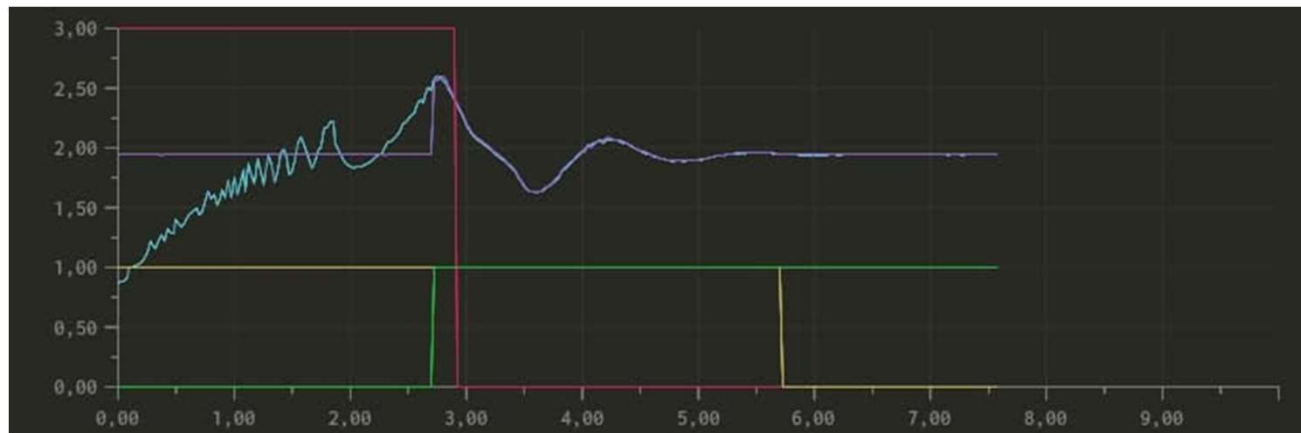




CLOUD EDITION

ALTERNATIVE SOFTWARE FÜR DEN GPSDO VON DL7UKM UND DL4ZAO



SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Motivation**

Warum überhaupt noch ein GPSDO und weshalb eine weitere Software?

• Motivation I

- Als Maker will ich meine Sachen möglichst gut verstehen und weitestgehend selbst machen
- Bekanntes anwenden, dazu lernen und Neues schaffen
- Kosten sparen ohne auf Funktionen/Features/Qualität/... zu verzichten
- Erfahrungen/Ergebnisse/Produkte an die Gemeinde zurück geben, von der man selbst partizipiert

• **Motivation I**

- Als Maker will ich meine Sachen möglichst gut verstehen und weitestgehend selbst machen
- Bekanntes anwenden, dazu lernen und Neues schaffen
- Kosten sparen ohne auf Funktionen/Features/Qualität/... zu verzichten
- Erfahrungen/Ergebnisse/Produkte an die Gemeinde zurück geben, von der man selbst partizipiert

• **Motivation I**

- Als Maker will ich meine Sachen möglichst gut verstehen und weitestgehend selbst machen
- Bekanntes anwenden, dazu lernen und Neues schaffen
- Kosten sparen ohne auf Funktionen/Features/Qualität/... zu verzichten
- Erfahrungen/Ergebnisse/Produkte an die Gemeinde zurück geben, von der man selbst partizipiert

• Motivation I

- Als Maker will ich meine Sachen möglichst gut verstehen und weitestgehend selbst machen
- Bekanntes anwenden, dazu lernen und Neues schaffen
- Kosten sparen ohne auf Funktionen/Features/Qualität/... zu verzichten
- Erfahrungen/Ergebnisse/Produkte an die Gemeinde zurück geben, von der man selbst partizipiert

• **Motivation I**

- Als Maker will ich meine Sachen möglichst gut verstehen und weitestgehend selbst machen
- Bekanntes anwenden, dazu lernen und Neues schaffen
- Kosten sparen ohne auf Funktionen/Features/Qualität/... zu verzichten
- Erfahrungen/Ergebnisse/Produkte an die Gemeinde zurück geben, von der man selbst partizipiert

- **Motivation II**

Warum überhaupt noch ein GPSDO und weshalb eine weitere Software?

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✎ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✎ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✎ ist einfach nachzubauen (*THT*)
- ✎ ist genau und stabil (*siehe Baumappe von DL4ZAO www.dl4zao.de*)
- ✎ beheimatet einen leicht zu programmierbaren μ Controller (*Atmega 328P in Form eines Arduino Nano*)

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✎ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✎ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✎ ist einfach nachzubauen (*THT*)
- ✎ ist genau und stabil (*siehe Baumappte von DL4ZAO www.dl4zao.de*)
- ✎ beheimatet einen leicht zu programmierbaren μ Controller (*Atmega 328P in Form eines Aduino Nano*)

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✎ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✎ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✎ ist einfach nachzubauen (*THT*)
- ✎ ist genau und stabil (*siehe Baumappte von DL4ZAO www.dl4zao.de*)
- ✎ beheimatet einen leicht zu programmierbaren μ Controller (*Atmega 328P in Form eines Adruino Nano*)

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✎ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✎ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✎ ist einfach nachzubauen (*THT*)
- ✎ ist genau und stabil (*siehe Baumappte von DL4ZAO www.dl4zao.de*)
- ✎ beheimatet einen leicht zu programmierbaren μ Controller
(*Atmega 328P in Form eines Aduino Nano*)

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✚ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✚ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✚ ist einfach nachzubauen (*THT*)
- ✚ ist genau und stabil (*siehe Baumappte von DL4ZAO www.dl4zao.de*)
- ✚ beheimatet einen leicht zu programmierbaren μ Controller
(*Atmega 328P in Form eines Aduino Nano*)

• Motivation II

Der GPSDO von DL7UKM und DL4ZAO:

- ✦ ist kostengünstig (*ca. 75,- bis 100,-€*)
- ✦ besteht aus leicht zu beschaffenden Komponenten (*Reichelt, ebay, ...*)
- ✦ ist einfach nachzubauen (*THT*)
- ✦ ist genau und stabil (*siehe Baumappte von DL4ZAO www.dl4zao.de*)
- ✦ beheimatet einen leicht zu programmierbaren μ Controller (*Atmega 328P in Form eines Aduino Nano*)

- **Motivation III**

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



• Motivation III

Der GPSDO von DL7UKM und DL4ZAO:
☒ hat noch **Potential in der Firmware**

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure for 'GPSDO' with folders like 'pio', '.vscode', 'backlog', 'Case', 'docs', 'include', 'src', and files like 'declare.h', 'README', 'GPSDO.h', 'GPSDO_Display', 'GPSDO_EEPROM', 'GPSDO_FHO', 'GPSDO_Morse', 'GPSDO_Ublox', 'OneWire', 'QTH_Locator', 'Temperature', 'TinyGPSPlus', and 'platformio.ini'. The main editor area shows the 'main.cpp' file with C++ code for a GPSDO. The code includes comments in German and functions like 'setup()', 'setupPins()', 'readCapacitivePin()', 'ubloxConfigure()', 'initSensors()', 'setHoldOverOnOrOff()', and 'void setup()'. The 'void setup()' function calls 'setSerialBridgeMode()', 'printVersion()', 'setupPins()', 'initDisplay()', and 'test(0x01)'. The bottom panel shows the 'TERMINAL' output, which displays the build process for 'TinyGPSPlus 1.0.2', 'Temperature', and 'OneWire 2.3.5'. It shows the compilation of 'main.cpp.o', linking of 'firmware.elf', and the final 'firmware.hex' file. The build was successful, taking 2.94 seconds. The status bar at the bottom indicates 'Ln 713, Col 16' and 'PlatformIO'.

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware besorgen und aufspielen**
- Entwicklungsumgebung einrichten und nutzen
- Firmware modifizieren und ergänzen

- **Firmware besorgen und aufspielen**
- **Entwicklungsumgebung einrichten und nutzen**
- Firmware modifizieren und ergänzen

- **Firmware besorgen und aufspielen**
- **Entwicklungsumgebung einrichten und nutzen**
- **Firmware modifizieren und ergänzen**

- **Firmware besorgen und aufspielen**

GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Entwicklungsumgebung einrichten und nutzen**

Visual Studio Code (PlatformIO)

Sourcetree und GitLab

- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)
Sourcetree und GitLab

- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)
Sourcetree und GitLab

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen
#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen
Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware II**

Setuproutine

Loop

Displaydetect/GUI-Aufbau

Temperatursensorerkennung

Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw, 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw, 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw, 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw, 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware V**

- **Firmware V**
Software optimieren mit dem Code-Inspektor

- Sicher Verpackt (eigenes Gehäuse für den 3D-Druck entwerfen)
- Externes Display -I2C -über kurz oder lang
- Ausblick
- Diskussion

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien



SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien



SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND

- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien



SOFTWARE FÜR EINEN GPSDO

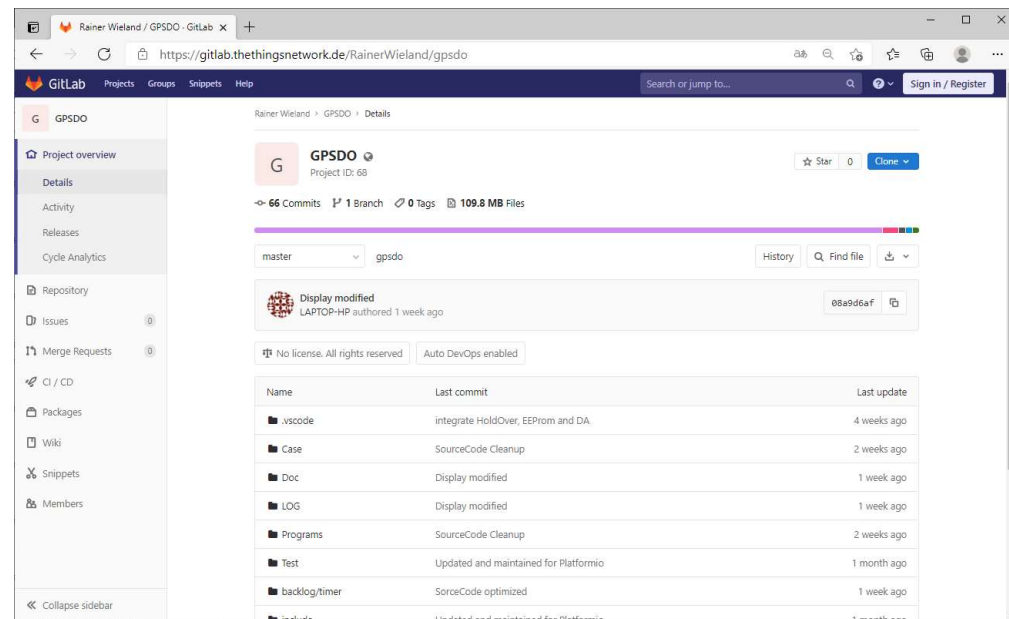
RAINER WIELAND



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien

[Rainer Wieland / GPSDO · GitLab \(thethingsnetwork.de\)](https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo)

<https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo>



SOFTWARE FÜR EINEN GPSDO

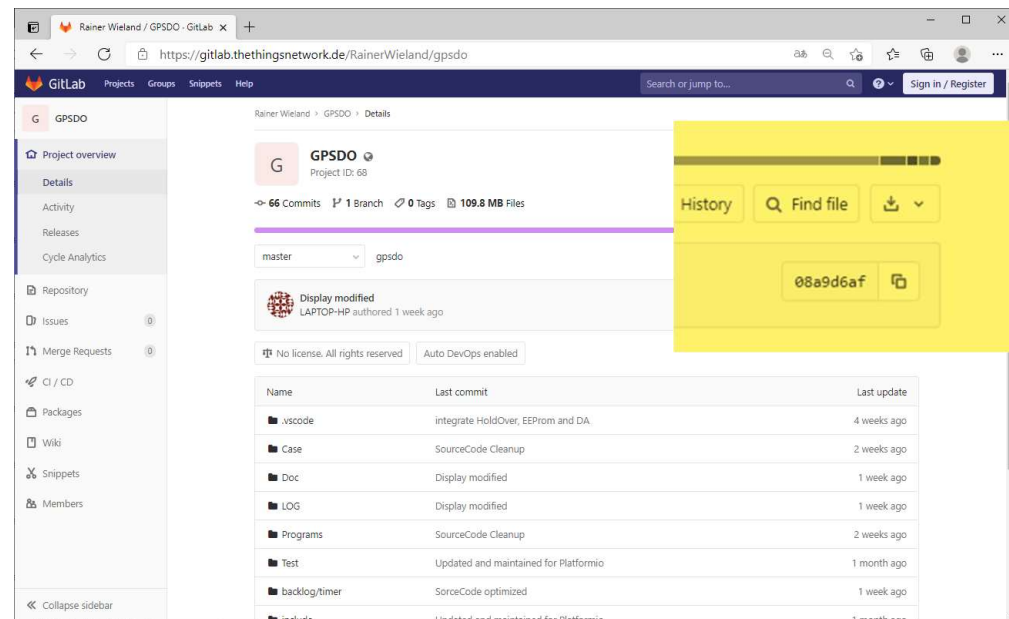
RAINER WIELAND



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien

[Rainer Wieland / GPSDO · GitLab \(thethingsnetwork.de\)](https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo)

