigst nur einen Namen für die Anwendung, um sie zu installieren, alle anderen Einstellungen kannst du ignorieren.

Pushover ist mit folgenden Umgebungsvariablen konfiguriert:

- `ENABLE -pushover` sollte zu der Liste der Plugins hinzugefügt werden, zum Beispiel: `ENABLE = "Pushover "`.
- `PUSHOVER_API_TOKEN` - Um Pushover Benachrichtigungen zu ermöglichen, ist dieses Token speziell für deine erstellte Anwendung [Pushover] (<https://pushover.net/>) *** [zusätzliche Pushover Informationen] (# Pushover) *** unten.
- `PUSHOVER_USER_KEY` - Deinen Pushover Benutzerschlüssel findest du in der oberen linken Ecke der [Pushover] (<https://pushover.net/>) Seite. Es kann sich dabei auch um einen Benutzerschlüssel für eine Pushover delivery group statt für einen einzelnen Benutzer handeln. Dies unterstützt auch eine durch Leerzeichen getrennte Liste von Schlüsseln. Um INFO Level pushes zu deaktivieren, stelle dies auf `off`.
- `PUSHOVER_ALARM_KEY` - Ein optionaler Pushover Benutzer- / Gruppenschlüssel wird für systemweite Alarme (`level>"WARN"`) verwendet. Wenn hier nichts definiert ist, wird ersatzweise auf `PUSHOVER_USER_KEY` zurückgegriffen. Eine mögliche Verwendung für dies ist, wichtige Meldungen und Alarme an einen CWD zu senden, an den du nicht alle Benachrichtigungen senden möchtest. Auch dies unterstützt eine durch Leerzeichen getrennte Liste von Schlüsseln. Zum Deaktivieren der Alarmpushes setze dies auf `off`.
- `PUSHOVER_ANNOUNCEMENT_KEY` - Ein optionaler Pushover Benutzer- / Gruppenschlüssel wird für systemweite vom Benutzer erzeugte Ankündigungen verwendet. Wenn hier nichts definiert ist, wird auf `PUSHOVER_USER_KEY` oder `PUSHOVER_ALARM_KEY` zurückgegriffen. Dies unterstützt ebenfalls eine durch Leerzeichen getrennte Liste von Schlüsseln. Zum Deaktivieren setzt du dies auf `off`.
- `BASE_URL` - Wird für Pushover Rückrufe genutzt, in der Regel die URL deiner Nightscout Website, verwende dafür möglichst `https`.
- `API_SECRET` - wird für die Pushover Rückrufanfrage als Bestätigung verwendet für deine Identifizierung.

Wenn du keine Infolevel Benachrichtigungen (treatments, Therapiemaßnahmen) erhalten möchtest, verwende `PUSHOVER_USER_KEY = " off "`

Wenn du keinen Alarm über Pushover erhalten möchtest, verwende `PUSHOVER_ALARM_KEY = " off "`

Wenn du keine Mitteilung über Pushover erhalten möchtest, verwende `PUSHOVER_ANNOUNCEMENT_KEY = "off"`

Wenn nur `PUSHOVER_USER_KEY` gesetzt ist, wird es für alle Info-Benachrichtigungen, Alarmer und Mitteilungen verwendet werden.

Für Test / Entwicklung versuche [localtunnel] (<http://localtunnel.me/>).

IFTTT Maker

Zusätzlich zu den normalen web-basierten Alarmen und zu Pushover, gibt es auch die Integration für [IFTTT Maker] (<https://ifttt.com/maker>).

Mit Maker kannst du dich mit allen anderen [IFTTT Kanälen] (<https://ifttt.com/channels>) verknüpfen. Zum Beispiel kannst du einen Tweet senden, wenn es eine Warnung gibt, die Farbe von Hue Lampen ändern, eine E-Mail oder eine SMS senden und vieles mehr.

1. Setup-IFTTT Konto: [Anmelden] (<https://ifttt.com/login>) oder [ein Konto erstellen] (<https://ifttt.com/join>)
2. Finde deinen Geheimschlüssel auf der [Hersteller Seite] (<https://ifttt.com/maker>)
3. Konfiguriere Nightscout mithilfe dieser Umgebungsvariablen:
 - `ENABLE-maker` sollte zu der Liste der Plugins hinzugefügt werden, zum Beispiel: `ENABLE = "maker"`.
 - `MAKER_KEY` - Setze diesen auf deinen Geheimschlüssel, siehe Schritt 2, zum Beispiel: `MAKER_KEY = "abcMyExampleabc123defjtlDeNsiftttmak-XQb69p"` Dies unterstützt auch eine durch Leerzeichen getrennte Liste von Schlüsseln.
 - `MAKER_ANNOUNCEMENT_KEY` - Ein optionaler Maker Schlüssel wird für systemweite Benutzer erzeugt Ankündigungen verwendet werden. Wenn nicht definiert ist, wird dies auf `MAKER_KEY` Rückfall. Eine mögliche Verwendung für diese sendet wichtige Meldungen und Alarmer an einen CWD, die Sie wollen nicht zu alle Benachrichtigungen zu senden. Dies ist auch eine durch Leerzeichen getrennte Liste von Schlüsseln unterstützen.
1. [Erstellen Sie ein Rezept] (<https://ifttt.com/myrecipes/personal/new>) oder siehe [ausführliche Anleitung] (lib / plugins / maker-setup.md # create-a-Rezept)

Plugins können benutzerdefinierte Ereignisse erstellen, aber alle Ereignisse an Hersteller gesendet werden mit `ns-` vorangestellt werden. Die Kern Ereignisse sind:

- `ns-event` - Dieses Ereignis wird für alle Alarmer und Benachrichtigungen an den Hersteller-Service gesendet. Das ist gut, fangen alle Ereignis für die allgemeine Protokollierung.
- `ns-allclear` - Dieses Ereignis wird an den Hersteller-Service gesendet wird, wenn ein Alarm ack'd wurde oder wenn der Server startet ohne Alarmer auslösen. Zum Beispiel könnten Sie mit diesem Ereignis ein Licht grün.
- `ns-info` - Plugins, die Benachrichtigungen auf der Infoebene generieren wird dieses Ereignis dazu führen, auch ausgelöst werden. Es wird zusätzlich zu `ns-event` gesendet werden.
- `ns-warning` - Alarmer auf der Warnstufe mit Ursache dieses Ereignis auch ausgelöst werden. Es wird zusätzlich zu `ns-event` gesendet werden.
- `ns-urgent` - Alarmer auf dringendes Ebene mit Ursache dieses Ereignis auch ausgelöst werden. Es wird zusätzlich zu `ns-event` gesendet werden.
- siehe die [vollständige Liste der Veranstaltungen] (lib / plugins / maker-setup.md # events)

2.6.8 Behandlungsprofil

Einige der Plugins verwenden ein Behandlungsprofil, das mit dem ***Profileditor*** bearbeitet werden kann

Behandlungsprofil - Attribute:

- * `timezone` (Zeitzone) - lokale Zeitzone. Sollte eingestellt werden.
- * `units` (Profil - Einheiten) - Blutglukoseeinheiten, die im Profil verwendet werden, entweder "mgdl" oder "mmol"
- * `dia` (Insulinwirkzeit) - Wert sollte die Dauer der Insulinwirkung bei der Berechnung, wie viel Insulin aktiv bleibt, sein. Die Voreinstellung beträgt 3 Stunden.
- * `carbs_hr` (KH pro Stunde) - Die Anzahl der Kohlenhydrate, die pro Stunde verarbeitet werden, näheres siehe [#DIYPS] (<https://diyyps.org/2014/05/29/determining-your-carbohydrate-absorption-rate-diyyps-lessons-learned/>)

- `carbratio` (KH Faktor) - Gramm pro Einheit Insulin
- `sens` (Insulin Sensitivität) - Um wieviel eine Einheit Insulin den BZ senkt
- `basal` Die Basalrate, welche zur Pumpe gesendet wird
- `target_high` - Oberes Ziel für Korrekturbolus.
- `target_low` - Unteres Ziel für Korrekturbolus.

