

Das Tytera MD380 ist für alle geeignet welche mal in den Bereich DMR rein gucken wollen und keinen OM haben welche ihnen mal ein Gerät leiht.

Das Gerät an und für sich ist gar nicht mal so schlecht, die Modulation mit dem eingebauten Mikrofon ist auch gut und fürs nur mal rein gucken, als Zeitgerät oder wenn man eh nur immer auf einem Repeater / Reflektor arbeitet vollkommen ausreichend.

Defizite tun sich eigentlich erst auf wenn man mit den Menüs arbeiten will, die sind zwar eigentlich auch selbsterklärend aber die Bedienung ist doch etwas träge.

Preislich so um die 140 EUR incl Programmierkabel, oder man nimmt gleich den Nachfolger MD390.

Was das Gerät aber sehr interessant macht ist die Experimentelle Firmware welche das Display beim Empfang durch eine bessere Ansicht ersetzt.



Dazu muss allerdings erst einmal die Userdatenbank eingespielt werden und das hat es leider in sich.

Es gibt zwar viele Anleitungen aber irgendwie fehlt immer wieder was, aber der Aufwand lohnt sich.

Eine relativ gute Anleitung zu Windows findet sich hier:

<https://github.com/travisgoodspeed/md380tools/blob/master/README.de.md>

aber auch hier fehlen Punkte wo man dann teilweise etwas nachdenken muss, wen man sich allerdings mit dem PC nicht so auskennt kann es da auch durchaus zu Problemen führen.

So hatte auch ich Probleme nach der Installation da ich Windows 10 - 64 Bit nutze und der Autor des Artikels vermutlich Win XP - 32 Bit

Da mich eben auf dem Reflektor auch jemand drum gebeten hat eine Idioten sichere Anleitung zu machen werde ich dem einfach mal nachkommen. Bei vielen Sachen wird sich der versierte Windowsuser denken das kann doch jeder, sind doch alles Basics aber dem ist nicht so.

Ich möchte hier einfach nochmal auf die Installation unter Windows 10 zurück kommen ohne mich mit fremden Federn zu schmücken, die Urversion stammt von KK4VCZ und die Übersetzung von DG9VH.

Also mein System mit dem die Screenshots entstanden sind

- Betriebssystem: Windows 10 - 64 Bit
- Browser: Firefox 50
- Dateifexplorer: Windows Explorer

Habt ihr eine andere Konfiguration können die Screenshots bei euch natürlich etwas anders ausschauen, besonders bei anderen Betriebssystemen kann es auch vorkommen das etwas nicht mehr passt.

Vorbereitung

Erstellt einen Ordner wo ihr alle Download speichert, da ich faul bin habe ich einen Ordner mit dem Namen „MD380-Tools“ auf dem Desktop erstellt.

Installation von Git

Download:

<https://git-scm.com/download/win>

Hierbei sollte der Download automatisch starten, falls er nicht startet müsst ihr die jeweilige Setup-Version auswählen, also 32 Bit oder 64 Bit. Bei mir startete der Download automatisch und es wurde auch automatisch die momentan aktuelle 64 Bit Version (Git-2.10.2-64-bit.exe)

ausgewählt.

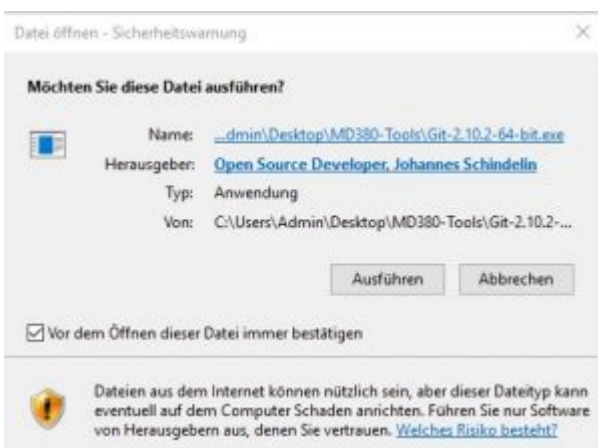
So sollte es ausschauen



Die Datei dann im vorher erstellten Ordner speichern.

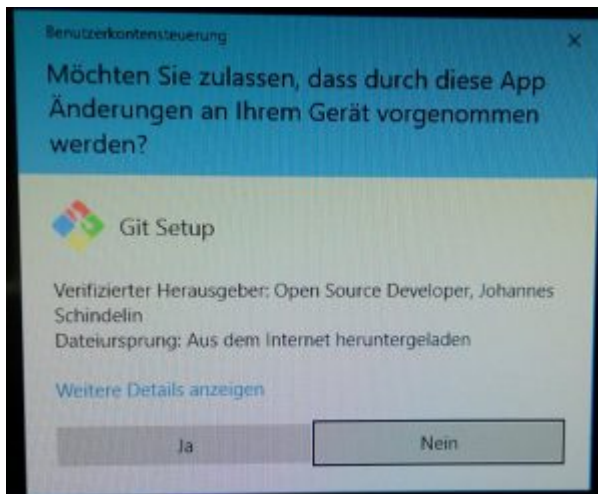
Doppelklick auf Git-2.10.2-64-bit.exe (bzw eure Version) um die Installation zu starten.

Vermutlich (wenn nicht durch euch deaktiviert) bekommt ihr die Sicherheitswarnung



welche ihr dann durch einen Klick auf „Ausführen“ bestätigen müsst.

Danach kommt eine weitere Sicherheitsmeldung welche ihr durch einen Klick auf „Ja“ bestätigen müsst.