<https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo>



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

SOFTWARE FÜR EINEN GPSDO

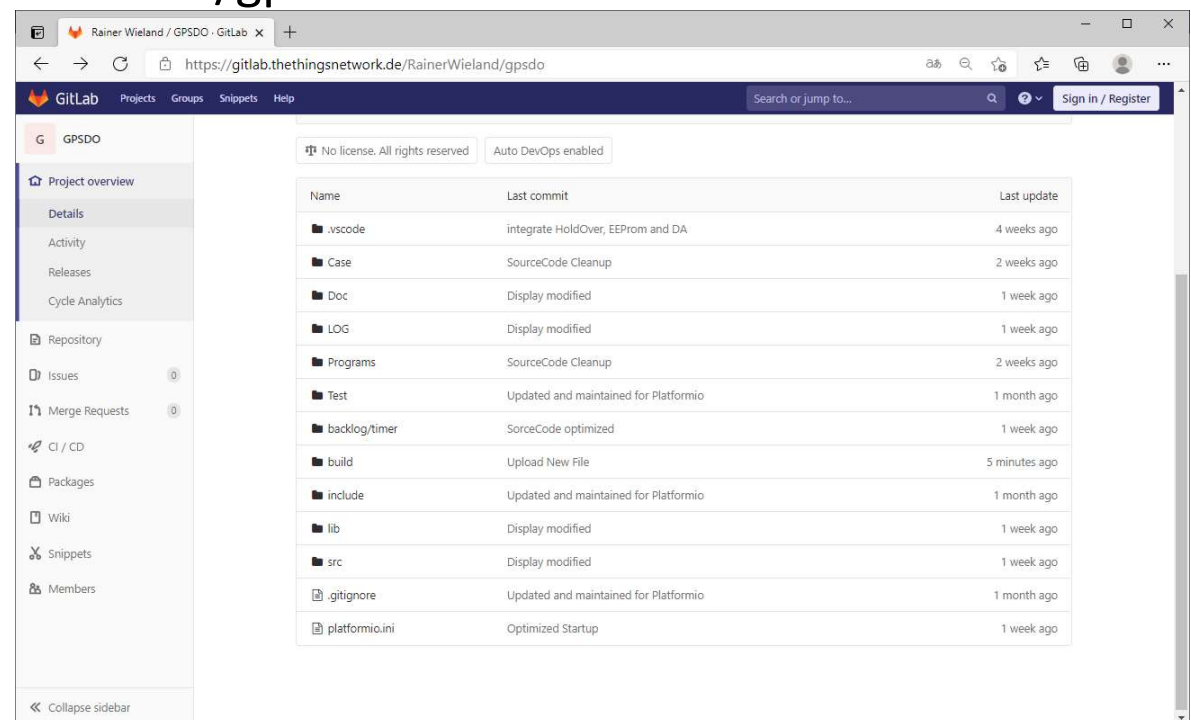
RAINER WIELAND



- **Firmware besorgen und aufspielen**
die Binärdatei im richtigen Format

[Rainer Wieland / GPSDO · GitLab \(thethingsnetwork.de\)](https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo)

<https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo>



SOFTWARE FÜR EINEN GPSDO

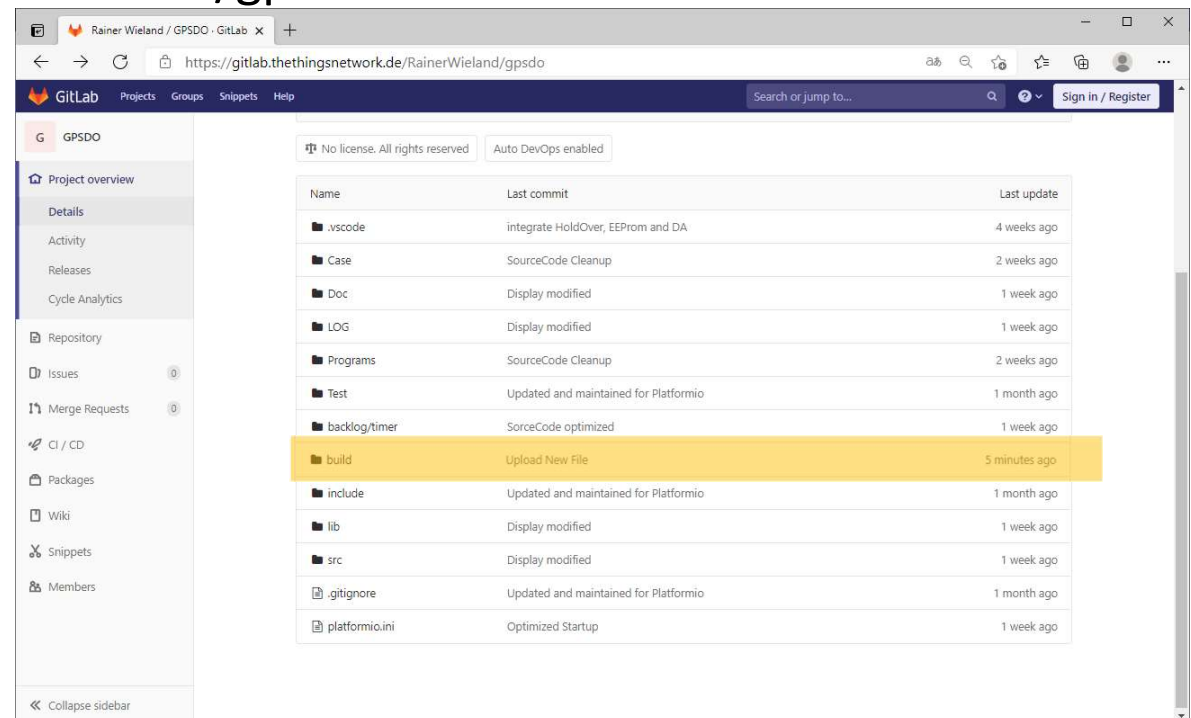
RAINER WIELAND



- **Firmware besorgen und aufspielen**
die Binärdatei im richtigen Format

[Rainer Wieland / GPSDO · GitLab \(thethingsnetwork.de\)](https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo)

<https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo>



SOFTWARE FÜR EINEN GPSDO

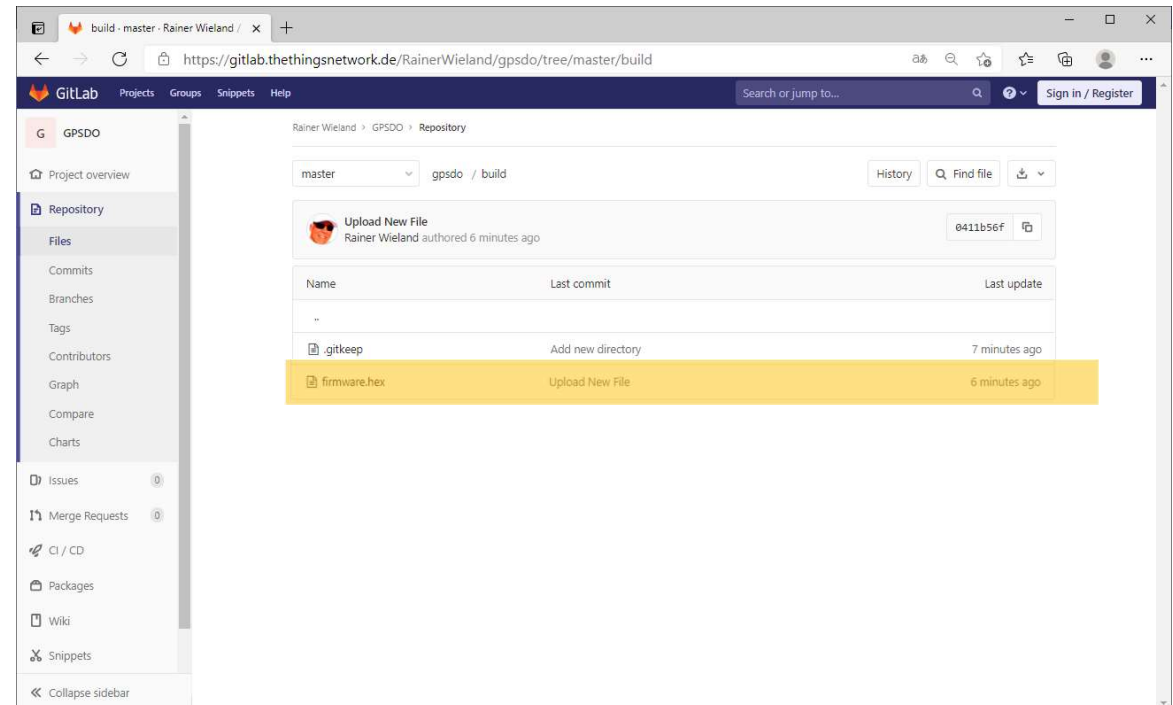
RAINER WIELAND



- **Firmware besorgen und aufspielen**
die Binärdatei im richtigen Format

[Rainer Wieland / GPSDO · GitLab \(thethingsnetwork.de\)](https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo)

<https://gitlab.thethingsnetwork.de/RainerWieland/gpsdo>



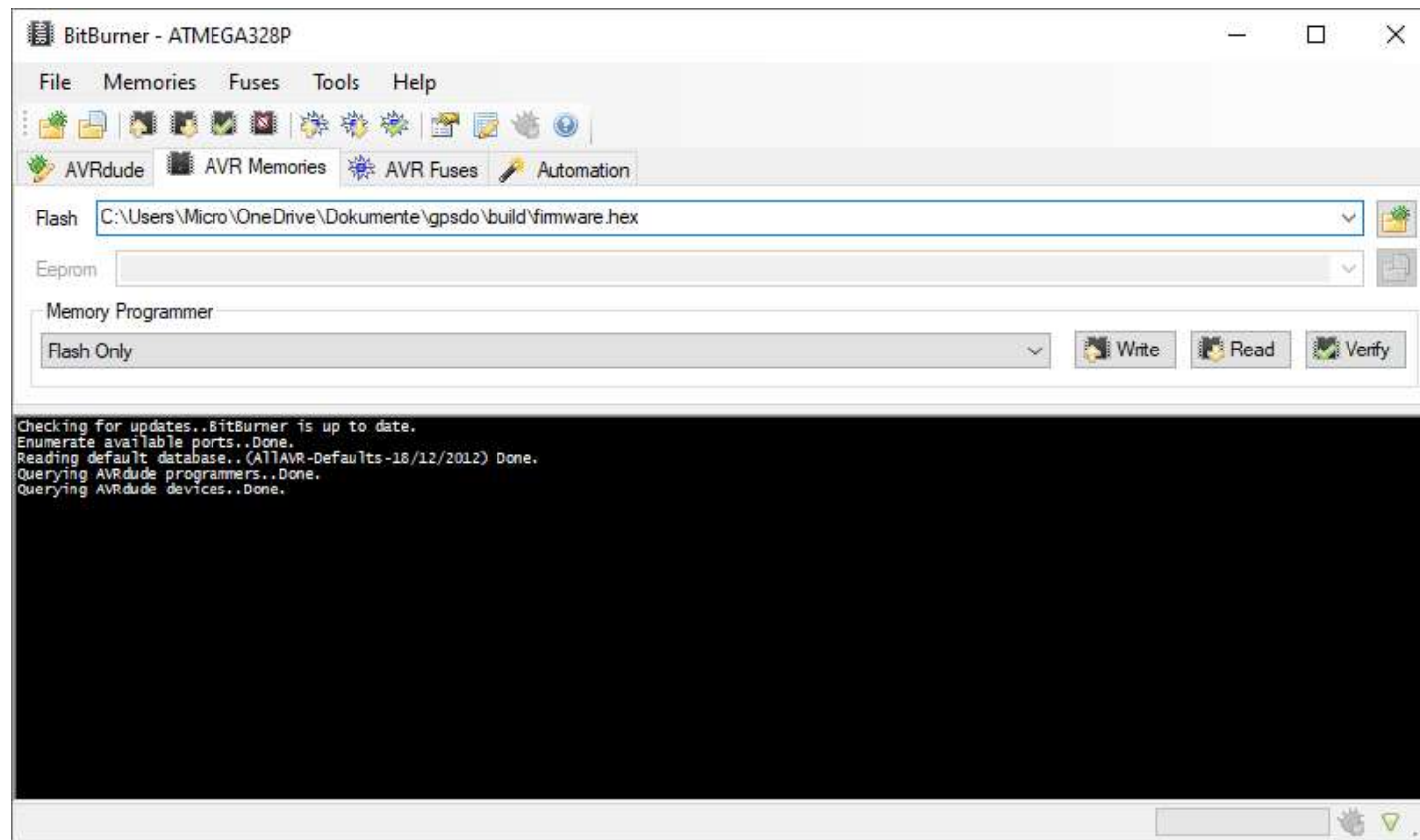
- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware besorgen und aufspielen**
das Hexfile in den Nano laden