Einige Beispiele findet man [hier] (<https://github.com/nightscout/cgm-remote-monitor/blob/master/example-profiles.md>)

2.6.9 Setting environment variables

Einfach zu emulieren auf der Kommandozeile:

```
echo 'MONGO_CONNECTION=mongodb://sally:sallypass@ds099999.mongolab.com:99999/'  
↪nightscout' >> my.env  
echo 'MONGO_COLLECTION=entries' >> my.env
```

Von nun an kann der Befehl ausgeführt werden:

```
$ env $(cat my.env) PORT=1337 node server.js
```

Ein Hosting Provider bietet evtl. ein GUI zur Eingabe an.

2.6.10 Vagrant installieren

Optionally, use Vagrant with the included Vagrantfile and setup.sh to install OS and node packages to a virtual machine.

Optional können Sie [Vagrant] (<https://www.vagrantup.com/>) mit der mitgelieferten Vagrantfile und setup.sh verwenden, um OS- und Knotenpakete auf einer virtuellen Maschine zu installieren.

```
host$ vagrant up  
host$ vagrant ssh  
vm$ setup.sh
```

Das Setup-Skript installiert die Betriebssystempakete und führt dann die Installation von npm durch.

The Vagrant VM serves to your host machine only on 192.168.33.10, you can access the web interface on <http://192.168.33.10:1337>

Die Vagrant VM dient nur auf Ihrem Host-Computer mit Beispiel IP 192.168.33.10, man kann auf das Web-Interface über <http://192.168.33.10:1337> zugreifen.

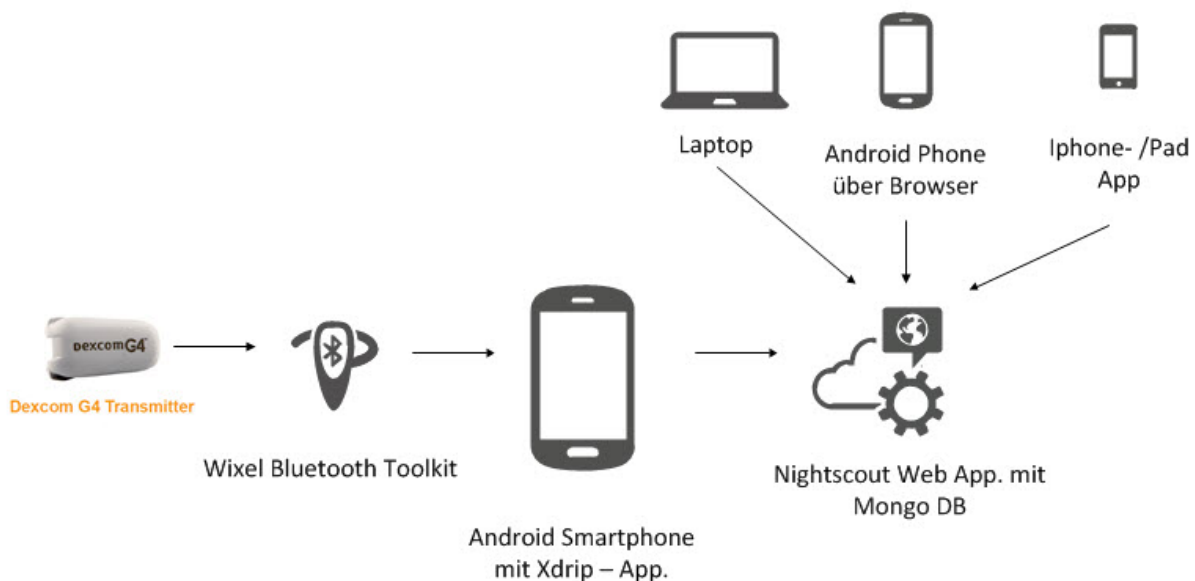
2.7 Die Nightscout Website

Die Nightscout Website ist die zentrale Website zur Darstellung und Speicherung von Diabetes - Daten. Sie kann als komplettes Tagebuch mit Eingabe von Insulin, Kohlenhydraten, Bewegung uvm. verwendet werden. Mit der Integration von **OpenAPS** wird der gesamte Verlauf der Pumpensteuerung mit temporären Basalraten dargestellt. Der gesamte CGM - Verlauf wird online dargestellt und zyklisch aktualisiert. Es gibt umfangreiche **Einstellungsmöglichkeiten** für die Website. Das **Behandlungsportal** bietet eine umfangliche Auswahl an Behandlungsdaten. Über den **Bolus Rechner** können für Mahlzeiten abzugebende Boluseinheiten berechnet werden. Das **Reporting** bietet die Möglichkeit, die Daten auszuwerten und für andere zur Analyse auszugeben. Mit dem **Profileditor** hat man die Möglichkeit, Parameter in unterschiedlichen Profilen abzuspeichern, die z.B. für die Pumpensteuerung benötigt werden. Die Grafik wird standardmäßig alle 60 Sekunden aktualisiert.

Möchte man sich die FGM/CGM - Werte anzeigen lassen, so sind die Einstellungen zum Übertragen der Daten in der Uploader-App (*xDrip*, NightScout-Uploader oder Glimp) erforderlich.

Ein Beispiel für eine mögliche Anbindung eines CGM Systems an die Nightscout Website:

Kommunikation:



2.7.1 Haupt Seite

Nightscout ist seit Oktober 2016 in der Version **0.9 - Grilled Cheese** verfügbar. Wer Vorgängerversionen nutzt, muss Einstellungen für die Website in Azure oder Heroku überprüfen und ergänzen.