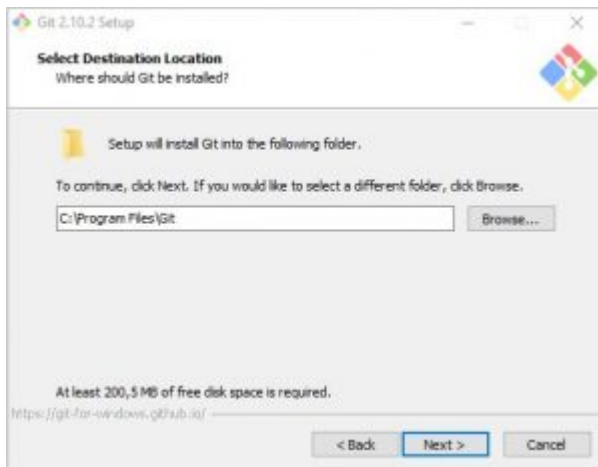


Diese Meldungen kommen teilweise auch bei den anderen Programmen und müssen jeweils mit „Ausführen“ bzw „Ja“ bestätigt werden, ich werde dazu im Verlauf KEINE weiteren Screenshots posten, denke mal diese Prozedur kennt eh jeder.

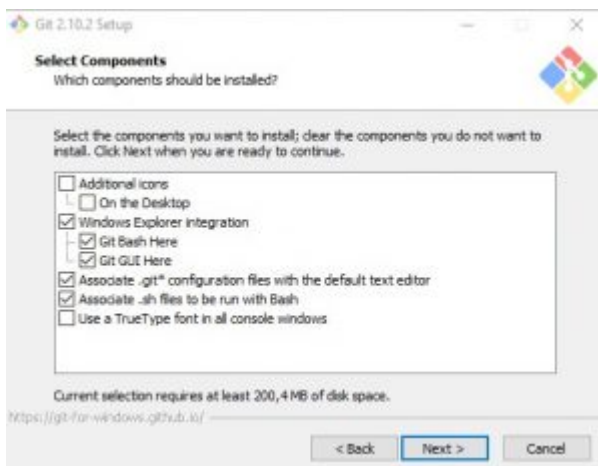
Weiter gehts mit, nach der Bestätigung der 2. Sicherheitsmeldung müssen wir die Lizenz durch einen Klick auf „Next >“ bestätigen.



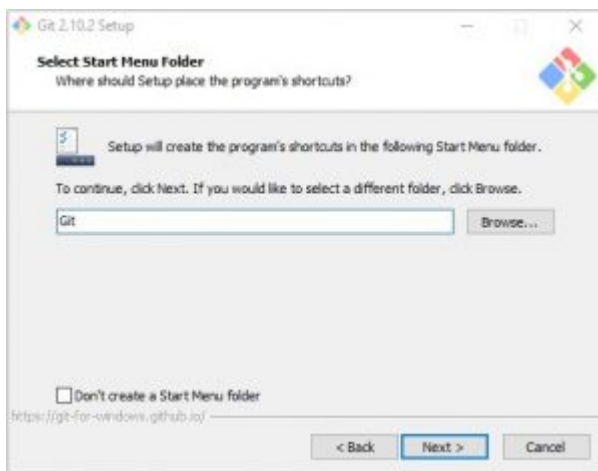
Der nächste Punkt ist die Auswahl des Installationsverzeichnis, hier bedarf es keiner Änderung, einfach mit einem Klick auf „Next >“ die Seite bestätigen.



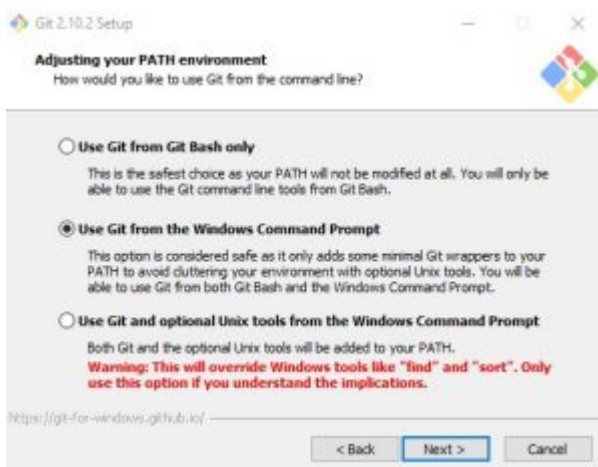
Auch bei der Auswahl der Komponenten müssen wir nichts ändern und können einfach mit einem Klick auf „Next >“ die Seite bestätigen.



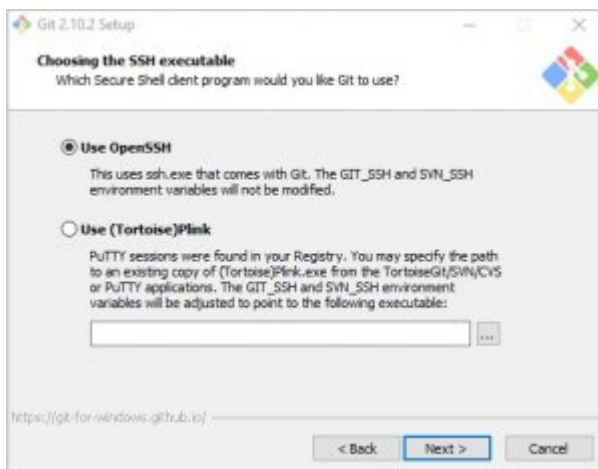
Auch beim Startmenüeintrag einfach mit einem Klick auf „Next >“ die Seite bestätigen.