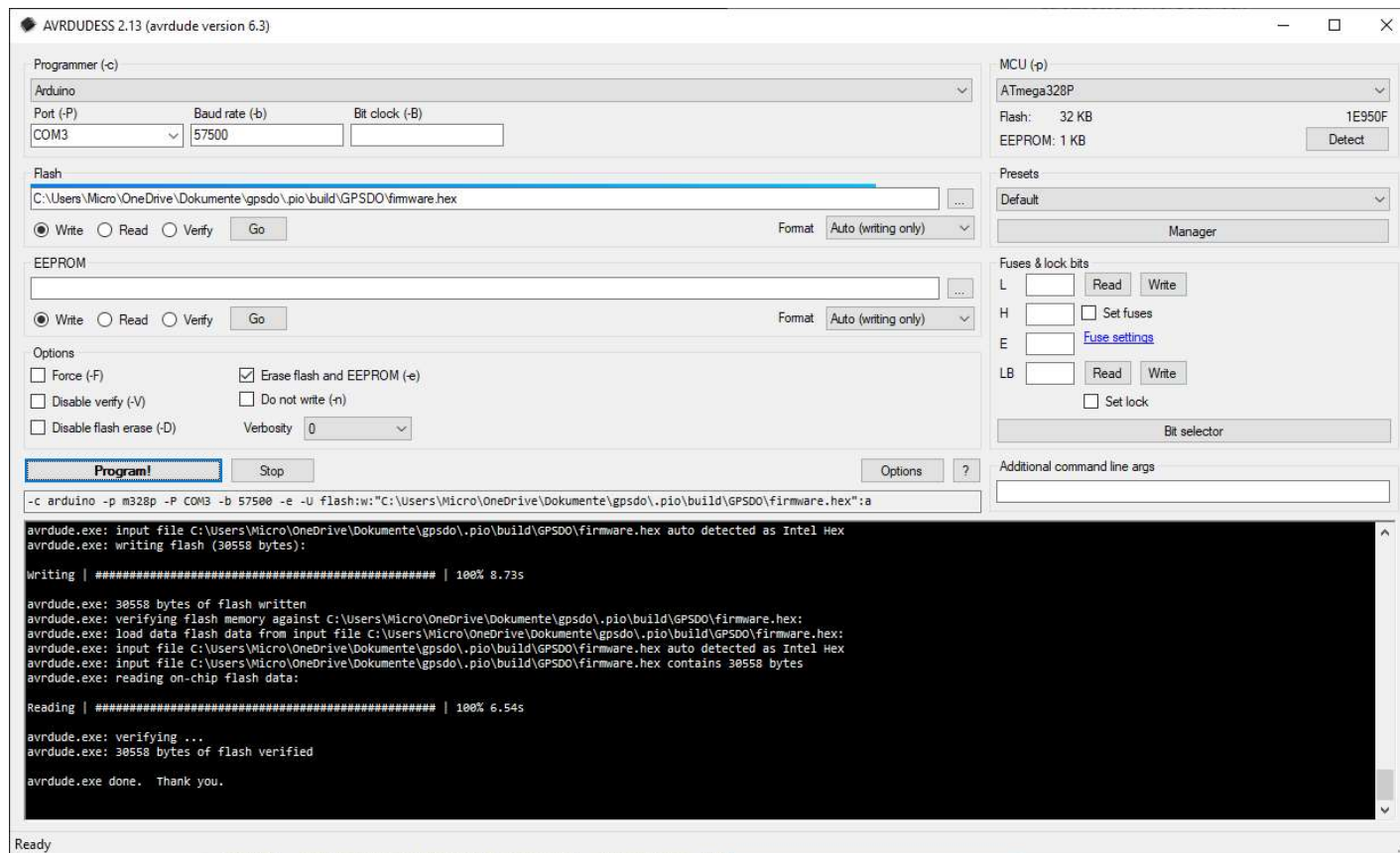


SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND

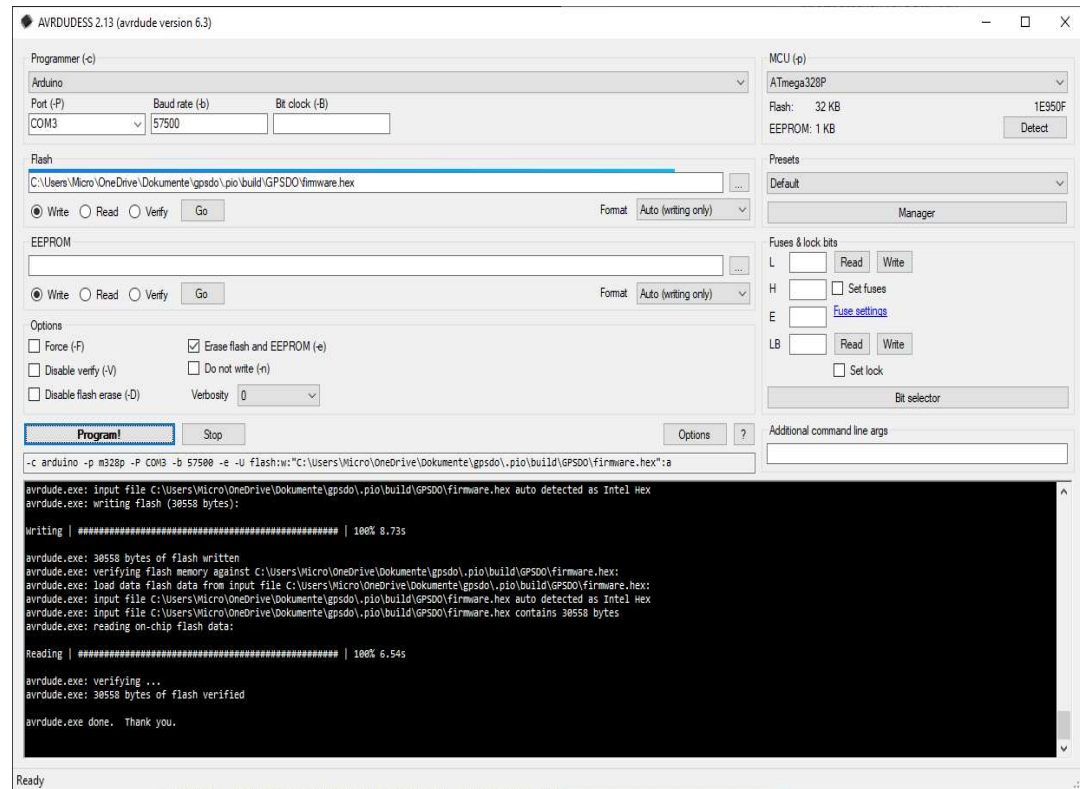


- **Firmware besorgen und aufspielen**
das Hexfile in den Nano laden



• Firmware besorgen und aufspielen das Hexfile in den Nano laden

- AVRDUDESS besorgen
<https://blog.zakkemble.net/avrdude-a-gui-for-avrdude/>
- Software auspacken und starten
- ATmega328P einstellen
- als Programmer Arduino auswählen
- richtige com-Schnittstelle auswählen
- Baudrate setzen (max. 57500Bd)
- Unter Flash das entsprechende Hex-File (firmware.hex) auswählen/eintragen
- Erase Flash and EEPROM anwählen
- **Programm!** Starten und warten bis der abgeschlossene Vorgang angezeigt wird.



- **Firmware besorgen und aufspielen**
GitLab, die Quelle aller Dateien
die Binärdatei im richtigen Format
das Hexfile in den Nano laden

- **Entwicklungsumgebung einrichten und nutzen**

Visual Studio Code (PlatformIO)

Sourcetree und GitLab

- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)
Sourcetree und GitLab

- **Entwicklungsumgebung einrichten und nutzen**

Visual Studio Code (PlatformIO)

Sourcetree und GitLab

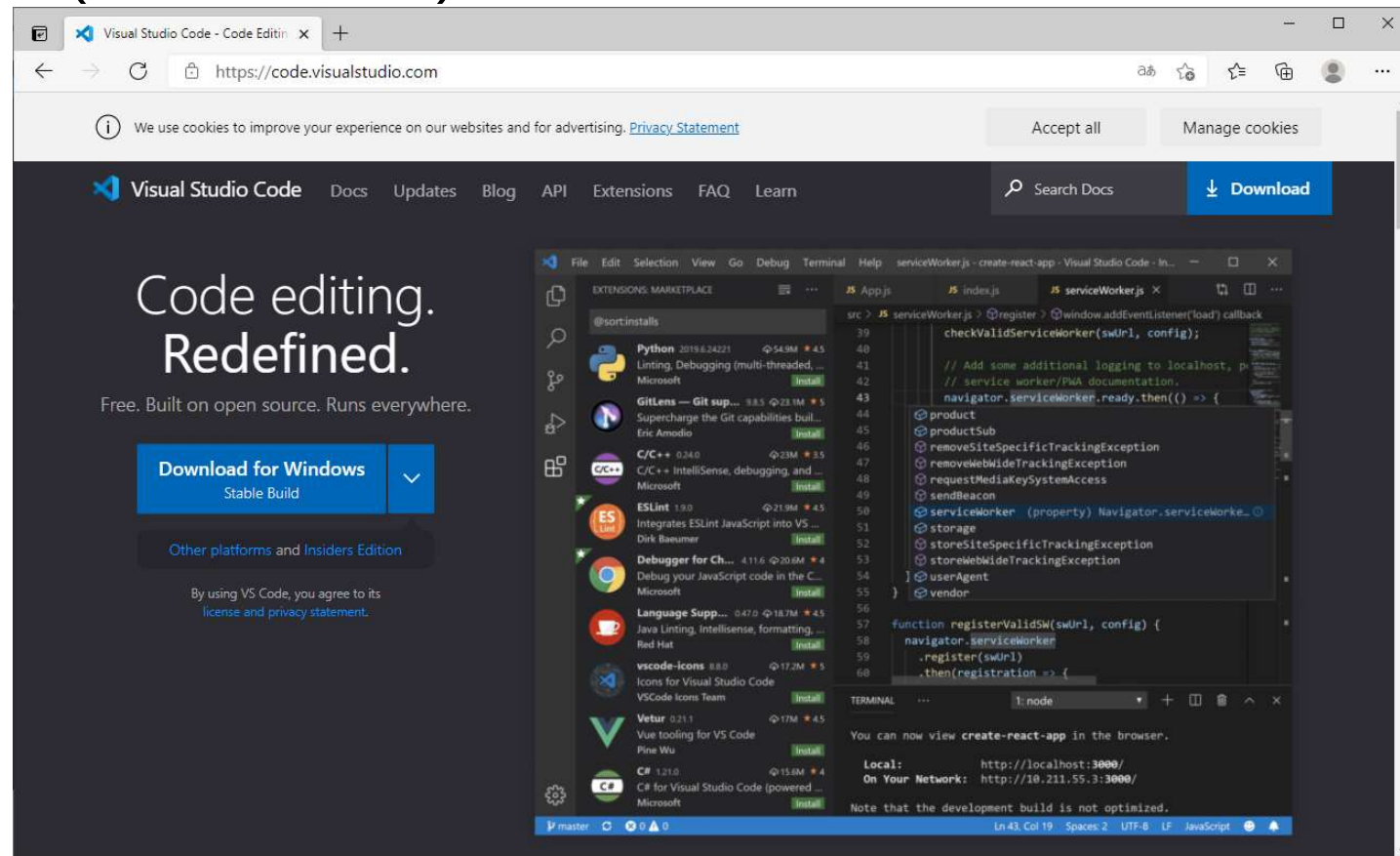
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)

<https://code.visualstudio.com/>



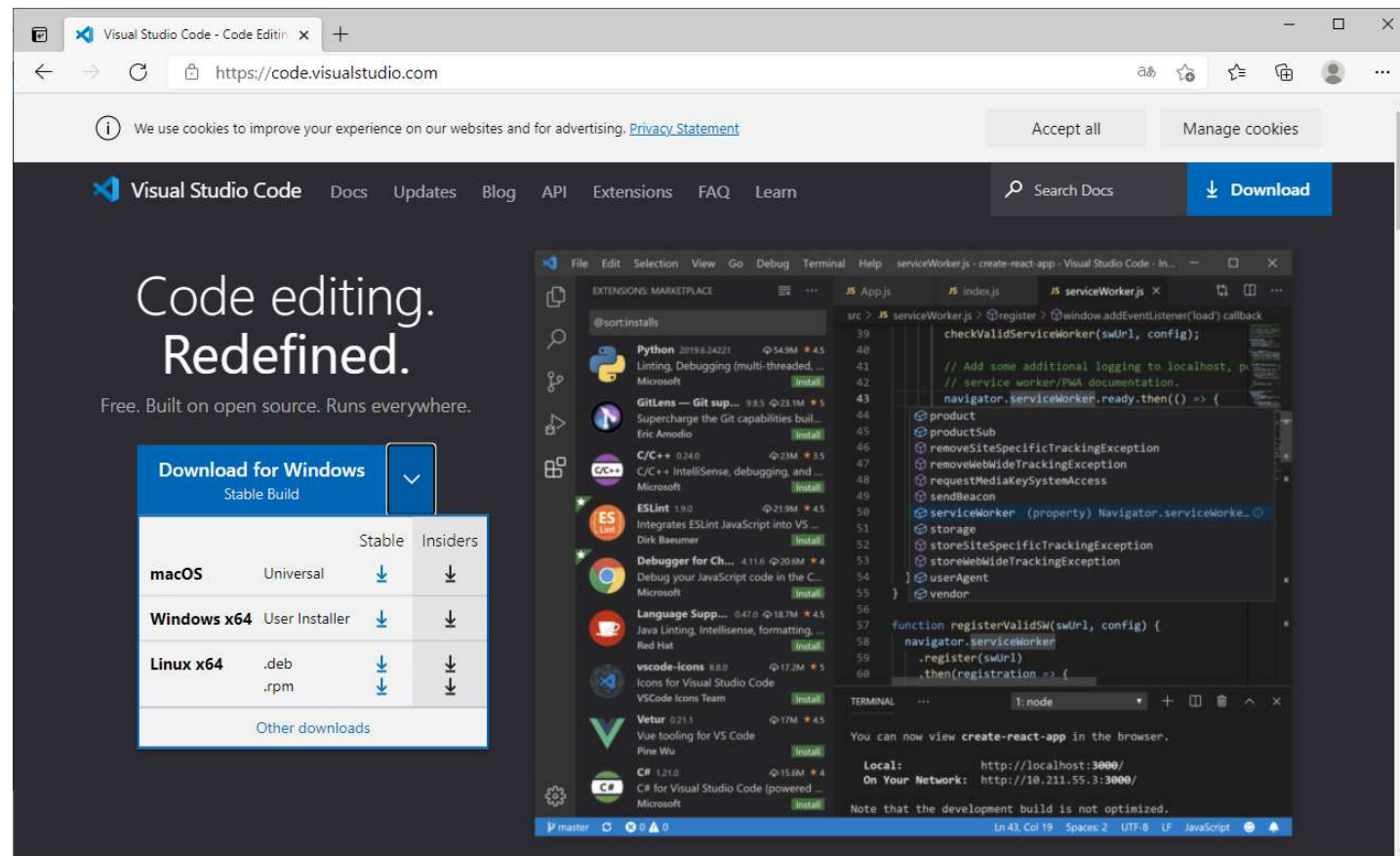
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)

<https://code.visualstudio.com/>



- **Entwicklungsumgebung einrichten und nutzen**
Visual Studio Code (PlatformIO)

- **Entwicklungsumgebung einrichten und nutzen**
 - Visual Studio Code (PlatformIO)
 - Sourcetree und GitLab

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Entwicklungsumgebung einrichten und nutzen**
Sourcetree und GitLab

[Sourcetree | Free Git GUI for Mac and Windows \(sourcetreeapp.com\)](https://www.sourcetreeapp.com/)

<https://www.sourcetreeapp.com/>

The screenshot shows the Sourcetree website in a web browser. The website has a clean, modern design with a light blue and white color scheme. The main heading reads "Simplicity and power in a beautiful Git GUI". Below this, there is a "Download for Windows" button and a link "Also available for Mac OS X". To the right, there is a preview of the Sourcetree application interface, which shows a commit history graph and a list of commits with their authors, descriptions, and dates. The application interface is dark-themed and includes a sidebar with navigation options like "Workspace", "History", "Branches", "Bookmarks", "Tags", "Remotes", "Shelved", and "Subrepositories".