Variable unter den App - Einstellungen zum Hinzufügen: `AUTH_DEFAULT_ROLES` mit `readable`, `status-only` als mögliche Werte. `devicestatus-Upload`

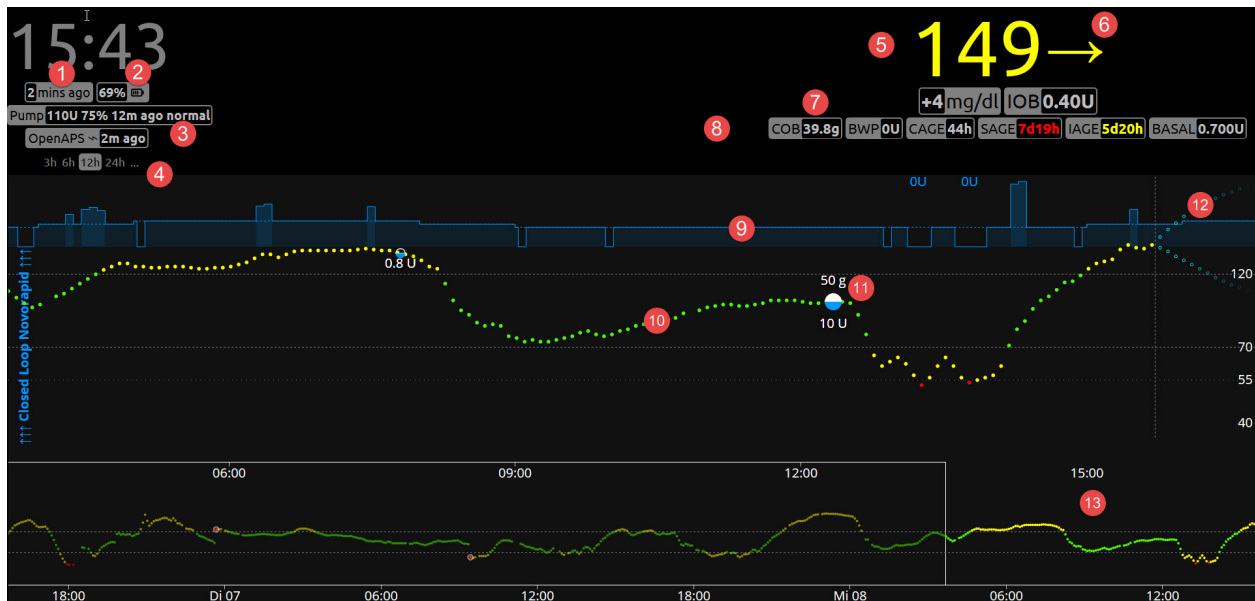
Wenn wir die Website aktualisieren und feststellen, dass Dinge, die wir aktiviert haben, nicht richtig angezeigt werden (wie `boluscalc`), bitte folgendes überprüfen:

Bitte Sicherstellen, dass es nur eine die ENABLE - Zeile in den gesamten Einstellungen der Website gibt. (Das bedeutet nur ein einziges Mal in den Gesamteinstellungen, einschließlich App-Einstellungen und Verbindungszeichenfolgen.) Wenn ENABLE in Connection Strings auftaucht , bitte in die App Settings verschieben. Speichern nicht vergessen und anschließend die Website Stoppen und Starten.

Grilled Cheese benötigt ein Profil

Wenn wir zuvor kein Profil auf der Website eingerichtet haben, werden wir beim ersten Aufruf der Website aufgefordert, ein Profil zu erstellen. Wir können alle Informationen eingeben, oder nur die Zeitzone (und das Profil später bearbeiten). Wir müssen die Website authentifizieren, bevor wir die Profiländerungen speichern können. Um sich zu authentifizieren, verwenden wir den Link am unteren Rand des Profils Bildschirms und geben das **API SECRET** ein. Sobald wir authentifiziert sind, ändern wir das Profil und klicken dann auf **Speichern**. Wenn das Profil korrekt gespeichert ist, sehen wir **Erfolg** unten und rechts oben. Wir können nun die Website öffnen.

Wir sehen uns nun die **Hauptseite** an. Die Grafik wird alle 60 Sekunden aktualisiert.



Was sehen wir ?

1. Zeit seit der Letzten CGM Sensor - Daten - Übertragung
2. Batterie - Ladezustand des Smartphones
3. Plugins für die Pumpensteuerung
4. Zeitrahmen der Darstellung von Behandlungsdaten
5. aktueller BZ - Wert
6. Trendpfeilangabe
7. Zunahme/Abnahme des BZ innerhalb von 2 Messpunkten
8. Plugins mit Anzeige von Werten wie Sensor - Alter , Insulin on Board
9. Basalratenverlauf

10. CGM - Meßwerte
11. Behandlungsdaten Kohlenhydrate - und Insulineinheiten
12. BZ - Trend - Voraussage
13. Zeitfenster der Darstellung im oberen Bereich

In der rechten oberen Ecke finden wir **Menü - Auswahlfelder**.



Bearbeitungsmodus

`EDIT_MODE (on)` - mögliche Werte `on` oder `off`. Wenn diese Option aktiviert ist (rot) , kann man z.B. Behandlungsdaten grafisch mit der Maus bearbeiten, indem man eingegebene Werte auf der Zeitachse verschiebt.

Mit dem **Lautsprecher** - Symbol kann man eingestellte Alarme testen.

Das Symbol mit den **3 Strichen** ist das Einstellungsmodul mit umfangreichen Möglichkeiten.

Mit dem Klick auf das **+** - Symbol rufen wir das **Behandlungsportal** auf.

Mit dem **Taschenrechner** - Symbol gelangen wir zum **Bolus - Rechner**.

2.7.2 Einstellungen

Die **Einstellungen** für die Website erreichen wir mit dem Klick auf das mit dem **roten** Pfeil markierte Symbol:

Oben haben wir die Wahl von 4 Menüpunkten:

- **Berichte**
- **Profil - Editor**
- **Nahrungsmittel - Editor**
- **Administrator Werkzeuge**

Unten sind die Browsereinstellungen gelistet. Wenn wir diese zentral für alle Browser gleich halten wollen, können wir dieses wie in Kapitel *Nightscout Plugins* beschrieben, umsetzen.



2.7.2.1 Berichte

Wir starten das **Reporting** mit dem Klick auf **Berichte** und erhalten diese Ansicht:

Nightscout reporting

Report - Arten

Bearbeitungsmöglichkeit eingetragener Werte

Von Tag zu Tag Tägliche Statistik Streuung Stündliche Statistik Durchschnittswert Wöchentlicher Erfolg Kalibrierung **Bearbeitung**

☒ Von: 2016-01-07 Bis: 2016-01-13 Heute letzten 2 Tage letzten 3 Tage letzte Woche letzten 2 Wochen letzter Monat letzten 3 Monate

☐ Erläuterungen:

☐ Ereignis-Typ:

☒ Mo ☒ Di ☒ Mi ☒ Do ☒ Fr ☒ Sa ☒ So

Untergrenze des Blutzuckerzielbereiches: 80 oben: 180

Order: ☒ oldest on top ☐ newest on top

Zeigen Report Ausgabe Starten

Darstellen: ☒ Insulin ☒ Kohlenhydrate ☒ Basalrate ☒ Notiz ☒ Essen ☐ Raw ☐ IOB ☐ COB Größe 1000x300px