Nun sind wir beim Punkt „Adjusting your PATH environment“ angelangt, auch hier bedarf es keiner Änderung einfach mit einem Klick auf „Next >“ die Seite bestätigen. Falls das Fenster später einmal anders ausschauen sollte, es muss der Punkt „Use Git from the Windows Command Prompt“ ausgewählt werden.



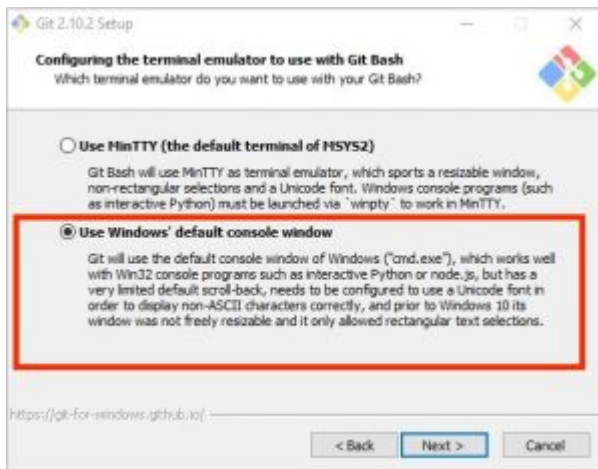
Auch bei der Auswahl des Secure Shell Client Programms bedarf es keiner Änderung einfach mit einem Klick auf „Next >“ die Seite bestätigen.



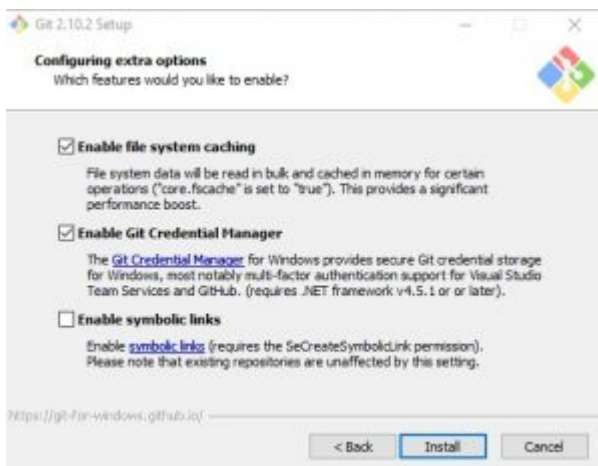
Auch bei „Configuring the line ending conversions“ einfach mit einem Klick auf „Next >“ die Seite bestätigen.



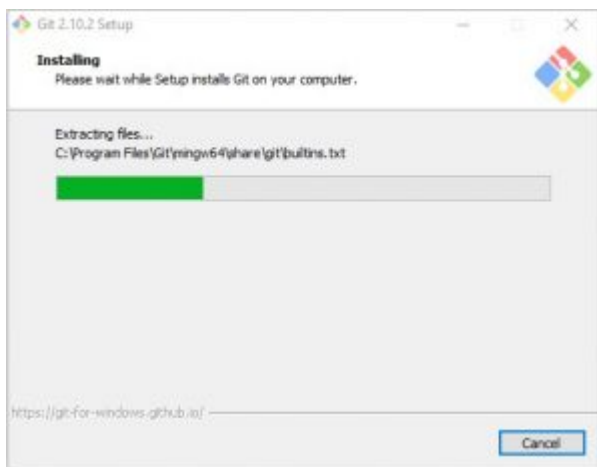
So nun müssen wir „endlich“ mal was ändern, also auf der Seite „Configuring the terminal emulator to use with Git Bash“ den unteren Punkt „Use Windows' default console window“ auswählen und mit einem Klick auf „Next >“ die Seite bestätigen.



Auf der nächsten Seite „Configuring extra options“ mit einem Klick auf „Install“ die Installation ausführen.



Nun läuft die Installation durch



und muss dann nur noch mit einem Klick auf „Finish“ beendet werden, wenn man die Release Notes nicht lesen möchte dann die Box „View Release Notes“ abwählen.



Installation von Make

Download: <http://gnuwin32.sourceforge.net/packages/make.htm>

auf der Seite dann bei „Complete package, except sources“ auf „Setup“ klicken oder im Text vorher auf „Setup program,,“, die jeweils richtigen Links sind im Screenshot markiert.

Download

If you download the **Setup program** of the package, any requirements for running applications, such as dynamic link libraries (DLL), the **dependencies** are **not** included. Developer files (header files and libraries) from other packages are however not included, so if y

Description	Download	Size	Last change	MD5sum
• Complete package, except sources	Setup	3384632	25 November 2006	848127942E3e4E283024687421708A
• Sources	Setup	1152945	25 November 2006	b096c0e82501040a1ad85040m2f208
• Binaries	Zip	485645	25 November 2006	30229802c27a212a260202f2be16043
• Dependencies	Zip	748204	25 November 2006	6372416ee524f68279413e4309e49043
• Documentation	Zip	2470170	25 November 2006	43a876449540a83a032312116490a2d7
• Sources	Zip	2084753	25 November 2006	0be8f0f17e5206893948e94778ee643

You can also download the files from the [GnuWin32 files page](#).

You can [monitor](#) new releases of the port of this package.

nun werdet ihr auf eine werbe geflutete Seite weitergeleitet auf der dann der Download in ca 5 Sekunden automatisch startet. Sollte das nicht funktionieren unter dem grünen Balken auf „Direct Link“ klicken.



Speicher die Datei make-3.81.exe (oder evtl schon eine neuere Version) in den am Anfang erstellen Download Ordner.