A free Git client for Windows and Mac

Sourcetree simplifies how you interact with your Git repositories so you can focus on coding. Visualize and

- **Entwicklungsumgebung einrichten und nutzen**
 - Visual Studio Code (PlatformIO)
 - Sourcetree und GitLab

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**
Program-Ablauf-Pläne- für die wesentlichen Algorithmen

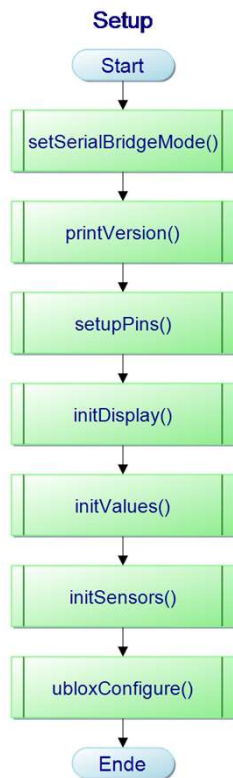
- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen



- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

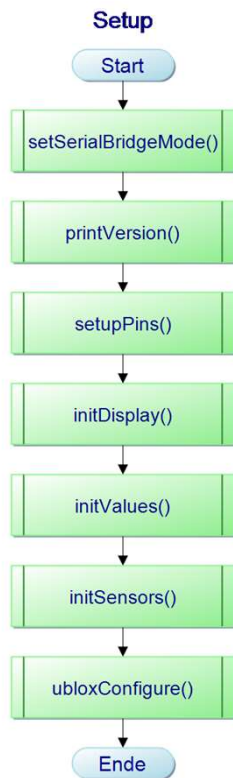


```
void setSerialBridgeMode()
{
    #if (USE_SERIAL_BRIDGE == ON)
        Serial.begin(BAUD_RATE);
        while (!Serial);
    #endif
}
```

```
void printVersion()
{
    Serial.println(F("\n\n\n====="));
    Serial.println(GPSDO_VARIANT);
    Serial.println(GPSDO_VERSION);
    Serial.println(F("Hardware-Design:DL7UKM and DL4ZA0"));
    Serial.println(F("Software-Design:R.Wieland -The one without a call-"));
    Serial.println(F("Putting it through its paces: DM4IM"));
    Serial.println(F("====="));
}
```

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen



```
void setupPins()
{
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(LOCK_DETECT_PIN, INPUT);

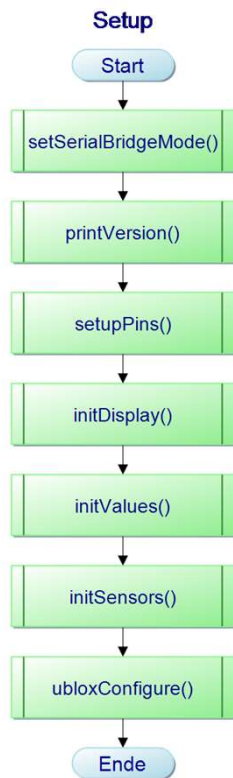
    digitalWrite(LED_LOCK_DETECT_PIN, LOCK_LOW);
    pinMode(LED_LOCK_DETECT_PIN, OUTPUT);
    digitalWrite(LED_LOCK_DETECT_PIN, LOCK_LOW);

    digitalWrite(LED_FORCED_HO_PIN, LOCK_LOW);
    pinMode(LED_FORCED_HO_PIN, OUTPUT);
    digitalWrite(LED_FORCED_HO_PIN, LOCK_LOW);

    digitalWrite(HO_RELAY, HO_OFF);
    pinMode(HO_RELAY, OUTPUT);
    digitalWrite(HO_RELAY, HO_OFF);
    if(!detectPinDA())
        analogPin=0xFF;
}
```

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen



```
void initDisplay()
{
    #if (GPSDO_DISPLAY != GPSDO_DISPLAY_NONE)
        display.init();
    #endif
}
```

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

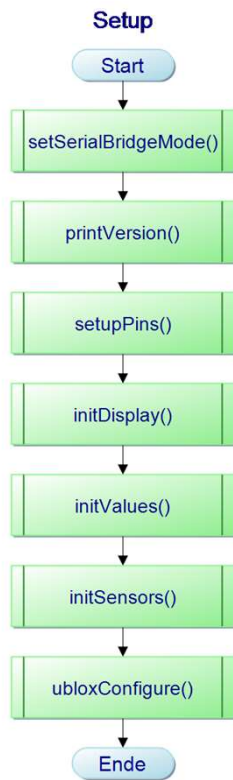
Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**
#Defines und die .ini-Datei



```
void initDisplay()
{
    #if (GPSDO_DISPLAY != GPSDO_DISPLAY_NONE)
        display.init();
    #endif
}
```

- **Firmware I**
#Defines und die .ini-Datei

```
void initDisplay()
{
    #if (GPSDO_DISPLAY != GPSDO_DISPLAY_NONE)
        display.init();
    #endif
}
```

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware I**
#Defines und die .ini-Datei

A screenshot of an IDE window showing the 'platformio.ini' file. The file contains a series of preprocessor definitions for the GPSDO project, including display type, LCD variants, and OLED specifications. The line numbers 64 through 78 are visible on the left side of the editor.

```
64 -D GPSDO_DISPLAY_NONE=0b000000000000
65 -D GPSDO_DISPLAY_LCD_2_VARIANT_1=0b000000000001
66 -D GPSDO_DISPLAY_LCD_2_VARIANT_2=0b000000000010
67 -D GPSDO_DISPLAY_LCD_2_VARIANT_3=0b000000000100
68 -D GPSDO_DISPLAY_LCD_2_VARIANT_4=0b000000001000
69 -D GPSDO_DISPLAY_LCD_4_VARIANT_1=0b000000010000
70 -D GPSDO_DISPLAY_LCD_4_VARIANT_2=0b000000100000
71 -D GPSDO_DISPLAY_LCD_4_VARIANT_3=0b000001000000
72 -D GPSDO_DISPLAY_LCD_4_VARIANT_4=0b000010000000
73 -D GPSDO_DISPLAY_OLED1106_128X64=0b001000000000
74 -D GPSDO_DISPLAY_OLED1306_128X64=0b010000000000
75 -D GPSDO_DISPLAY_OLED1306_VARIANT_1=0b100000000000
76 -D GPSDO_DISPLAY_AUTODETECT=0b111111111111
77 -D GPSDO_DISPLAY=GPSDO_DISPLAY_AUTODETECT
78
```

```
void initDisplay()
{
    #if (GPSDO_DISPLAY != GPSDO_DISPLAY_NONE)
        display.init();
    #endif
}
```

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen
#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen
Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Bedingte Übersetzungen mit Präprozessoranweisungen

```
void initDisplay()
{
    #if (GPSDO_DISPLAY != GPSDO_DISPLAY_NONE)
        display.init();
    #endif
}
```

- **Firmware I**

Bedingte Übersetzungen mit Präprozessoranweisungen

```
void GPSDO_Display::init()
{
    #if (DEBUG_GPSDO & DEBUG_GPSDO_DISPLAY == DEBUG_GPSDO_DISPLAY)
        Serial.println(F("DEBUG_GPSDO_DISPLAY:ON"));
    #endif

    #if (GPSDO_DISPLAY == GPSDO_DISPLAY_OLED1106_128X64)
        Serial.println(F("DISPLAY:ARDUINO_GPSDO_OLED 1106_128X64"));
    #endif

    #if (GPSDO_DISPLAY == GPSDO_DISPLAY_OLED1306_128X64 || \
        GPSDO_DISPLAY == GPSDO_DISPLAY_OLED1306_VARIANT_1 )
        Serial.println(F("DISPLAY:ARDUINO_GPSDO_OLED SSD1306"));
    #endif

    #if (GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_2_VARIANT_1 || \
        GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_2_VARIANT_2 || \
        GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_2_VARIANT_3 || \
        GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_2_VARIANT_4 )
        Serial.println(F("DISPLAY:GPSDO_DISPLAY_LCD_2"));
    #endif
}
```

```
#if (GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_4_VARIANT_1 || \
    GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_4_VARIANT_2 || \
    GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_4_VARIANT_3 || \
    GPSDO_DISPLAY == GPSDO_DISPLAY_LCD_4_VARIANT_4 )
    Serial.println(F("DISPLAY:ARDUINO_GPSDO_LCD_4"));
#endif

    #if (GPSDO_DISPLAY == GPSDO_DISPLAY_AUTODETECT)
        Serial.println(F("DISPLAY:GPSDO_DISPLAY_AUTODETECT"));
    #endif

    #endif
    setDisplayType();

    switch (displayType) ...

    switch (displayType) ...
}
```

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

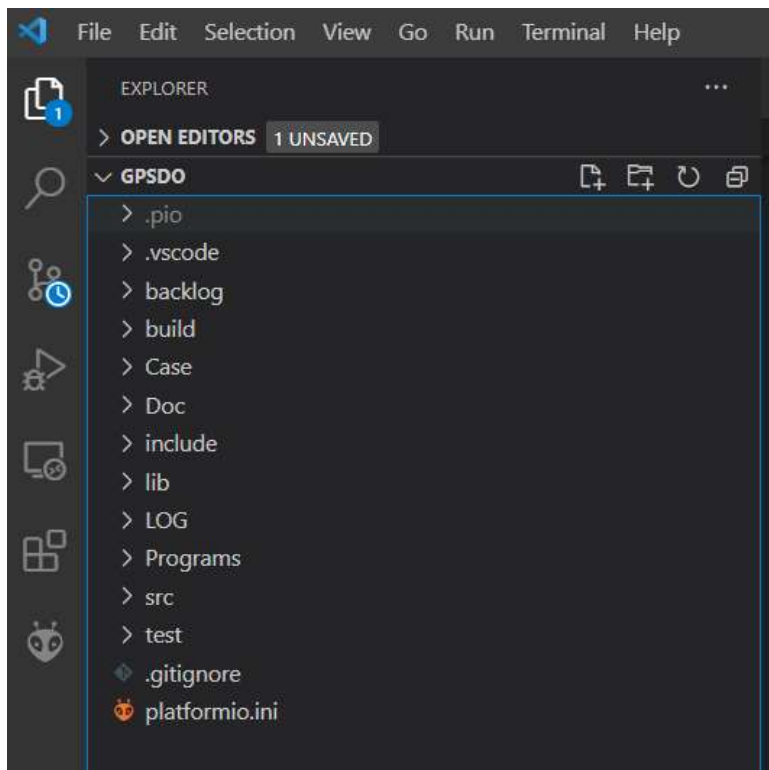
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware I**

Sourcedateien, Includes und Bibliotheken



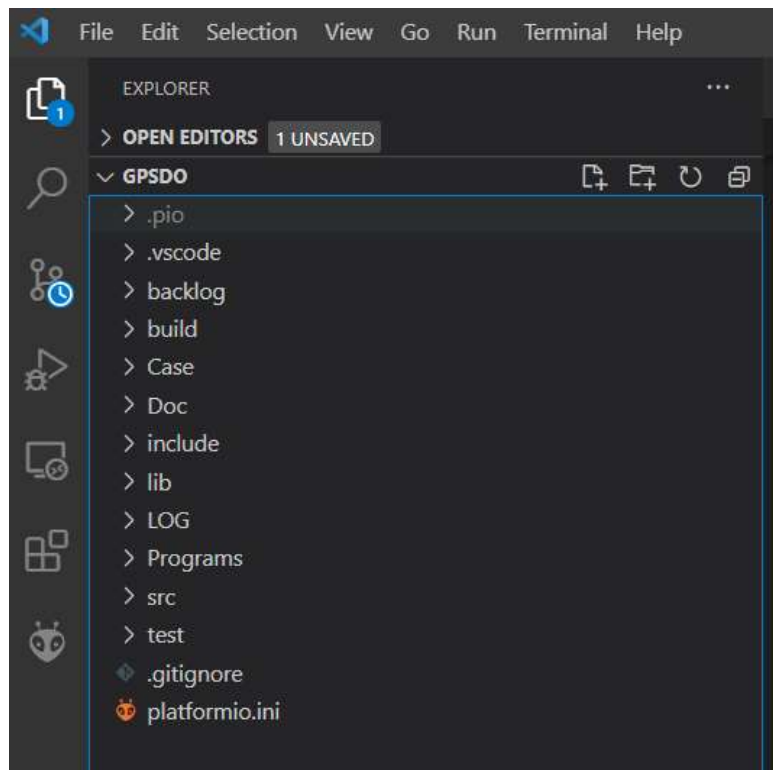
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware I**

Sourcedateien, Includes und Bibliotheken



Jetzt geht's ins Projekt...

- **Firmware I**

Program-Ablauf-Pläne- für die wesentlichen Algorithmen

#Defines und die .ini-Datei

Bedingte Übersetzungen mit Präprozessoranweisungen

Sourcedateien, Includes und Bibliotheken

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware II**

- Setuproutine

- Loop

- Displaydetect/GUI-Aufbau

- Temperatursensorerkennung

- Lockdetection -Einflussgrößen, Verarbeitung und Ausgabe

- Remanente Datenspeicherung im EEPROM

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw, 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware III**

Ublox-Typenauswertung (*Hard- und Softwareversion*)

Ublox-Konfiguration (*10KHz bzw. 100KHz-Puls, Stationary-Mode, Message-Disable*) und Auswertung

Proprietäres Protokoll im NMEA-Standard (*\$PGPSDO, Realtime-Ausgabe und Verarbeitung unter Processing*)