Messen: ☐ Linear ☒ Logarithmisch

Auswahlmöglichkeiten Inhalt

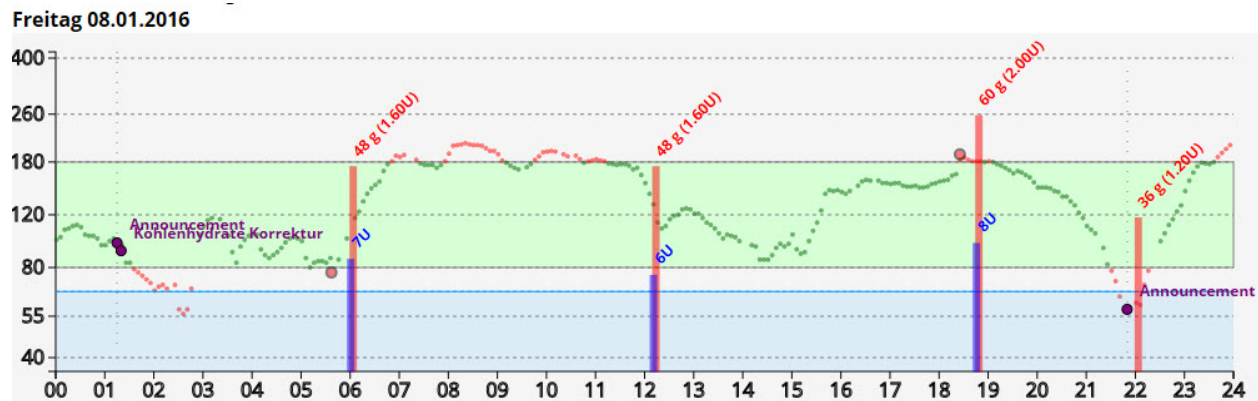
Authentications Status:
Gerät authentifiziert (Entfernen)

Die **Report - Arten** sind umfangreich. Wir können sie uns mit Klicks auf den Namen aufrufen.

Mit der Eingabe der **Zeiträume** bestimmt man den Report - Umfang. Wir können vorgefertigte Zeiträume und individuelle Zeiträume auswählen. Weitere Filter sind möglich.

Wir starten die Ausgabe des Reports mit dem Klick auf **Zeigen**.

Mit dem Report **Von Tag zu Tag** gibt es alle Daten auf einen Blick:



Wir haben auch die Möglichkeit, eingetragene Werte mit dem Klick auf **Bearbeitung** zu **Korrigieren** oder zu **Löschen**:

Nightscout reporting

[Von Tag zu Tag](#) | [Tägliche Statistik](#) | [Streuung](#) | [Stündliche Statistik](#) | [Durchschnittswert](#) | [Wöchentlicher Erfolg](#) | [Kalibrierung](#) | [Bearbeitung](#)

☒ Von: Bis:
[Heute](#) [letzten 2 Tage](#) [letzten 3 Tage](#) [letzte Woche](#) [letzten 2 Wochen](#) [letzter Monat](#) [letzten 3 Monate](#)

☐ Erläuterungen:

☐ Ereignis-Typ:

☒ Mo ☒ Di ☒ Mi ☒ Do ☒ Fr ☒ Sa ☒ So

Untergrenze des Blutzuckerzielbereiches: oben:

Order: ☒ oldest on top ☐ newest on top

Bearbeitung

	Zeit	Ereignis-Typ	Blutglukose	Insulin	Kohlenhydrate	Duration	Percent	Basal value	Eingabe durch	Notiz
Mittwoch 13.01.2016										
✗/✓	01:35:00	Announcement								CGM Alarm
✗/✓	01:40:00	Kohlenhydrate Korrektur								
✗/✓	06:45:00	Mahlzeiten Bolus		8	48					
✗/✓	11:45:00	Mahlzeiten Bolus		6	48					
✗/✓	18:40:16	Mahlzeiten Bolus		5	36					
✗/✓	20:30:00	Kohlenhydrate Korrektur			18					Mattigkeit
✗/✓	20:41:31	CGM Alarm								CGM Alarm
✗/✓	21:45:00	Announcement								CGM Alarm
✗/✓	21:50:00	Kohlenhydrate Korrektur			24					
Donnerstag 14.01.2016										

2.7.2.2 Profil - Editor

Der Behandlungs Profil Editor speichert angepasste Werte für die persönliche Therapie. Mit dem Profil - Editor konfigurieren wir Einheiten und Werte für Plugin iob, cob, bwp und Basalraten.

Über einen neuen Datenbankeintrag durch Klicken auf das + - Symbol können wir ein alternatives Profil erstellen. Der NS android Client zur Steuerung der Kommunikation zwischen Pumpe und CGM erfragt Werte aus den abgespeicherten Profilen.

Wichtig: Ohne Änderungen im Editor werden für die Plugins keine verwendbaren Werte angezeigt. Seit Grilled Cheese, Version 0.9 ist das Anlegen eines Profils beim ersten Aufruf der Website erforderlich.


Nightscout

Profil-Editor

Allgemeine Profileinstellungen

Überschrift: Nightscout
Einheiten: mg/dl
Datumsformat: 24h

Datenbankeinträge:
Gültig ab: 13.7.2016 13:13:45
+
X
↺

Eintrag gültig ab:
2016-07-13
13:13

Gesicherte Profile:
Default
+
X
↺

Name: sports

Zeitzone: Europe/Berlin

Dauer der Insulinaktivität (DIA) [Stunden]: 3

Insulin/Kohlenhydrate-Verhältnis (I:KH) [g]:

Von: 00:00
I:KH : 7
+

Insulinsensibilitätsfaktor (ISF) [mg/dL/U, mmol/L/U]:

Von: 00:00
ISF : 50
+

Kohlenhydrate

Kohlenhydrataktivität / Aufnahme Kohlenhydrate: [g/Std]
20

Basalraten [Einheit/h]

Von: 00:00
Basalrate : 0.08
+

Blutzucker-Zielbereich [mg/dL, mmol/L]

Von: 00:00
Tief : 0
Hoch : 0
+

Speichern

Wenn wir die Maus oder den Finger über Fachbegriffe bewegen, werden **Erklärungstexte** in deutscher Sprache angezeigt sofern Nightscout auf Deutsch eingestellt wurde.

Bei **Zeitzone** bitte die entsprechende Zeitzone Einstellen.

Mit **Dauer der Insulinaktivität (DIA) in Stunden** definieren wir die Wirkdauer des Bolus - Insulins. Die Plugins iob, bwp benötigen diese Eingaben.

Mit **Insulin/Kohlenhydrate-Verhältnis (I:KH) in Gramm** wird angegeben, wieviel Boluseinheiten für eine definierte Menge an KH verwendet werden müssen. Die Plugins iob und cob benötigen diese Eingaben.

Mit dem **Insulinsensibilitätsfaktor (iSF) in mg/dl/U oder mmol/L/U** können wir den Korrekturfaktor in mg/dl oder mmol/L angeben. Mit mehrfachen Eingaben können unterschiedliche Faktoren bestimmt werden.

Mit **Kohlenhydrataktivität / Aufnahme Kohlenhydrate in Gramm/Stunde** können wir bestimmen, wieviel Gramm Kohlenhydrate pro Stunde verstoffwechselt werden. Das Plugin cob benötigt diese Eingaben.