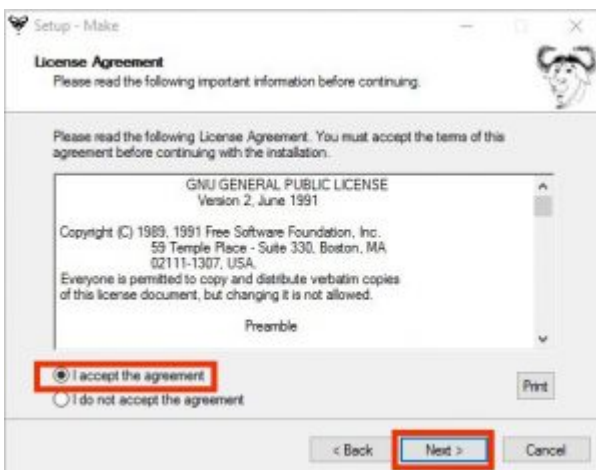
Rechtsklick auf die Datei make-3.81.exe und „Als Administrator ausführen“ auswählen. (Evtl geht es auch ohne Administratorenrechte, aber ich hatte mit Make Probleme gehabt)



Die ersten Seite können wir gleich mit einem Klick auf „Next >“ bestätigen



Bei den Lizenzbestimmungen „I accept the agreement“ auswählen und mit einem Klick auf „Next >“ bestätigen.



Auf der nächsten Seite ist die Auswahl des Installationsverzeichnis zu bestätigen, hier braucht normalerweise nichts geändert zu werden.

Allerdings ist der hier angezeigte und auf dem Screenshot **rot markierte** Path

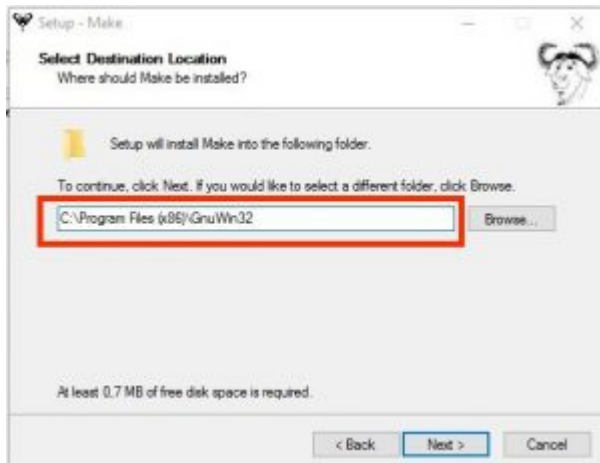
Bei 64 Bit: *C:\Program Files (x86)\GnuWin32*

Bei 32 Bit: *C:\Program Files\GnuWin32*

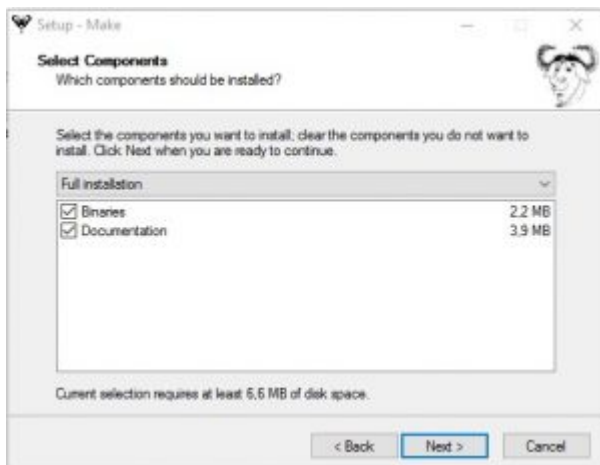
später wichtig.

Das war auch bei mir ein Problempunkt da in der originalen Anleitung von einem 32 Bit

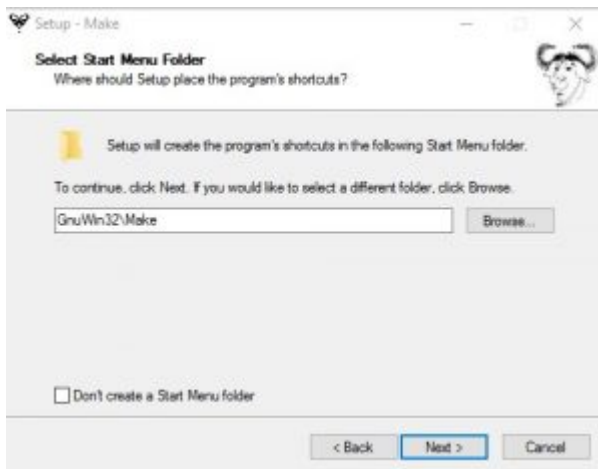
System, ausgegangen wird.



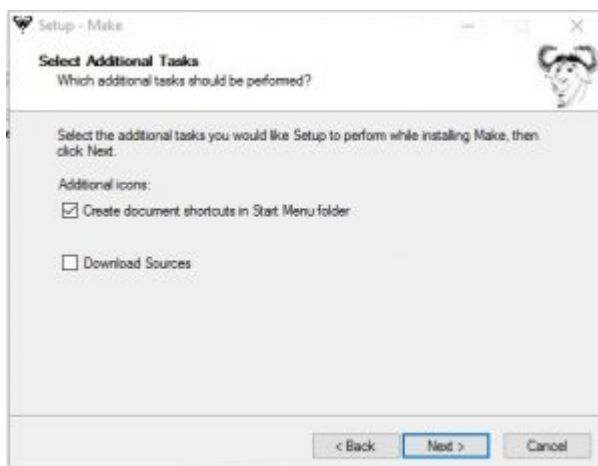
Die Komponentenliste wieder ohne erforderliche Änderungen mit einem Klick auf „Next >“ bestätigen.