- **Firmware IV**
Holdover -digitales Schalten von analogen Werten

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

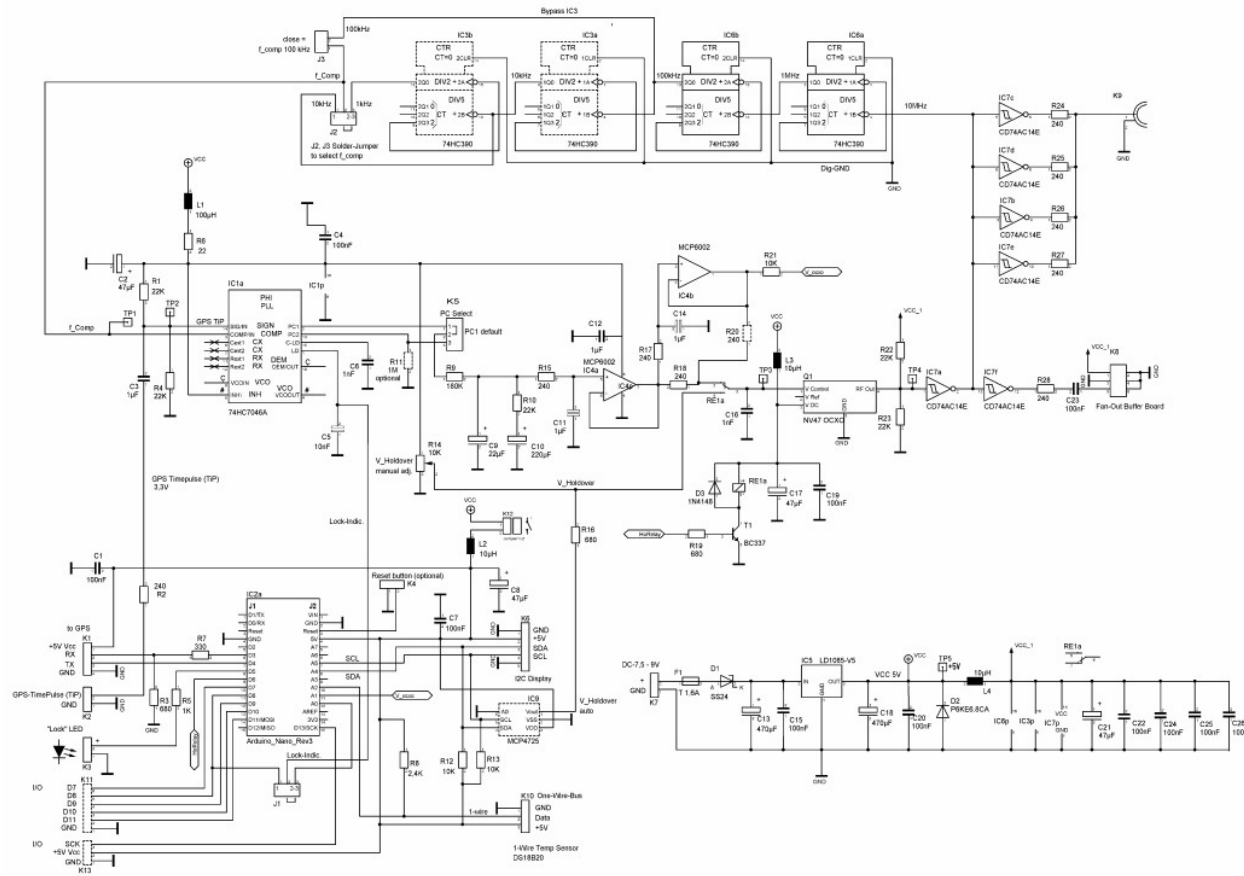
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- Firmware IV**

Holdover -digitales Schalten von analogen Werten



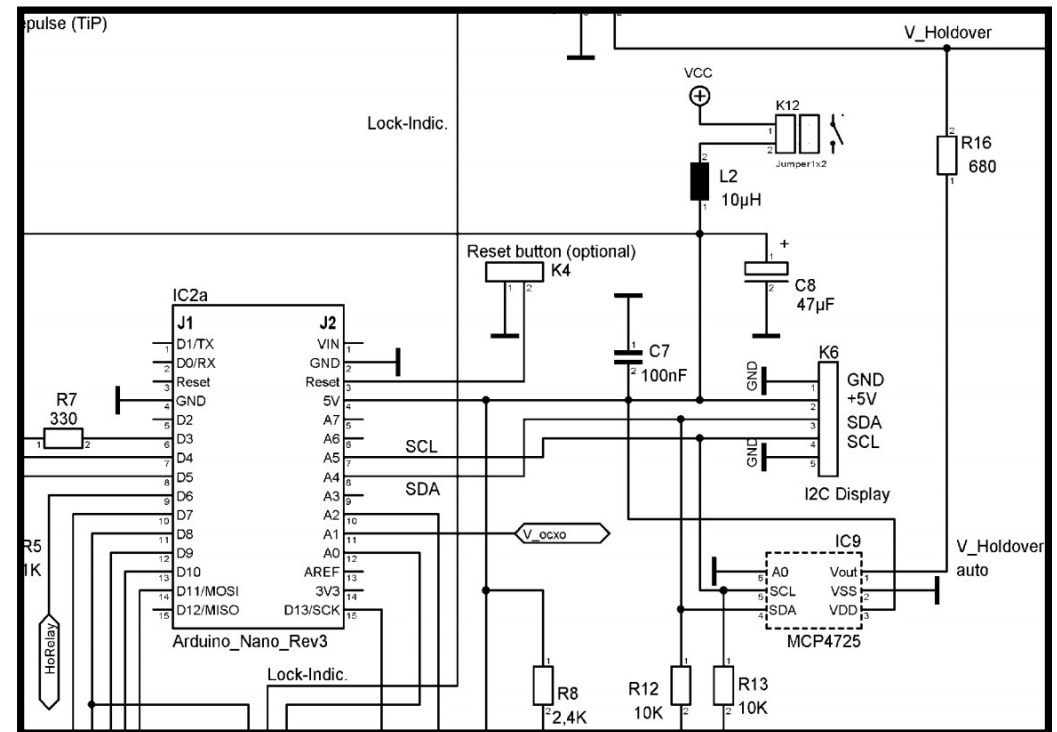
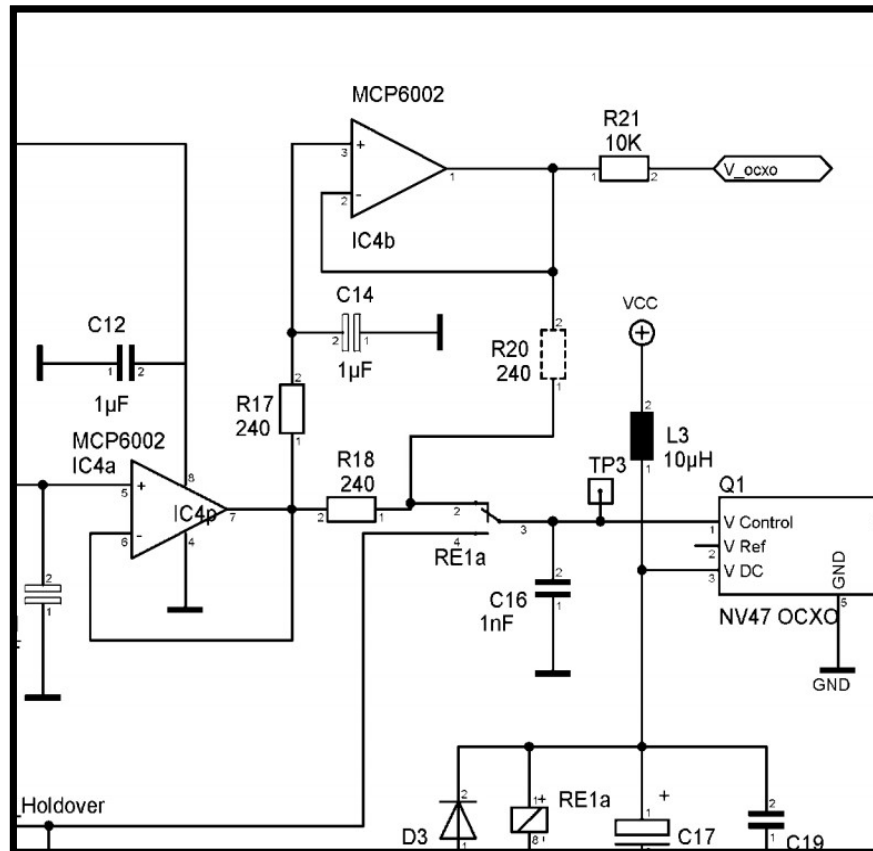
SOFTWARE FÜR EINEN GPSDO

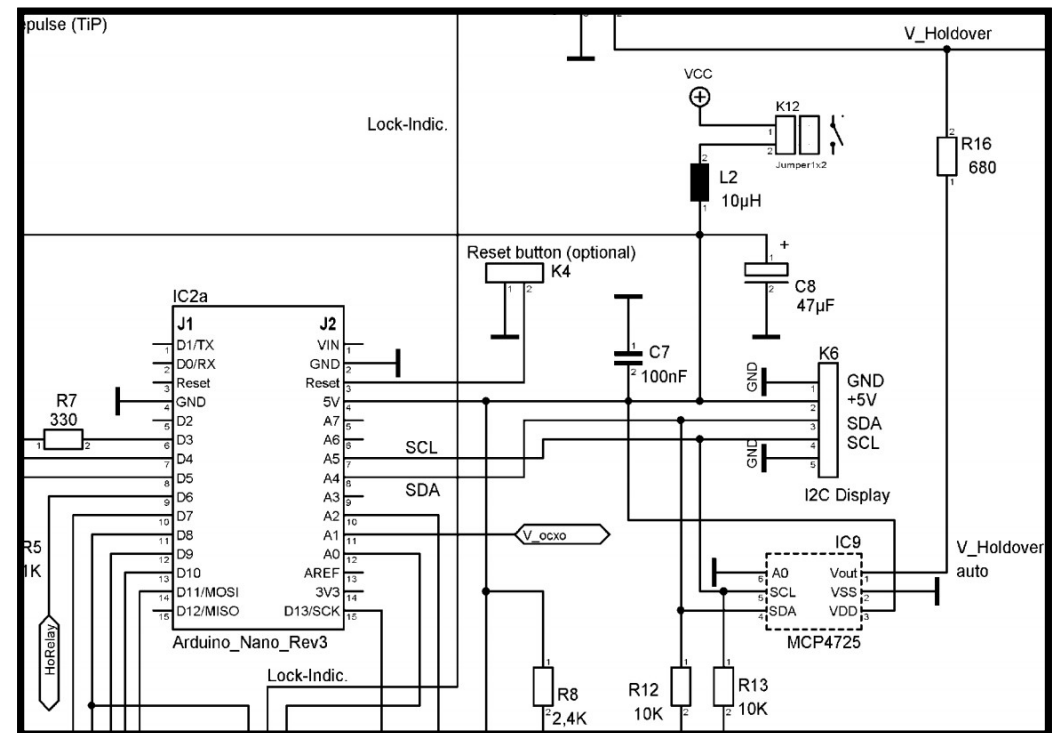
RAINER WIELAND



- Firmware IV**

Holdover -digitales Schalten von analogen Werten



[illegible]

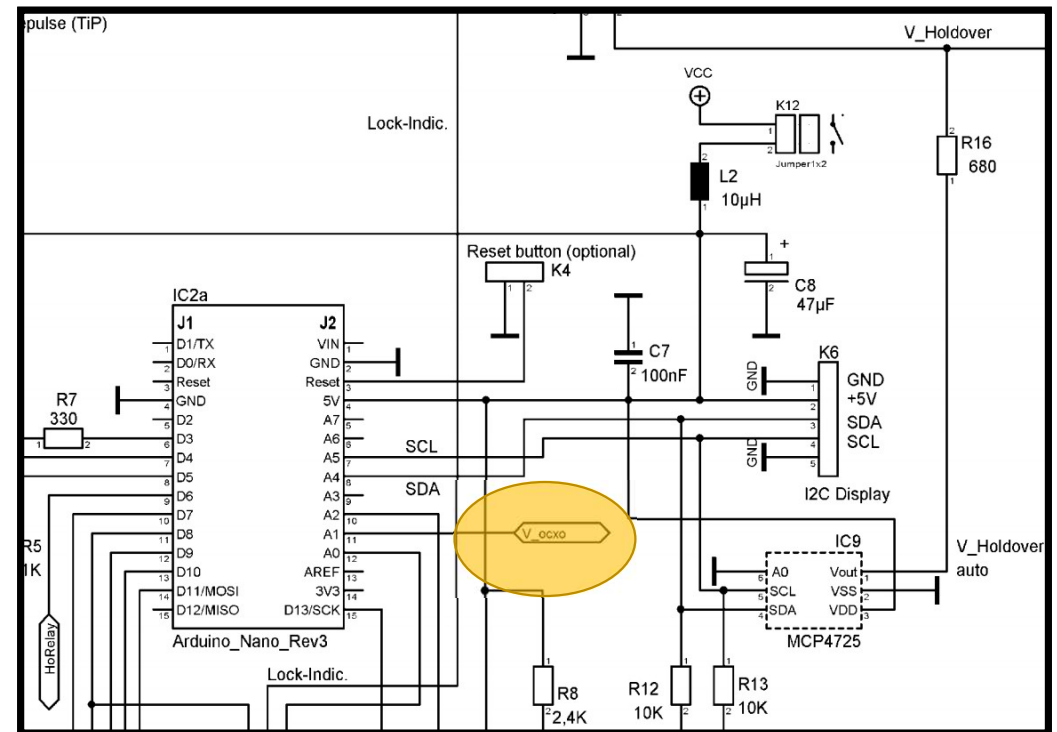
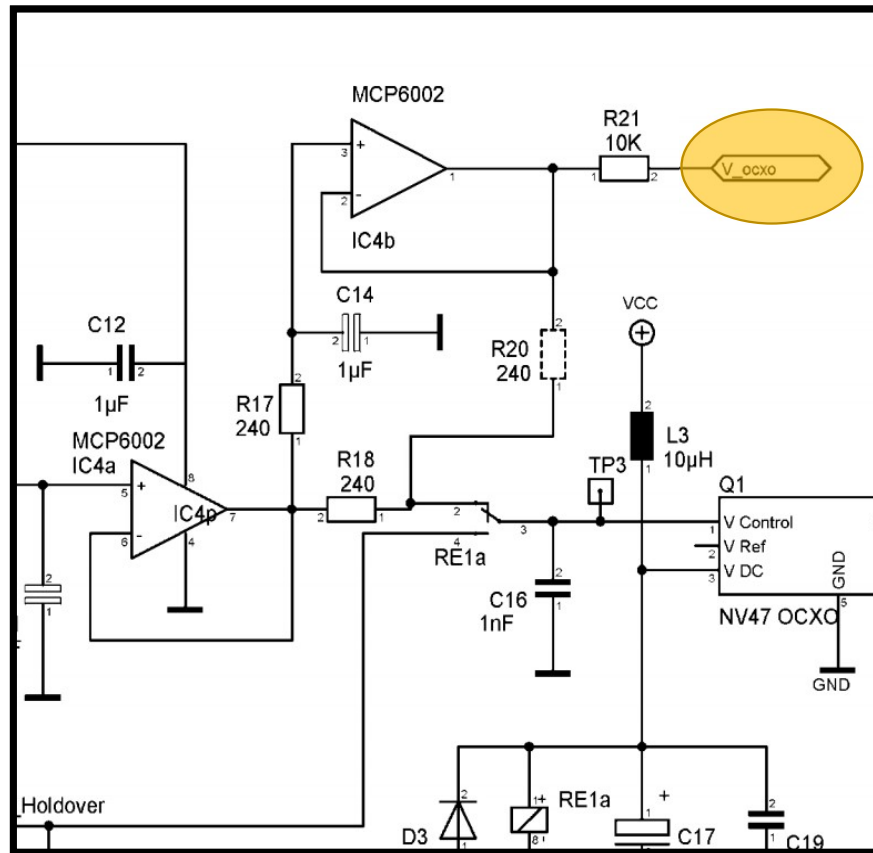
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- Firmware IV**