Mit **Basalraten in Einheiten/Stunde** können wir Basalraten für bestimmte Zeitbereiche eingeben.

Am Ende das **Speichern** nicht vergessen.

2.7.2.3 Nahrungsmittel - Editor

Nightscout Nahrungsmittel Editor

Kategorie: (nichts) Unterkategorie: (nichts) Name:

Name	Portion	Einheit	Kohlenhydrate	GI [1-3]	Kategorie	Unterkategorie
Frühstück	0	g	48	2		
Mittagessen	0	g	48	2		

2

1

Name: Portion: Einheit: Kohlenhydrate: [g] GI: Kategorie: (nichts) Unterkategorie:

g Mittel

Erstelle neuen Datensatz Löschen

Speichern

Verborgene Verberge nach Gebrauch Kohlenhydrate: 48 g

Name	Portion	Kohlenhydrate	Portionen:
breakfast	0	48	1

Authentifikations Status:
Admin authorized ([Entfernen](#))

Im Nahrungsmittel - Editor können wir Mahlzeiten Mit Angabe von Bezeichnung , Kategorie und Kohlenhydraten. Daten sind für den **Bolus Kalkulator** optional wählbar. Über Ernährungsprogramme lassen sich die Daten über csv- oder json - Format in die **mongodb** übertragen.

1. Eingabe von Nahrungsmittel - Daten
2. Ändern, Löschen von Datensätzen

2.7.2.4 Administrator Werkzeuge

Administrator Werkzeuge

Subjects - People, Devices, etc.

Each subject will have a unique access token and 1 or more roles. Click on the access token to open a new view with the selected subject, this secret link can then be shared.

Name	Roles	Access Token	Notiz
✓ Viewer	status-only	viewer-5c76bb43d5d1d745	
✓ Viewer	status-only	viewer-b0363113c483b02e	

Add new Subject Database contains 2 subjects **1**

Roles - Groups of People, Devices, etc.

Each role will have a 1 or more permissions. The * permission is a wildcard, permissions are a hierarchy using : as a separator.

Name	Permissions	Notiz
✓ admin	*	[system default]
✓ careportal	apitreatments:create	[system default]
✓ denied	[none]	[system default]
✓ devicestatus-upload	apiddevicestatus:create	[system default]
✓ readable	***:read	[system default]
✓ status-only	apistatus:read	[system default]

Add new Role Database contains 6 roles **2**

Bereinige mongo status-datenbank

Lösche alle Dokumente der Gerätestatus-Sammlung
Diese Aufgabe entfernt alle Dokumente aus der Gerätestatus-Sammlung. Nützlich wenn der Uploader-Batteriestatus sich nicht aktualisiert.

Lösche alle Dokumente Datenbank enthält 500+ Einträge **3**

Entferne zukünftige Objekte aus mongo-datenbank

Finde und entferne zukünftige Behandlungen
Finde und entferne Behandlungen in der Zukunft.

Zeit	Ereignis-Typ	Blutglukose	Insulin	Kohlenhydrate	Eingabe durch	Notiz
Entferne Behandlungen in der Zukunft Datenbank enthält 0 zukünftige Einträge 4						

Finde und entferne Einträge in der Zukunft
Finde und entferne CGM Daten in der Zukunft, die vom Uploader mit falschem Datum/Uhrzeit erstellt wurden.

Entferne Einträge in der Zukunft Datenbank enthält 0 zukünftige Einträge

Authentifikations Status:
admin authorized ([Entfernen](#))

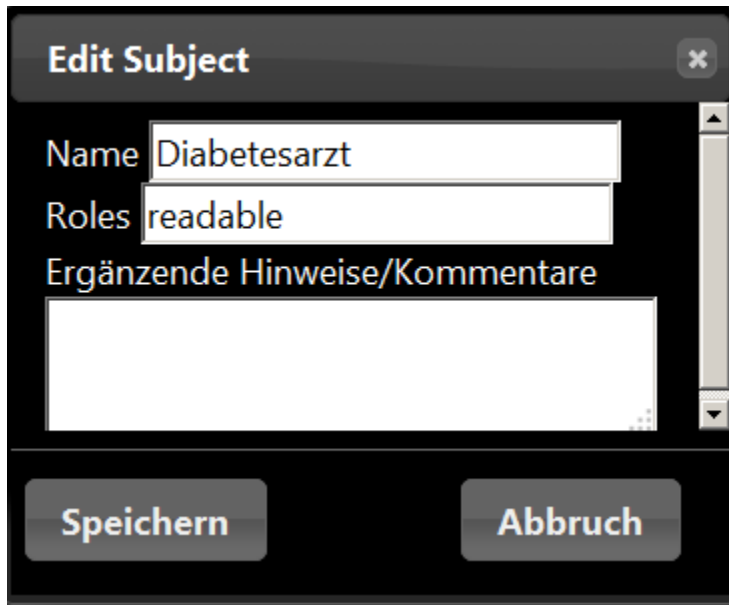
Mit den Administrator - Werkzeugen können wir Berechtigungen zum Zugriff auf die Nightscout - Website verwalten und Datensätze aus der mongo DB löschen.

2.7.2.4.1 Verwalten von Berechtigungen

Für Berechtigungen benötigt man **Subjekte** (Menschen, Institutionen, etc.), denen man eine Rolle zuweist. Für jedes Subjekt wird ein **Zugriffstoken** generiert, mit dem der Betrachter Zugriff auf die Website bekommt. Dieses Token muss man dem Betrachter mitteilen. Innerhalb der Rolle werden Zugriffsberechtigungen verteilt, das höchste Recht ist Vollzugriff, symbolisiert durch ein * - Symbol, das niedrigste ist **kein Zugriff**.

1. Anlegen von Subjekten
2. Anlegen von Rollen
3. Löschen von Dokumenten
4. Behandlungsdaten in der Zukunft

Anlegen von Subjekten:



Es gibt standardmäßig 6 Rollen, die in den meisten Fällen ausreichend sind. Hinter den Rollen sind die Zugriffsrechte gelistet.

Man kann eigene Rollen wie beim anlegen von Subjekten definieren und speichern.