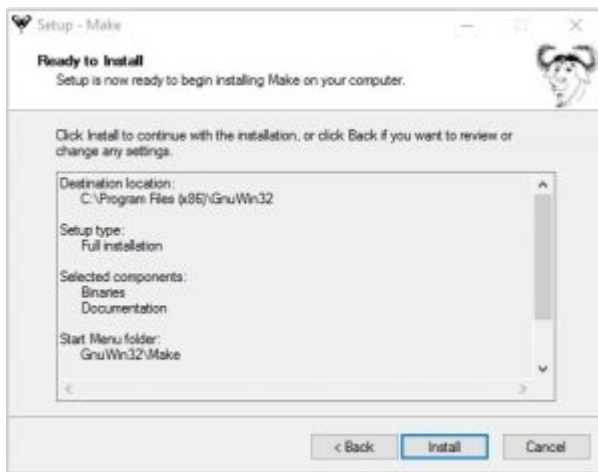
Auch den Startmenüeintrag ohne erforderliche Änderungen mit einem Klick auf „Next >“ bestätigen.



Auch die Seite „Select Additional Tasks“ mit einem Klick auf „Next >“ bestätigen.



Im nächsten Schritt dann die Installation mit einem Klick auf „Install“ starten.



Und die Installation mit einem Klick auf „Finish“ beenden.



Installation von Python 2.7

Download: <https://www.python.org/downloads/>

Auf der sich öffnenden Seite die Version 2.7.x (in meinem Fall war das die Datei: python-2.7.12.msi) durch einen Klick runter laden. Also die 2.7 und nicht die 3er Version.

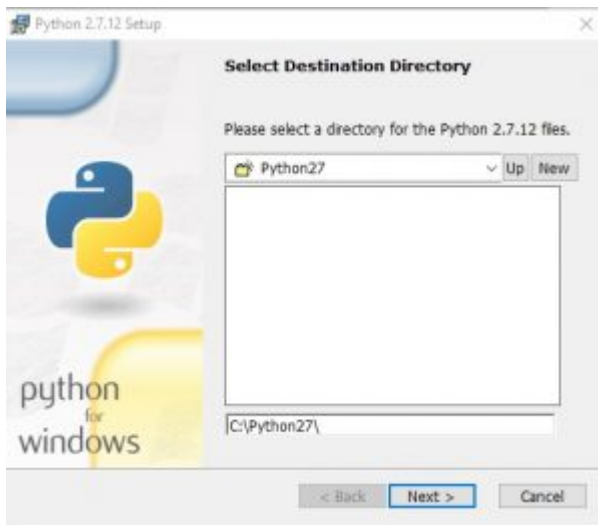
Die Datei python-2.7.12.msi (bzw evtl eine neuere Version der 2er Reihe) dann wieder im extra erstellten Download Ordner speichern.



Jetzt Doppelklick auf die Datei python-2.7.12.msi und mit einem Klick auf „Next >“ die Seite bestätigen.



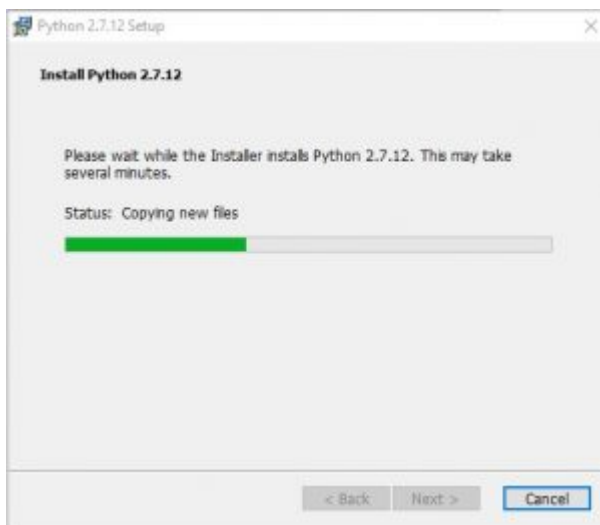
Auch die Seite „Select Destination Directory“ wieder mit einem Klick auf „Next >“ bestätigen.



Die Seite „Customize Python 2.7.12“ auch wieder mit einem Klick auf „Next >“ bestätigen.



Nun startet die Installation, hierbei muss man wieder eine Sicherheitsmeldung bestätigen. Dieser Schritt kann etwas dauern, also nicht wundern wenn 1 Minute lang nichts passiert.



Mit einem Klick auf „Finish“ beenden wir die Installation



Jetzt wechseln wir mit dem Windows Explorer in das Verzeichnis

`C:\Python27`

wechseln und dort eine Kopie der Datei `python.exe` erstellen welche dann den Namen `python2.exe` bekommt.

Wie das funktioniert sollte vermutlich jeder wissen, ansonsten

- Datei markieren (einfacher Klick mit der linken Maustaste)
- Die Tasten STRG und C drücken (um die Kopie in der Zwischenablage zu erstellen)

- Die Tasten STRG und V drücken (um die Kopie in den Ordner zu schreiben)
- jetzt sollte dort eine Datei mit dem Namen python - Kopie.exe erstellt worden sein
- Taste F2 drücken um die Datei zum umbenennen vorzubereiten und dann dort den neuen Namen `python2.exe` eintragen.