Holdover -digitales Schalten von analogen Werten



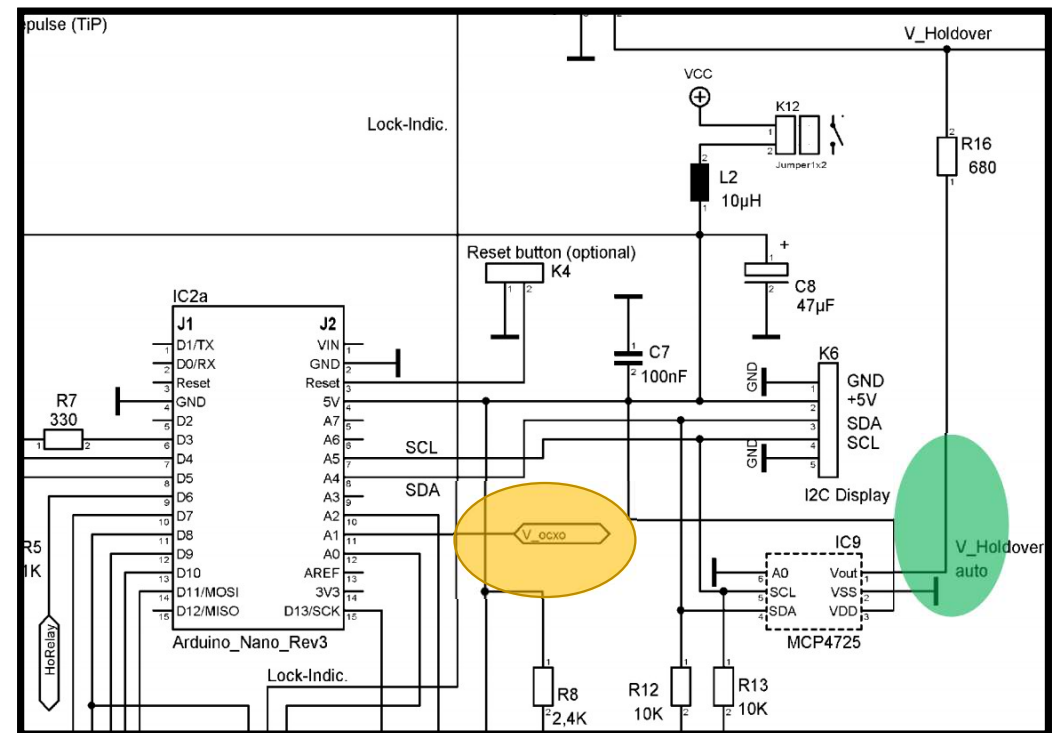
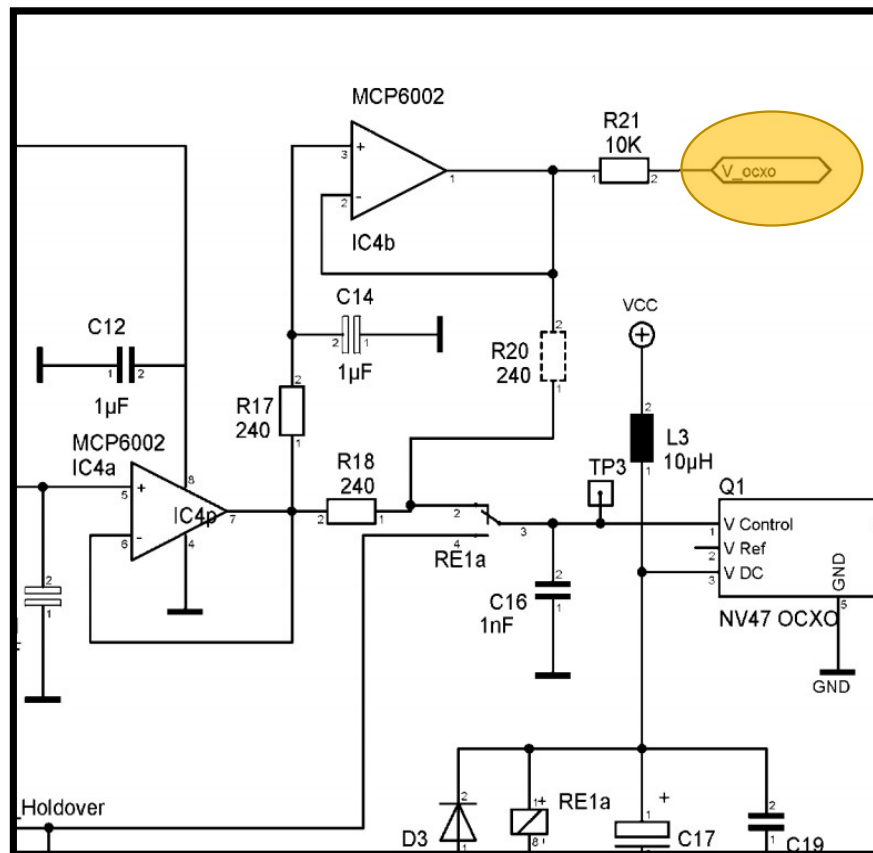
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- Firmware IV**

Holdover -digitales Schalten von analogen Werten



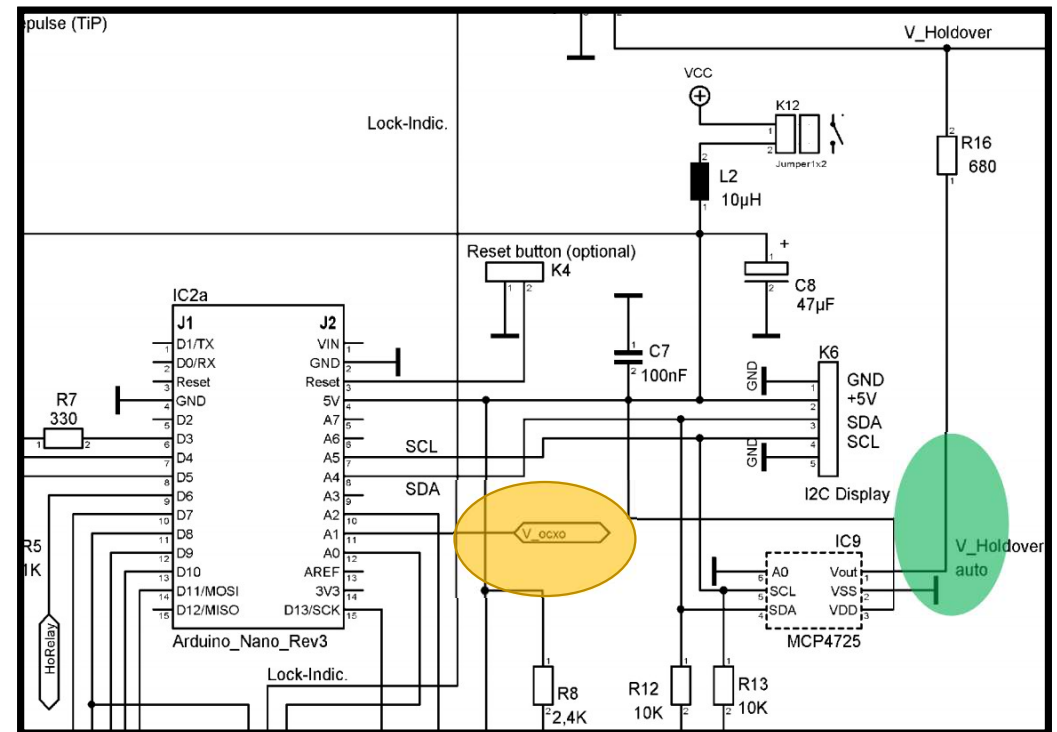
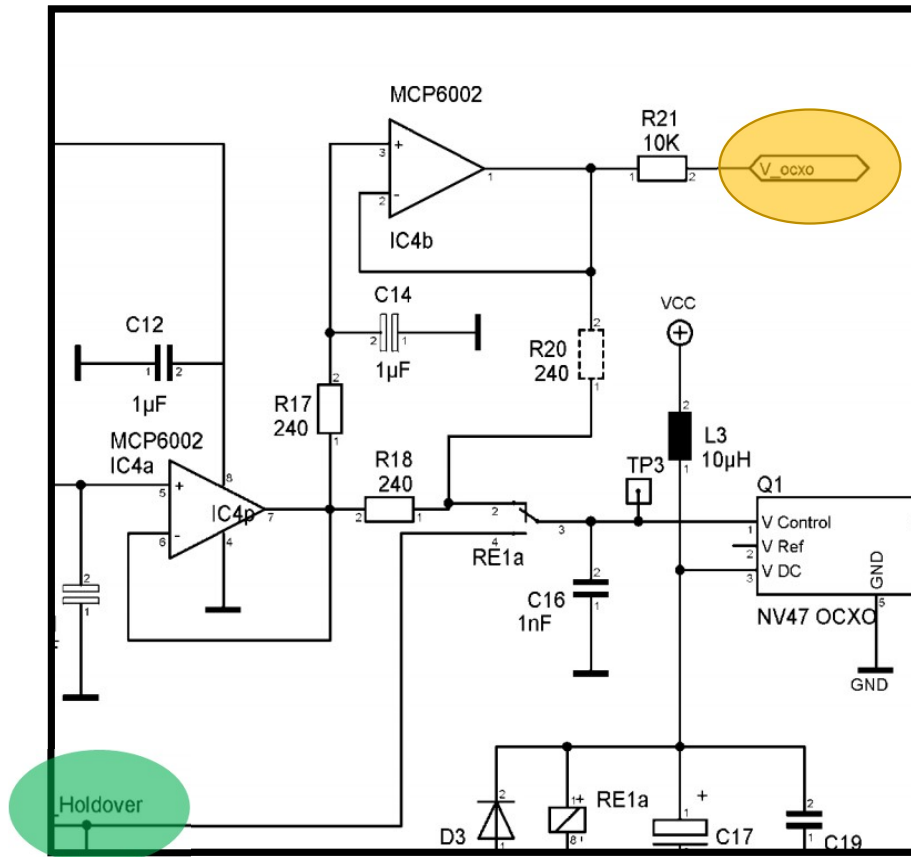
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND