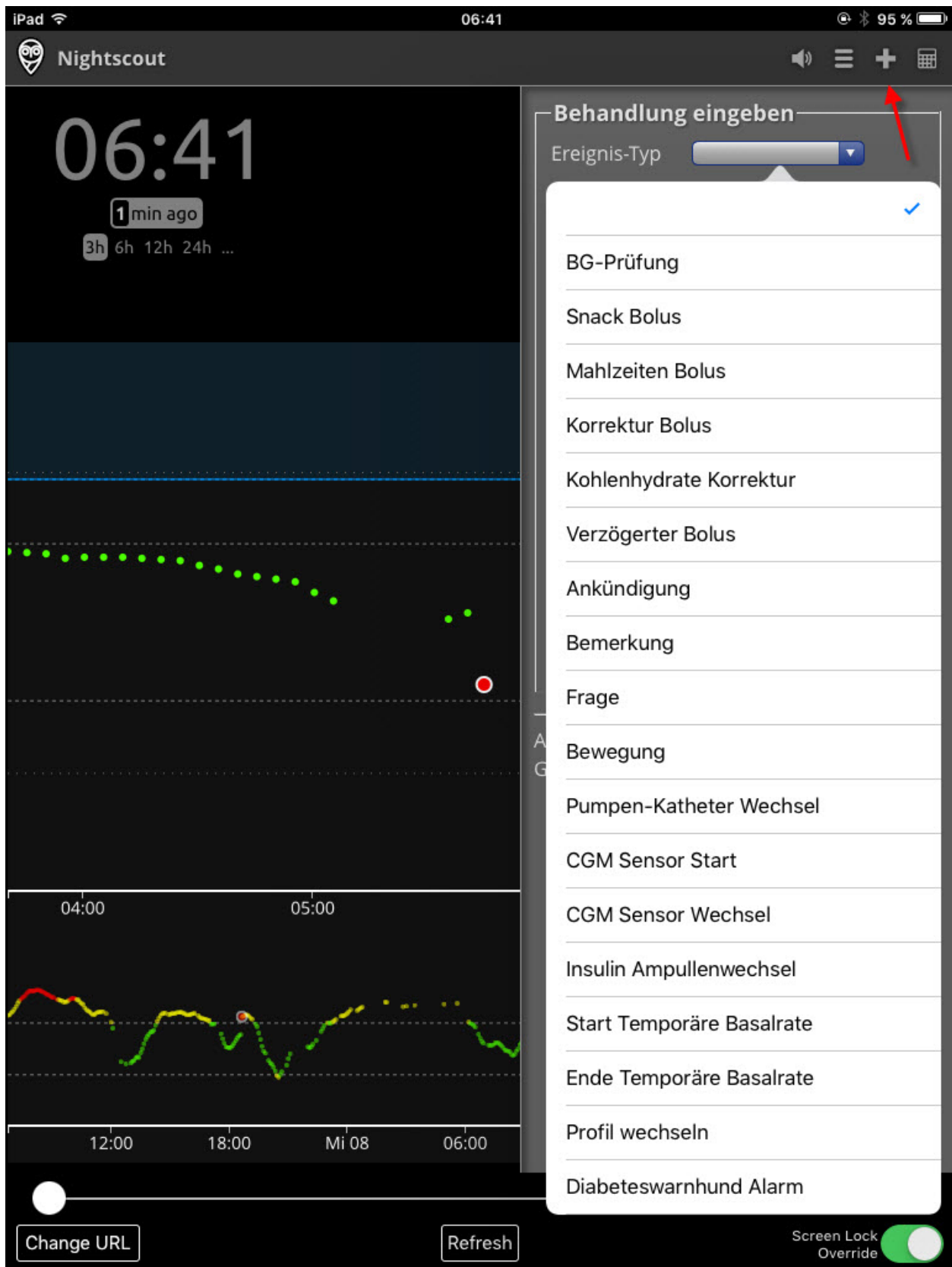
Es ist unter Punkt 3 möglich, die Gerätestatus - Sammlung (Collection in der mongo db) mit diesem Menüpunkt zu löschen.

Mit Punkt 4 kann man fälschlicherweise Behandlungen mit Datum in der Zukunft aus der Datenbank löschen.

2.7.3 Behandlungsportal

Wir beschäftigen uns hier genauer mit der Dateneingabe und starten mit dem Klick auf das + - Symbol. Der Ereignis-Typ ist ein DropDown - Menü mit Auswahlwerten. Beispielhaft wählen wir jetzt einen `Mahlzeiten Bolus` aus, um Insulinmenge und Kohlenhydrate (KH) hinzuzufügen.

Ereignis-Typ:



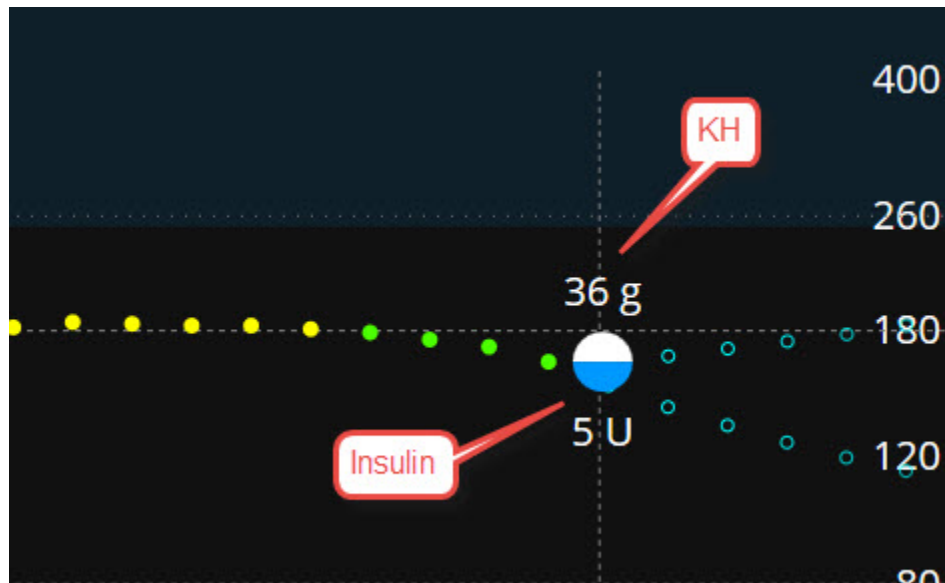
Die Auswahlfelder werden entsprechend der Auswahl angepasst.

Eingabe von KH und Insulineinheiten erfolgen jetzt. Wichtig ist auch noch die Eingabe der korrekten **Zeit**, falls man Daten nicht online eingeben kann, da diese zusammen mit den CGM - Werten grafisch dargestellt werden.

In dieser Maske können wir auch temporäre Basalraten mit Temp Basal Start/End eingeben. Diese werden anschließend grafisch dargestellt.

Nach dem Klick auf **Eingabe Senden** wird man noch aufgefordert, die eingaben zu bestätigen. Mit **OK** ist alles gespeichert.

Nun können wir uns alles grafisch ansehen:



Benachrichtigungen über Therapie - Behandlungsdaten

Wenn Benachrichtigungen über Behandlungsdaten aktiviert sind, werden Benachrichtigungen generiert wenn eine Eingabe im Careportal gemacht wurde. Mögliche Alarmeinstellungen werden kurz danach unterdrückt. Die Standard - Unterdrückungsdauer beträgt 10 Minuten. Dieses kann über Azure App.-Einstellungen geändert werden, in dem man die TREATMENTNOTIFY_SNOOZE_MINS Variable ändert.