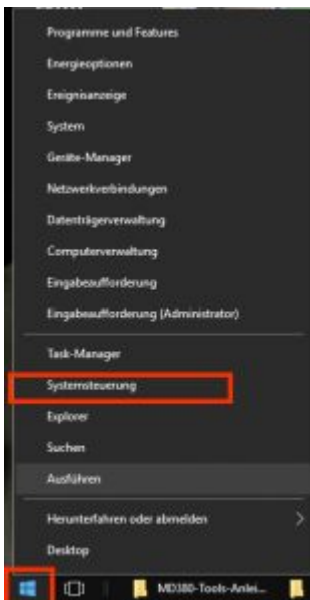
Info: Solltet ihr in eurem Verzeichnis keine Dateiendungen sehen dann natürlich auch das `.exe` am Ende vom Dateinamen nicht eintragen.

Umgebungsvariablen anpassen

Jetzt müssen wir die **Umgebungsvariablen anpassen**, wie dies unter Win XP geht steht im verlinkten Originalbeitrag, hier zeige ich euch wie das unter Windows 10 geht.

Dieser Schritt ist wichtig denn wenn hier was nicht passt dann funktioniert das Ganze nachher nicht, so war es auch bei mir gewesen.

Hier klicken wir zuerst mit der rechten Maustaste auf das Windows Icon unten links und im sich öffnenden Menü klicken wir dann auf Systemsteuerung.



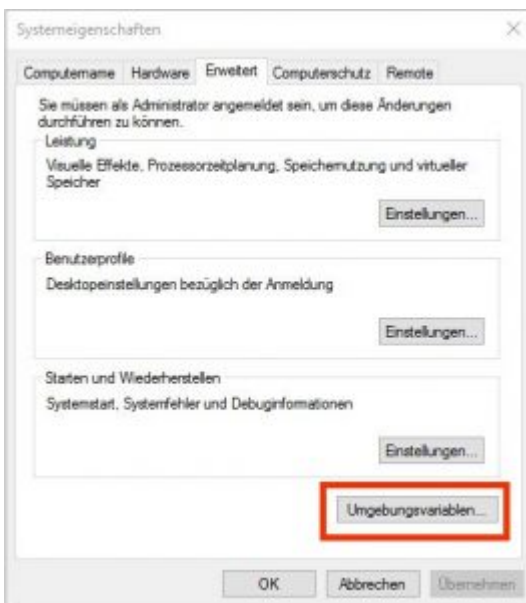
Nun öffnet sich die Systemsteuerung, dort geben wir oben rechts in das Suchfeld das Wort: „Umgebungsvariablen“ ein und klicken dann beim Suchergebnis auf „System“



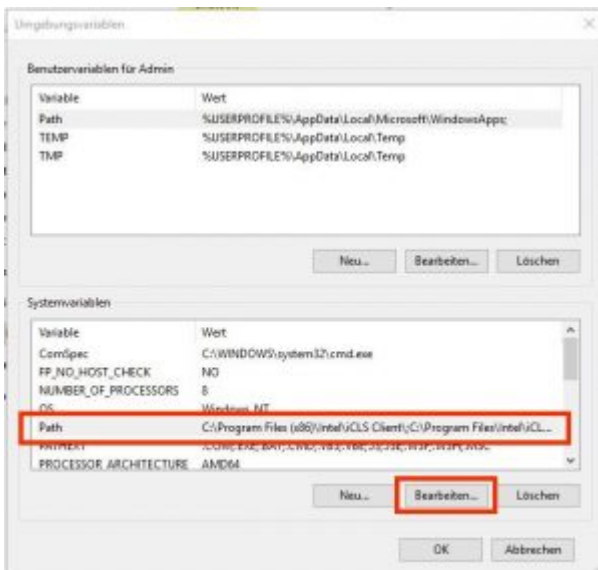
Dort dann auf „Erweiterte Systemeinstellungen“ klicken



und im nächsten Fenster dann auf „Umgebungsvariablen“



Hier wählen wir dann unten die Zeile „Path“ aus und klicken auf „Bearbeiten“



PS: Ja man kann diese Einstellungen auch nur für den gerade eingeloggten Benutzer machen das ist mir bekannt ☐

Im nächsten Fenster klicken wir oben rechts auf „Neu“ und tragen dann

Bei einem 64 Bit System:

C:\Program Files (x86)\GnuWin32\bin
ein

Bzw bei einem 32 Bit System:

C:\Program Files\GnuWin32\bin

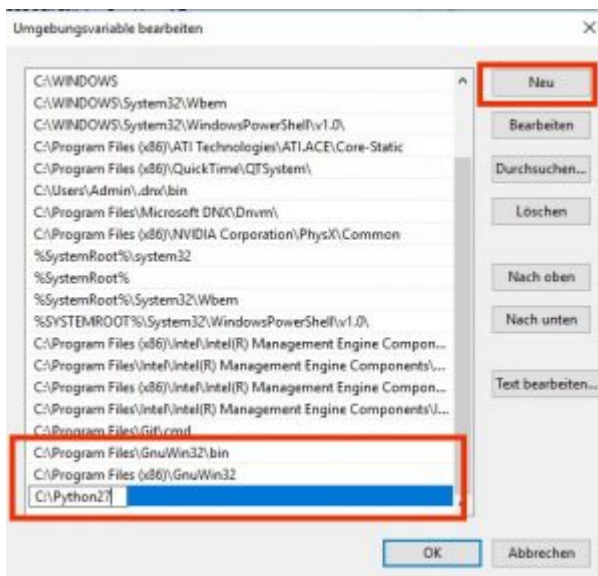
ein und bestätigen den Eintrag mit der ENTER Taste, dabei drauf achten das ihr am Anfang und am Ende keine zusätzlichen Leerzeichen habt. Dieser Path ist der Path den ich bei der Installation von Make erwähnt hatte. (mit zusätzlich einem \bin hinten dran) Theoretisch kann man hier auch den Path für beide Versionen eintragen, dann wird aber vermutlich intern irgendwo geloggt das es den Path nicht gibt.