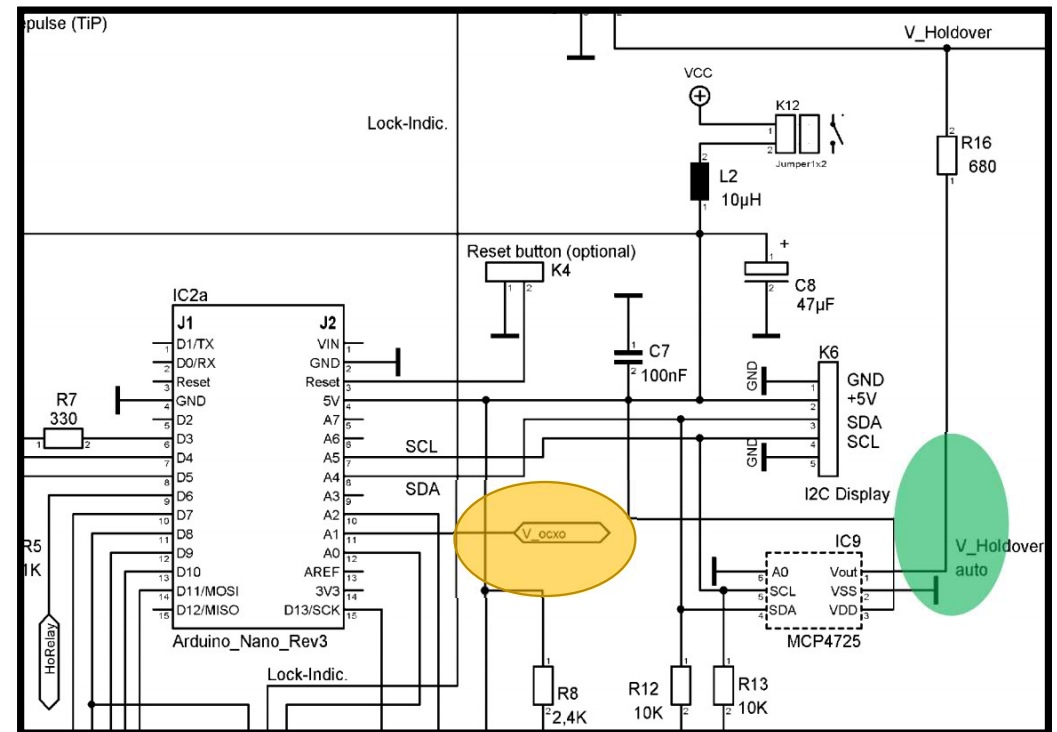


- Firmware IV**

Holdover -digitales Schalten von analogen Werten



Holdover -digitales Schalten von analogen Werten



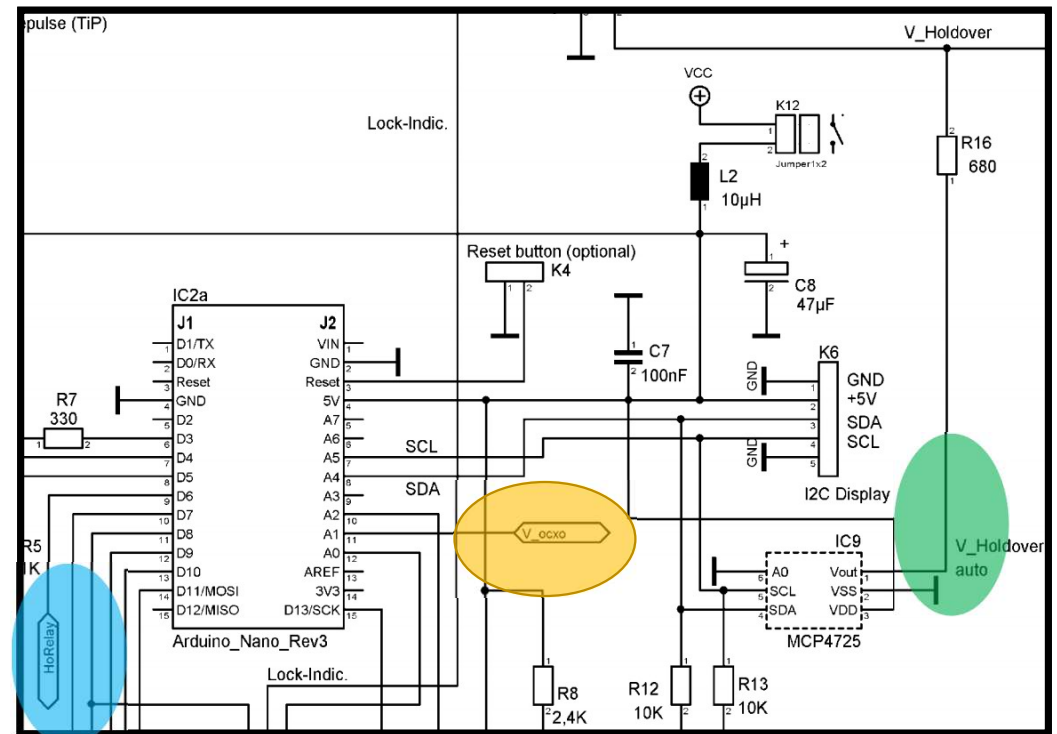
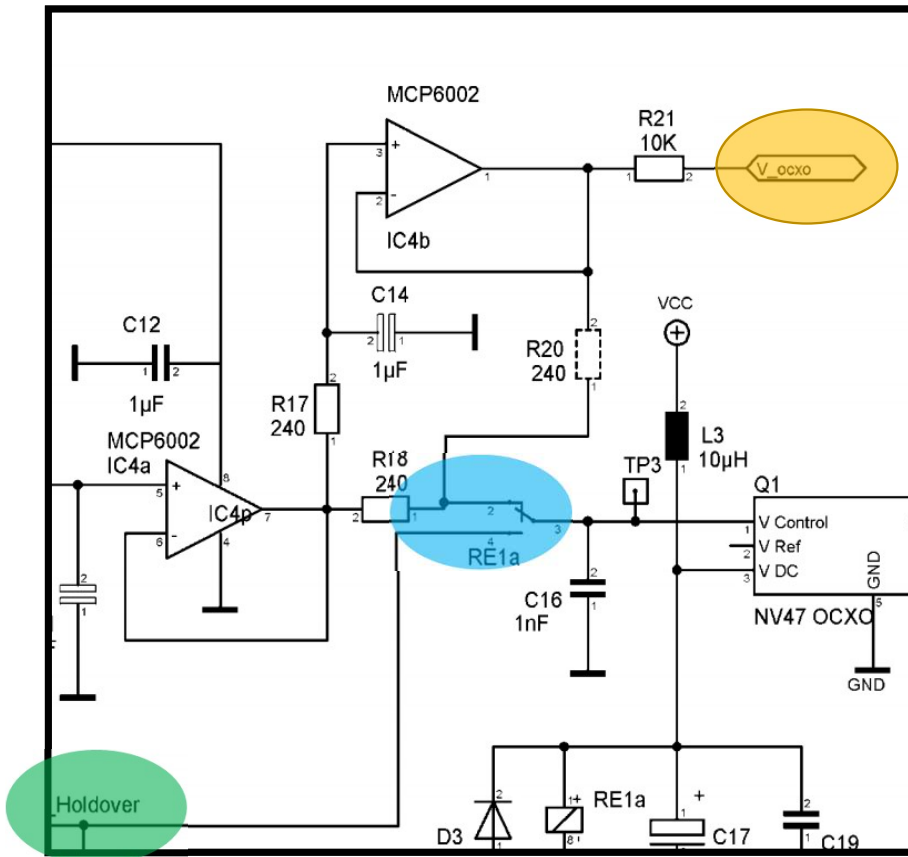
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



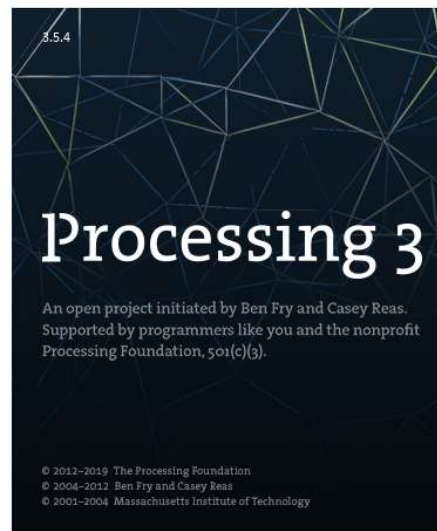
- Firmware IV**

Holdover -digitales Schalten von analogen Werten



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



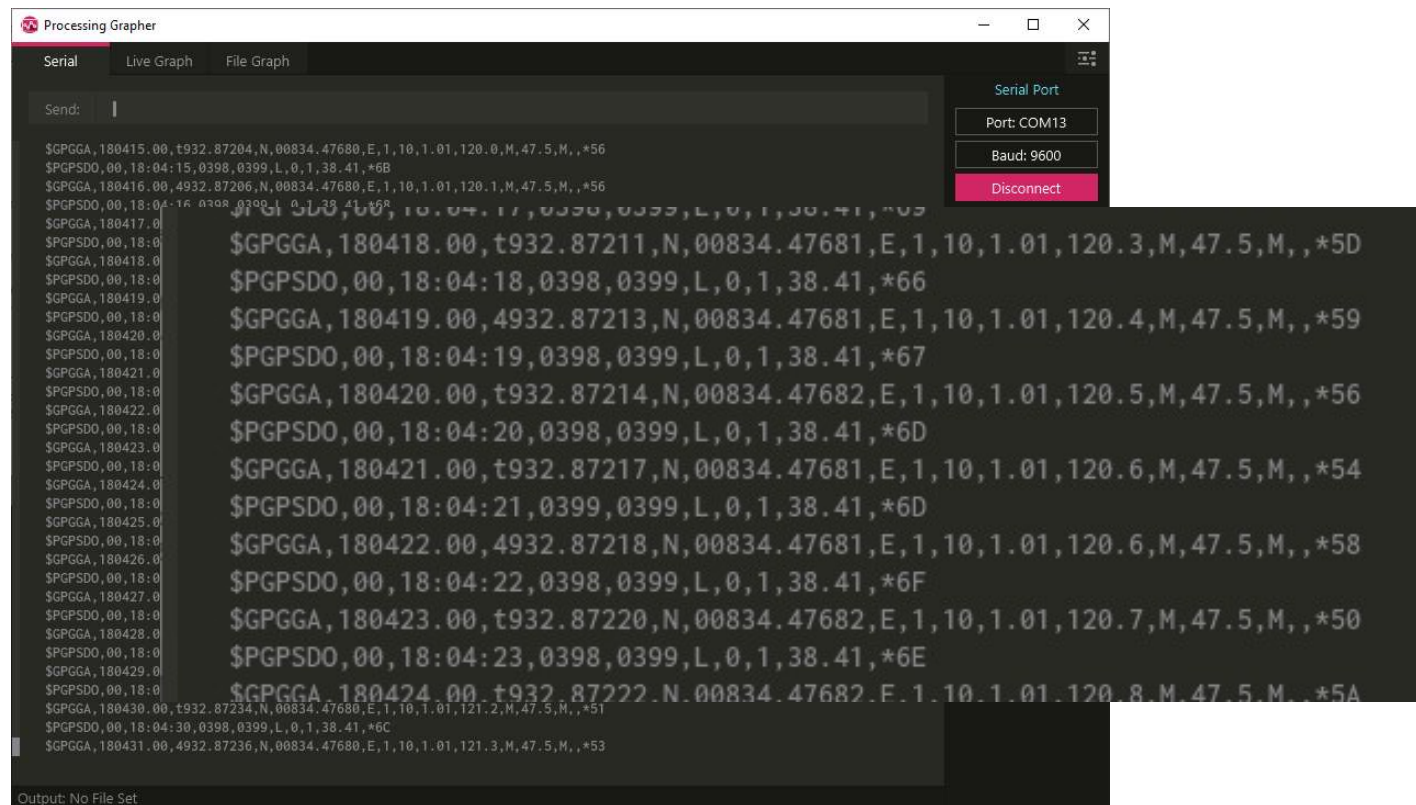
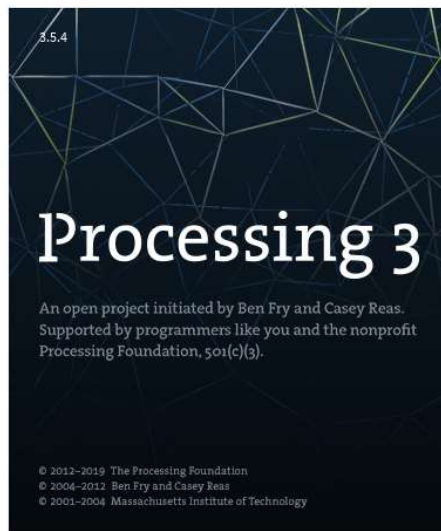
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



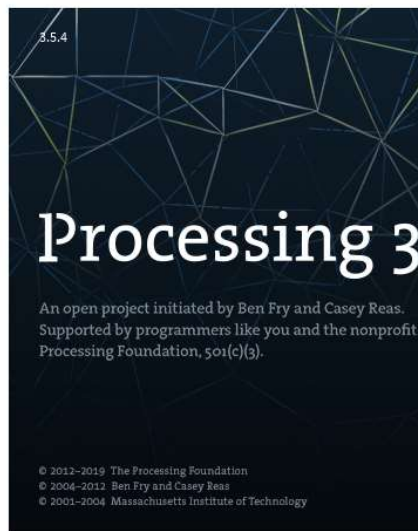
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

A screenshot of the Processing IDE window titled 'GPSDO_ProcessingGrapher_pde | Processing 3.5.4'. The window has a menu bar with 'Datei', 'Bearbeiten', 'Sketch', 'Debugger', 'Tools', and 'Hilfe'. Below the menu bar is a toolbar with a play button, a stop button, and a 'Java' dropdown menu. The main area shows the code for 'GPSDO_ProcessingGrapher_pde'. The code is a comment block containing metadata and a license. The console at the bottom shows the output: 'Plotting: 250, 0.0', 'Plotting: 250, 0.0', and 'Plotting: 250, 1.0'.

```
1  /* *****  
2  * PROCESSING GRAPHER  
3  *  
4  * @file      ProcessingGrapher.pde  
5  * @brief     Serial monitor and real-time graphing program  
6  * @author    Simon Bluett  
7  * @website   https://wired.chillibasket.com/processing-graphe/  
8  *  
9  * @copyright GNU General Public License v3  
10 * @date      6th March 2021  
11 * @version   1.2.2  
12 * ***** */  
13  
14 /*  
15 * Copyright (C) 2021 - Simon Bluett <hello@chillibasket.com>  
16 *  
17 * This file is part of ProcessingGrapher  
18 * <https://github.com/chillibasket/processing-graphe>  
19 *  
20 * ProcessingGrapher is free software: you can redistribute it and/or modify  
21 * it under the terms of the GNU General Public License as published by  
22 * the Free Software Foundation, either version 3 of the License, or  
23 * (at your option) any later version.  
24 */
```

Plotting: 250, 0.0
Plotting: 250, 0.0
Plotting: 250, 1.0

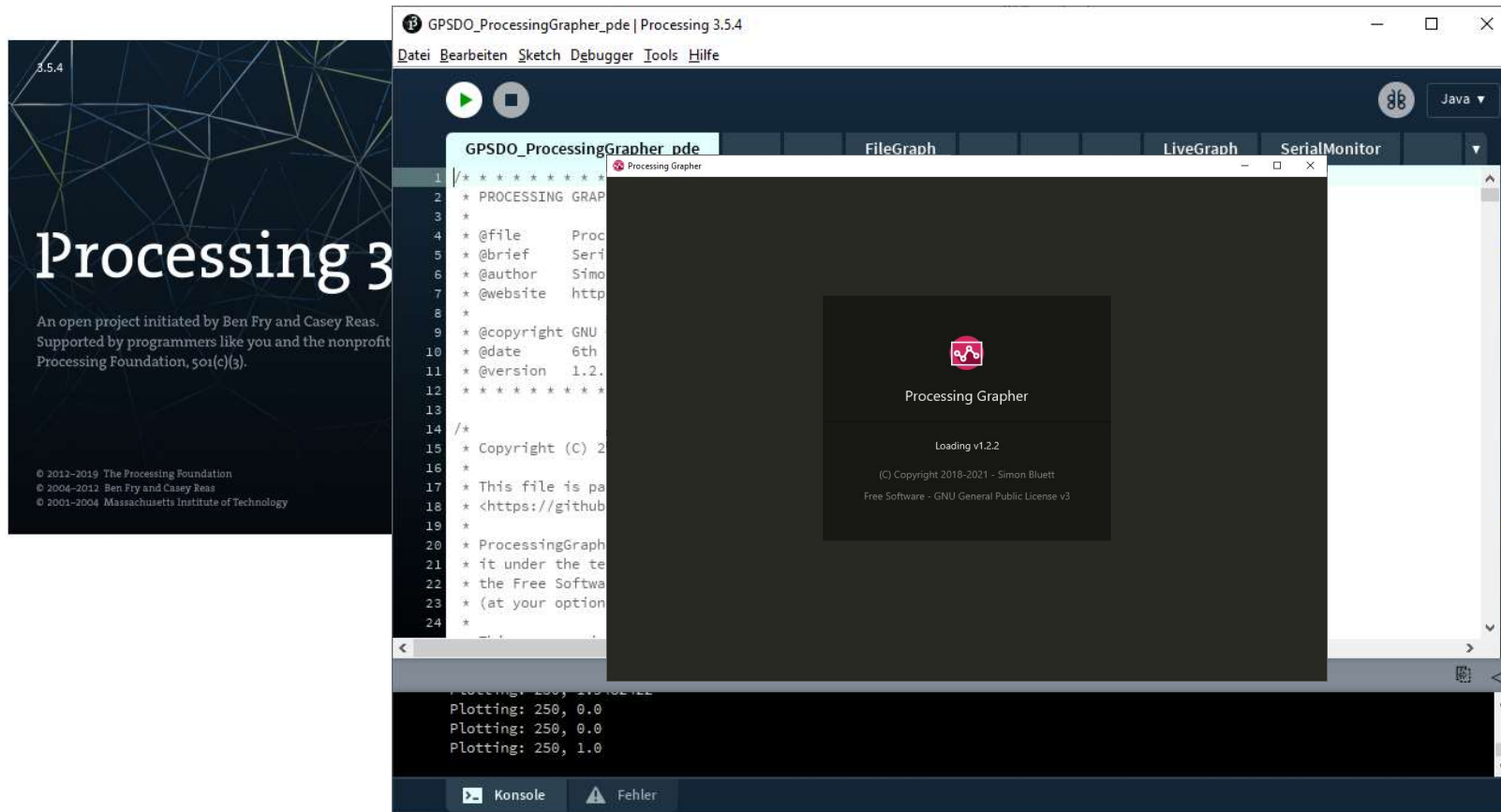
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