2.7.4 Bolus Rechner

Der Bolus Rechner berechnet die abzugebende Insulinmenge für eine Mahlzeit. Wir erreichen das Menü durch Klick auf das Rechner - Symbol ganz oben rechts:

Bolus-Kalkulator

BG:

135

☒
☐
☐
☐
☐
☐

123

0.00

Schnellauswahl:

(nichts)

oder

Ergänzt aus Datenbank

☒ Kohlenhydrate:

36

g

6.00

☐ COB:

g

☒ IOB:

0.00

Weitere Korrektur:

0.00

Gerundet:

0.00

☒ Benötigtes Insulin:

6.00

Basalrate:

1.000

Kohlenhydrat Zeit:

Zeit in Minuten

Ergänzende Hinweise/Kommentare

Eingabe durch:

Ereignis Zeit:

☐ Jetzt
☒ Weitere

2016-06-02

11:45

RETRO MODUS

Formular absenden

Die KHS können entweder manuell oder über gespeicherte Mahlzeiten aus der Nahrungsmittel - Tabelle eingegeben werden.

Der Bolus Rechner greift auf Werte aus dem `Profil Editor` zu, genauer dem `ISF` und dem
mit `Formular` absenden werden die Daten gespeichert.

2.8 Installation ohne Cloud Service Provider

Nightscout lässt sich lokal installieren, so dass die Website auf einem eigenen Server, z.B. auf einem Rasberry PI liegt.

Windows

Folgender Link beschreibt die Installation und Konfiguration eines windows server 2012, IIS und Nightscout Website (englische Version):

[Nightscout on Windows Server](#)

2.9 Programmierbarkeit

Nightscout besteht aus einer **Applikations-** und einer **Datenbankschicht**.

Applikationsschicht

Nightscout ist eine Open Source Software. Der Software Code ist frei im Internet über [GitHub](#) einsehbar. Jeder, der Interesse hat und sich zutraut am Programmcode mitzuentwickeln kann dieses tun. Die Kommunikation verläuft über [Gitter Chat](#) in englischer Sprache.

Die Software wird auf einem Webserver gehostet. Sie wird über Webservices angesprochen, welche das API beschreibt.

Das **API** ist mit [Swagger](#) dokumentiert. Swagger ist ein API Dokumentationswerkzeug, welches auf die [Open API Initiative](#) aufbaut.

Wir können das Nightscout API mit der Adresse:

<https://meinwebsitename.azurewebsites.net/api-docs.html#/> aufrufen:

Nightscout API:

Nightscout API
Own your DData with the Nightscout API
AGPL 3

Debug Show/Hide List Operations Expand Operations

Entries Show/Hide List Operations Expand Operations

Method	Endpoint	Description
GET	/entries/{spec}	All Entries matching query
GET	/slice/{storage}/{field}/{type}/{prefix}/{regex}	All Entries matching query
GET	/echo/{storage}/{spec}	View generated Mongo Query object
GET	/times/echo/{prefix}/{regex}	Echo the query object to be used.
GET	/times/{prefix}/{regex}	All Entries matching query
DELETE	/entries	Delete entries matching query.
GET	/entries	All Entries matching query
POST	/entries	Add new entries.

Profile Show/Hide List Operations Expand Operations

Status Show/Hide List Operations Expand Operations

Treatments Show/Hide List Operations Expand Operations

[BASE URL: /api/v1 , API VERSION: 0.8.0]

ERROR { }

Sollten die Details nicht automatisch dargestellt werden, so bitte den Wert

<https://meinwebsitename.azurewebsites.net/swagger.yaml> eintragen und den Button Explore klicken.

IFTTT

Der Internet - Event Dienst IFTTT (If This Than That) verknüpft verschiedene Webanwendungen, um Daten auszutauschen. So können Benachrichtigungen, welche über Nightscout erzeugt werden, z.B. als Email weitergeleitet werden.

Daneben gibt es weitere Möglichkeiten, z.B. über Azure Webjobs Daten zwischen Nightscout und anderen Tagebüchern, wie Sport- oder Ernährungstagebüchern zu synchronisieren, sofern diese ein dokumentiertes API besitzen.

Datenbankschicht

Als Nutzer der mongodb hat man Vollzugriff auf seine Daten. Über Tools, wie in Kapitel 2.5 beschrieben, kann man sich die CGM- und Behandlungsdaten sichern (im json- und csv Format). Es sind auch Uploads über die json und csv Formate möglich.

Weiter gehts mit der Anbindung von Smartwatches.

3.1 Android Wear

3.2 Pebble

https://apps.getpebble.com/en_US/application/568fb97705f633b362000045

3.3 Apple Watch

3.4 Garmin

CGM-Watch

Kompatible Geräte:

- ForeAthlete 920XTJ
- Forerunner® 920XT
- epix™
- fēnix™ 3
- fēnix™ 3 HR
- vívoactive™

CGM-Run

Kompatible Geräte:

- ForeAthlete® 920XTJ
- Forerunner® 920XT

- epix™
- vívoactive®

CGM-Hike

Kompatible Geräte:

- ForeAthlete® 920XTJ
- Forerunner® 920XT
- epix™
- vívoactive®

4.1 Nightscout Seite bleibt einfach weiß:

- API_SECRET mindestens 12 Zeichen.
- Keine Verbindung zur MongoDB. Mögliche Ursachen:
 - – Fehler in der URI.
 - – Falschen Datenbank Benutzer angegeben.
 - – Datenbank Benutzer vergessen anzulegen.

4.2 Eingabe von Treatments (Insulin, Kohlenhydrate usw.) wird nicht auf der Hauptseite angezeigt:

- Mit einem anderen Browser gegenprüfen. Ggf. Adblocker /script Blocker deaktivieren.
- Überprüfen, ob das Gerät (Browser) authentifiziert ist.
- In Nightscout “zurücksetzen und Voreinstellungen verwenden” drücken.

4.3 Eingegebene Werte erscheinen in der Nightscout Verlaufsgrafik an einem anderen Tag:

- Bitte unter Profil-Einstellungen die Zeitzone überprüfen und ggf. ändern.

4.4 Die Website reagiert sehr langsam, wenn sie längere Zeit nicht verwendet wurde:

Wenn die Website längere Zeit nicht verwendet wird, z.B. wenn die Daten nachts direkt zur Mongo-Datenbank hochgeladen werden und keine Uhr Daten periodisch abfragt, baut Azure Verbindungen ab. Diese wieder herzustellen, braucht eine gewisse Zeit. Abhilfe schaffen Dienste, die die Website periodisch (z.B. alle 5 Minuten einmal) abrufen. Ein solcher Dienst ist <https://uptimerobot.com/>.

Links:

[deutsche Gitter Chat Gruppe zu OpenAPS, DIY, Nightscout, xDrip, CGM, Closed Loop](#)
[Nightscout Deutschland Facebook-Gruppe](#)

Flyer:

[Download Flyer Vorderseite](#)
[Download Flyer Rückseite](#)

Verlängerung der Tragezeit:

[Download Verlängerung der Tragezeit des Sensors](#)

Schablonen:

[Download Dexcom Pflaster Schablone](#)
[xDrip 1-Platinen Layout](#)

#WeAreNotWaiting Ausdruck der Empörung über die Medizintechnik, die aktuelle technische Möglichkeiten nicht ausschöpft, um betroffene Typ 1 Diabetiker zu unterstützen. Wir warten nicht darauf, dass uns geholfen wird, wir helfen uns selbst.