Zusätzlich wiederholen wir diesen Schritt noch für die Python Installation, also wieder auf „Neu“ klicken und dann dort:

C:\Python27

eintragen und mit ENTER bestätigen

Info: auf dem Screenshot fehlt das \bin am Ende beim x86 Eintrag, davon nicht verwirren lassen ☐



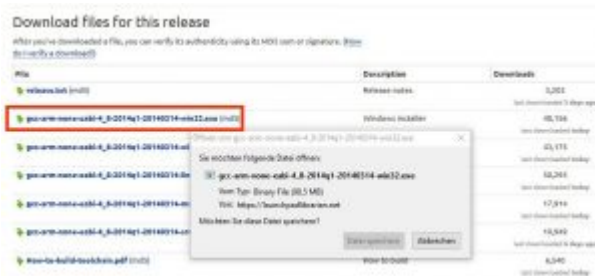
Jetzt schließen wir das Fenster mit OK und auch das andere noch offene Fenster mit OK schließen.

Das Fenster Systemeigenschaften und die Systemsteuerung mit einem Klick oben rechts auf das X schließen.

Installation von gcc-arm-none-eabi

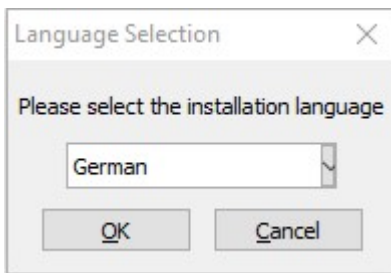
Download: <https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q1-update>

Auf der sich öffnenden Seite den Windows Installer auswählen



und die Datei wieder im erstellten Download Ordner speichern, bei mir lautete der Dateiname: gcc-arm-none-eabi-4_8-2014q1-20140314-win32.exe

Jetzt Doppelklicken wir wieder auf die eben runter geladene Datei und müssen als erstes die Installationssprache auswählen, Deutsch sollte hier schon vorausgewählt sein und wir müssen es nur noch durch einen Klick auf OK bestätigen.



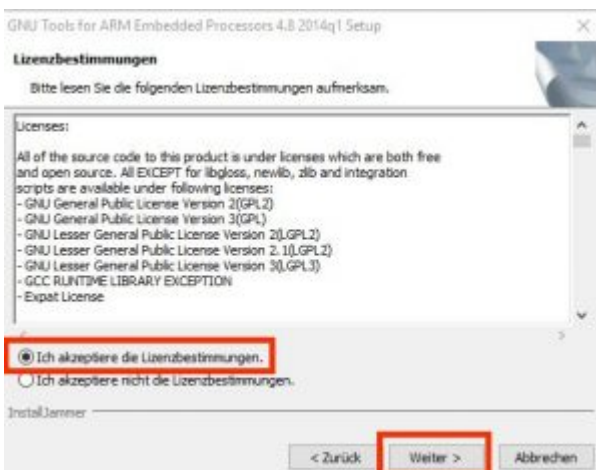
Im nächsten Fenster klicken wir auf „Ja“



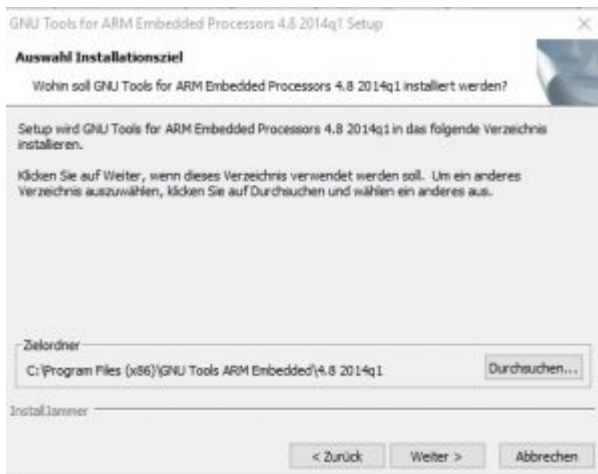
dann klicken wir auf „Weiter >“



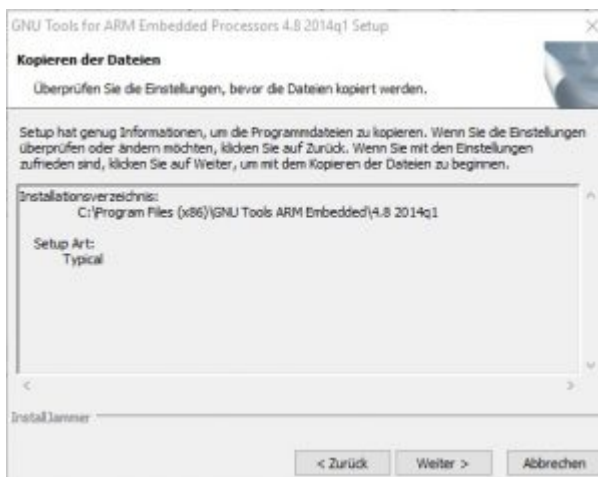
Wählen „Ich akzeptiere die Lizenzbestimmungen“ aus und klicken auf „Weiter >“