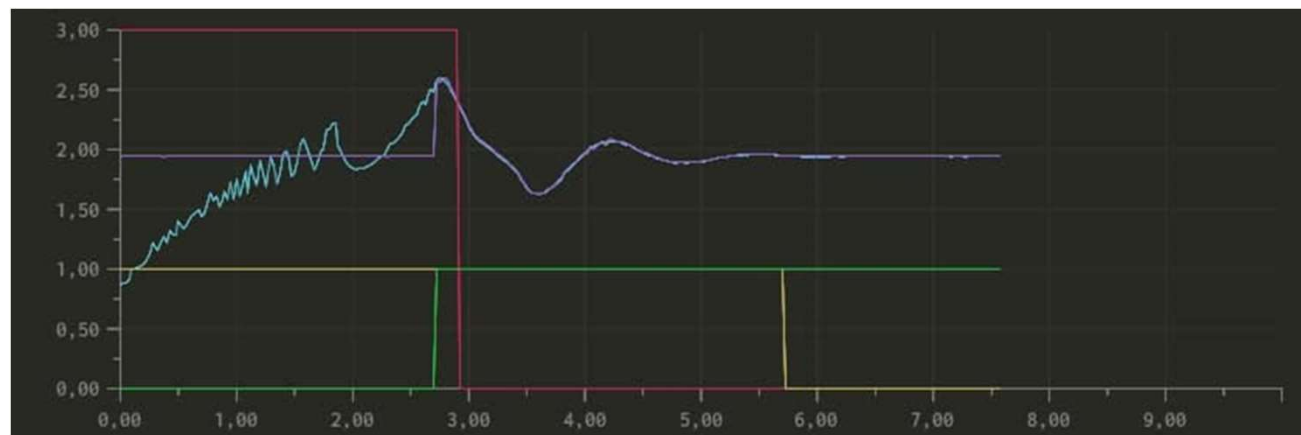
- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼



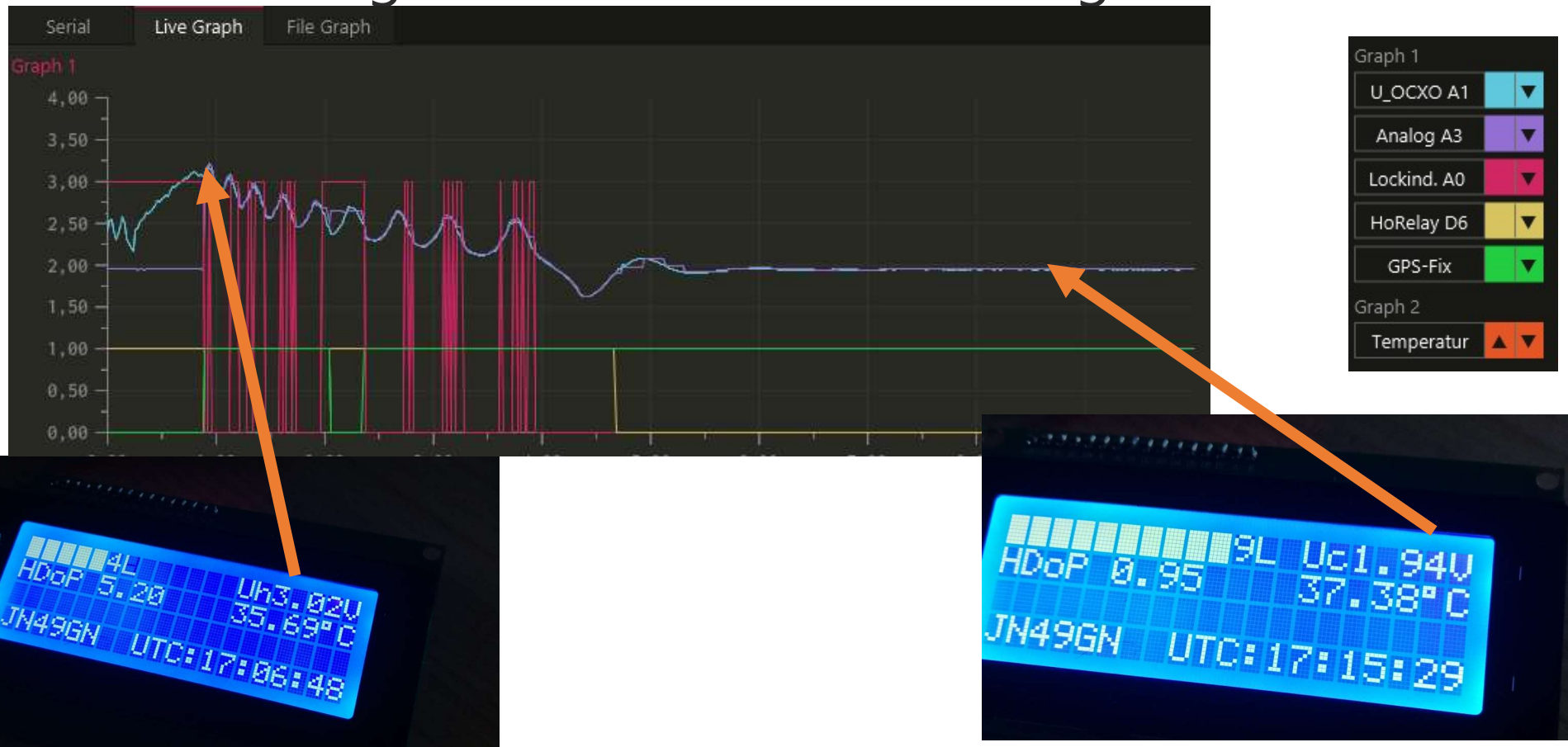
SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten



- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware IV**
Freier Lauf mit Forced Holdover

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



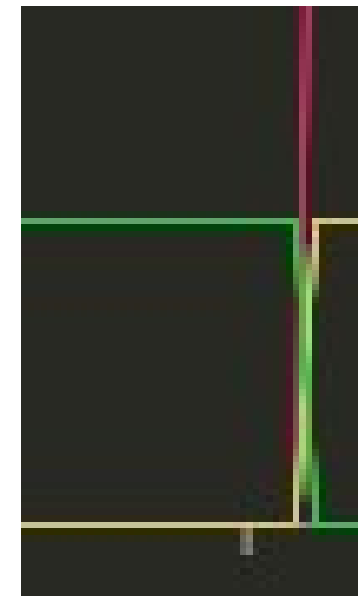
Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

- **Firmware IV**
Freier Lauf mit Forced Holdover

```
main.cpp • platformio.ini
src > main.cpp > readCapacitivePin(int)
573  * readCapacitivePin
574  * Source adapted from http://shelvin.de/simpler-kapazitiver-schalter-auf-einem-beliebigen-arduino-pin/
575  */
576  uint8_t readCapacitivePin(int pinToMeasure)
577  {
578
579      // Variables used to translate from Arduino to AVR pin naming
580      volatile uint8_t* port;
```

- **Firmware IV**
Freier Lauf mit Forced Holdover



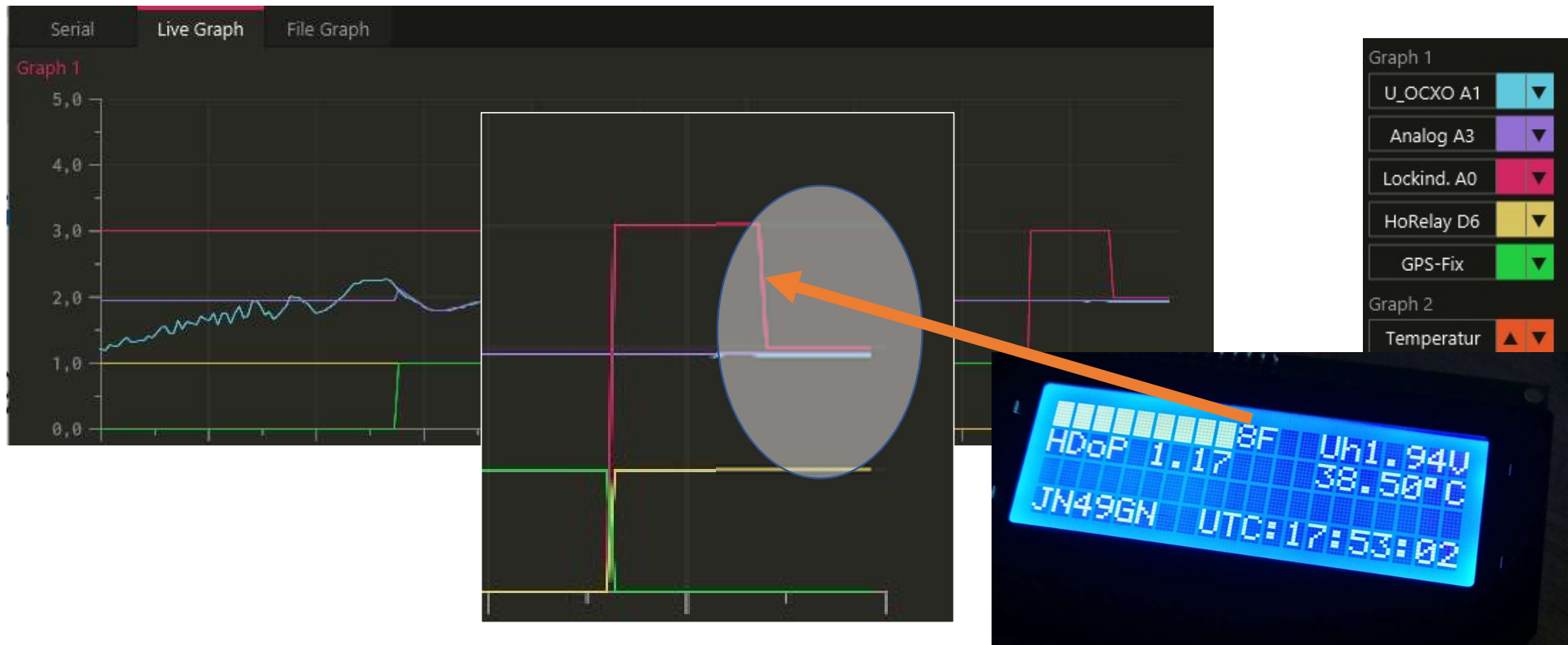
Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover

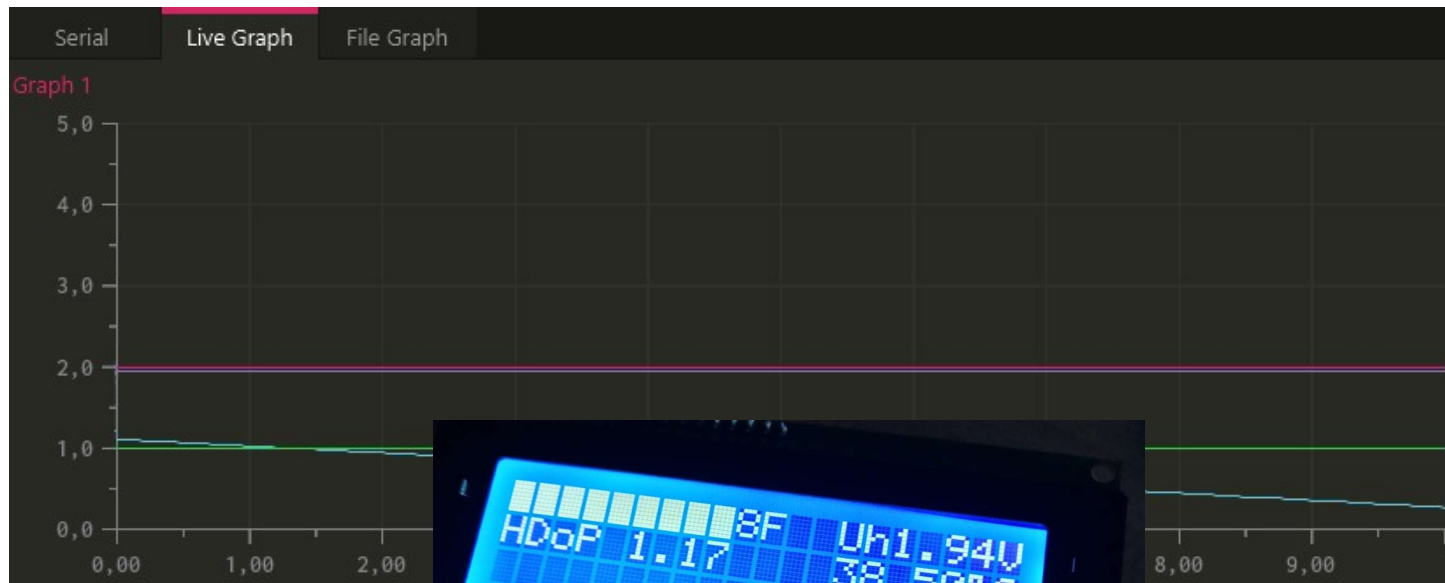


SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



Graph 1	
U_OCXO A1	▼
Analog A3	▼
Lockind. A0	▼
HoRelay D6	▼
GPS-Fix	▼
Graph 2	
Temperatur	▲ ▼

SOFTWARE FÜR EINEN GPSDO

RAINER WIELAND



- **Firmware IV**
Freier Lauf mit Forced Holdover



- **Firmware IV**
Freier Lauf mit Forced Holdover

- **Firmware IV**

Holdover -digitales Schalten von analogen Werten

Freier Lauf mit Forced Holdover

Fehlererkennung -Ausgabe mittels Morsecode

- **Firmware V**
Software optimieren mit dem Code-Inspektor

- **Firmware V**
Software optimieren mit dem Code-Inspektor

- Sicher Verpackt (eigenes Gehäuse für den 3D-Druck entwerfen)
- Externes Display -I2C -über kurz oder lang
- Ausblick
- Diskussion

- Sicher Verpackt (eigenes Gehäuse für den 3D-Druck entwerfen)
- Externes Display -I2C -über kurz oder lang
- Ausblick
- Diskussion

Fragen?