5,5 5,5 mmol/l ist die Entsprechung von 100mg/dl (ein gerne gefeierter Zielwert)

API SECRET Das API SECRET - Application Programming Interface - SECRET ist ein Schlüssel zur sicheren Übertragung von Daten über das Internet. Es besteht aus numerischen und alphanumerischen Zeichen und identifiziert den Absender. Näheres siehe: (<https://de.wikipedia.org/wiki/Programmierschnittstelle>)

APS (engl.: Artificial Pancreas System, APS) CGM ist ferner integraler Bestandteil einer dauerhaft funktionierenden künstlichen Bauchspeicheldrüse, nämlich einer vom Blutzucker über einen Algorithmus geregelten Insulinpumpe. Ein solcher Regelkreis besteht grundsätzlich aus einem kontinuierlich messenden Glucosesensor und einem Regler, der eine Insulinpumpe steuert.

AR2 (engl.: Auto Regression = AR2) ein mathematisches Modell zur Vorausberechnung von Werten unter Berücksichtigung des bisherigen Verlaufs. Näheres siehe: https://en.wikipedia.org/wiki/Autoregressive_model#AR.282.29

bwp (engl.: Bolus Wizard Preview = Bolus Vorschau Assistent, bwp) Der Bolus Vorschau Assistent kann anhand von Wirkungszeit des Insulins (iob) und Zeitpunkt der Injektion oder Abgabe von Insulin durch die Pumpe eine Vorschau berechnen. Diese kann dafür genutzt werden, um Entscheidungen über mögliche Korrekturen zu erleichtern. Dieses können Bolus - Korrekturen bei erhöhten BZ Werten oder Kohlenhydrat - Korrekturen bei zu niedrigen Werten sein. Voraussetzung hierfür sind die genaue Dokumentation von Insulininjektionen und Wirkdauer - über Careportal und den Nightscout Profil Editor. Weitere Erläuterungen in englischer Sprache findet man hier: <http://www.nightscout.info/wiki/welcome/website-features/funnel-cake-0-8-features/bwp-bolus-wizard-preview>

cage Insulinpumpen Katheter Setzzeit (engl.: Cannula Age, Cage) Zeigt die verstrichene Zeit an seit dem letzten Infusionsset - Setzen oder -Wechseln.

CGM Die Kontinuierliche Glucosemessung (engl.: Continuous Glucose Monitoring, CGM) wird in der Regel bei Menschen mit Diabetes mellitus eingesetzt, um die Therapie besser steuern zu können. Im Gegensatz zur herkömmlichen Blutzuckermessung wird nicht der Glucosegehalt in einem dafür gewonnenen Blutstropfen gemessen, sondern im Gewebe des Körpers. Veränderungen treten dort meist mit einer kurzen zeitlichen Verzögerung auf. Kommerzielle CGM-Systeme bestehen derzeit aus einem Nadelsensor, einem Transmitter

(Übertragung) und einem Empfangsgerät, das alle Werte speichert und auf dem der Gewebeglucosewert zusammen mit einem Trend auf dem Display angezeigt werden kann.

cob Kohlenhydrate an Bord (engl.: Carbs On Board, cob) ist die Menge an Kohlenhydraten aus vorangegangenen Mahlzeiten, Snacks und Kohlenhydrat - Korrekturen. Die Zeitdauer, wie lange aufgenommene Kohlenhydrate aktiv sind, wird beeinflusst durch die Menge an Kohlenhydraten, die gegessen wurden und die Kohlenhydratart. (Kohlenhydrate in Schwarzbrot werden langsamer aufgenommen als Kohlenhydrate in Cola, wirken dafür aber länger).

DIY Do-It-Yourself (engl.) selbstgemacht. DIY-Projekt bedeutet in unserem Fall, dass es keine Firma gibt, von der man ein fertiges, funktionierendes System kaufen kann. Die Elemente des Systems sind von Teilnehmern des Projektes entwickelt und zusammengesetzt und werden dir kostenlos zur Verfügung gestellt. DIY bedeutet für dich, dass du deinen Teil dazu beitragen musst, damit das System für dich funktioniert. Wir hoffen, dass DIY dich so sehr begeistert wird wie uns - sodass du wiederum irgendwann einen kleinen Teil dazu beitragen kannst, dass das System für alle besser wird.

iob Insulin an Bord (engl.: Insulin On Board, iob) ist das aktive, schnellwirksame Bolus Insulin , welches nach einer Injektion oder Abgabe durch eine Insulinpumpe noch im Körper wirkt. Es hilft zu entscheiden, ob mehr Insulin oder mehr Kohlenhydrate benötigt werden, um Blutzuckerspitzen oder Hypoglykämien zu verhindern. Die Wirkdauer des Insulins ist den Herstellerangaben zu entnehmen.

json (engl.: = Java Script Object Notation) ist ein einfaches Datenformat zum Austausch und zur Speicherung von Daten. Das Datenformat wird u.a. bei Abfragen über Webservices und Speicherung von Daten in Dokumentendatenbanken wie mongodb eingesetzt. Näheres siehe hier: https://de.wikipedia.org/wiki/JavaScript_Object_Notation

Nightscout Was ist das Nightscout Projekt? (NS in der Cloud) ist ein Open-Source-DIY-Projekt, das in Echtzeit einen Zugang zu einer persönlichen Website erstellt, auf der man seine CGM-Daten über das Internet sehen und auswerten kann. Entwickelt wurde es ursprünglich von Eltern, die die Werte ihrer Kinder aus der Ferne beobachten wollen. Nightscout (frei übersetzt Nachtwache - wie die wachsame Eule, die über dich wacht, auch während du schläfst). Mittlerweile nutzen es auch Erwachsene, um die Daten weiter zu nutzen. Die Daten können nicht nur in einem Internetbrowser angezeigt werden, sondern von dort aus auch über Apps und Widgets auf dem Smartphone, Tablets und auf Smartwatches.

NS Siehe Nightscout

Open Source Software Open Source Software wird als Begriff für Software verwendet, deren Quelltext offenliegt und frei verfügbar ist. Die Open Source Bewegung entstand bereits in der 60/70er Jahren des letzten Jahrhunderts. Eine sehr hilfreiche Erklärung findet man hier: https://de.wikipedia.org/wiki/Open_Source

Plugin Plugins sind Software - Erweiterungen oder auch Software - Module, mit welchen man den Funktionsumfang einer Software - wie hier die Nightscout Website - konfigurieren kann. Eine detaillierte Beschreibung gibt es hier: <https://de.wikipedia.org/wiki/Plug-in>

Symbols

#WeAreNotWaiting, [127](#)
5, 5, [127](#)

A

API SECRET, [127](#)
APS, [127](#)
AR2, [127](#)

B

bwp, [127](#)

C

cage, [127](#)
CGM, [127](#)
cob, [128](#)

D

DIY, [128](#)

I

iob, [128](#)

J

json, [128](#)

N

Nightscout, [128](#)
NS, [128](#)

O

Open Source Software, [128](#)

P

Plugin, [128](#)