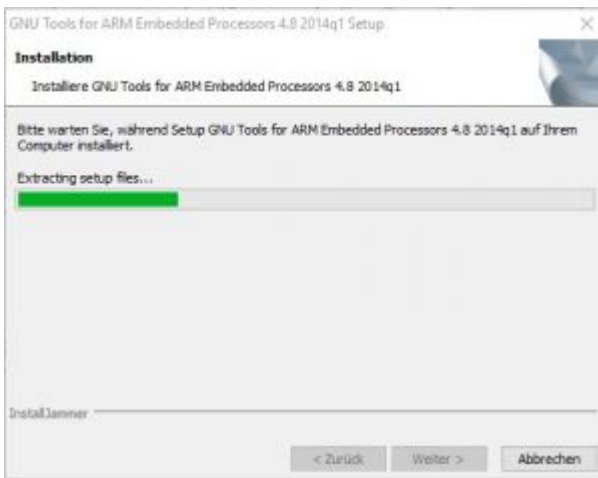
und dann bei der Auswahl des Installationsziels auch auf „Weiter >“ klicken.



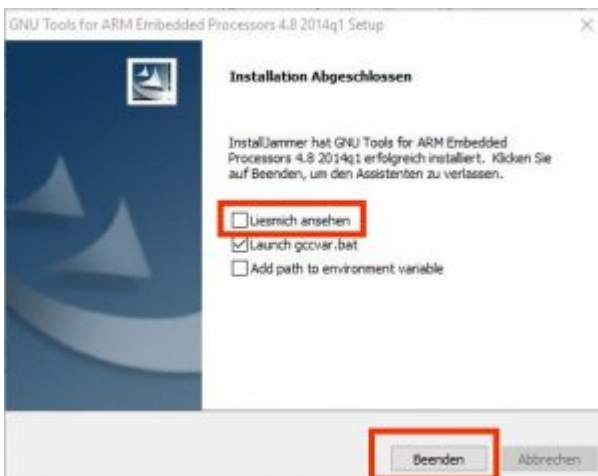
das Selbe auch auf der folgenden Seite, also wieder auf „Weiter >“ klicken.



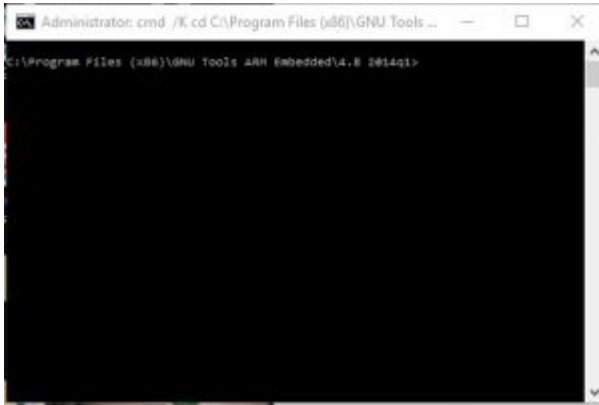
Nun startet die Installation was auch wieder einige Zeit dauern kann.



Jetzt wählen wir den Punkt „Liesmich ansehen“ (es sei denn man möchte das lesen) ab und klicken auf „Beenden“.



und schließen danach das sich öffnende Fenster mit dem X oben rechts.



Die Original Anleitung sagt das wir jetzt den PC neu starten sollen, ich versuche es diesmal ohne Neustart, aber ein Restart kann ja nie schaden ☐

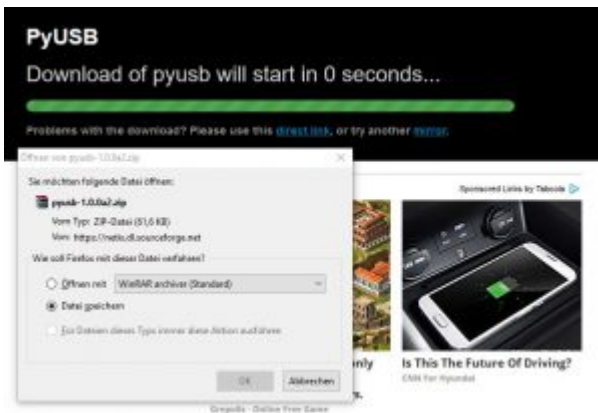
Installation von PyUSB

Download: <https://sourceforge.net/projects/pyusb/>

Auf der sich öffnenden Seite klicken wir auf den Download Button



Der Download sollte nun automatisch starten, wenn nicht wieder auf den „Direct Link“ unter dem grünen Balken klicken

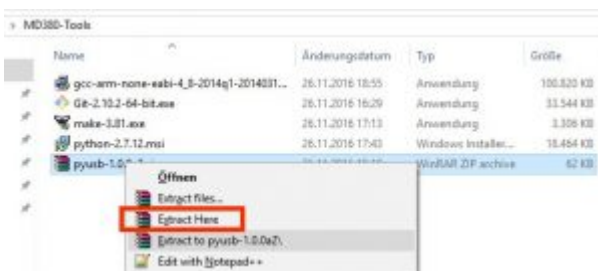


Die Datei, in meinem Fall die Datei pyusb-1.0.0a2.zip wieder im angelegten Download Ordner speichern.

Da diese Datei eine gepackte Datei ist (zip Datei) müssen wir sie nun entpacken, solltet ihr keinen Dateientpacker installiert haben (wobei Windows hat glaube ich inzwischen einen eigenen) dann installiert euch einfach WinRAR

<https://www.winrar.de/download.php>

OK weiter im Text, nun machen wir einen Rechtsklick auf die Zip-Datei (pyusb-1.0.0a2.zip) und wählen den Punkt „Extract Here“ aus, falls ihr ein anderes Entpackprogramm als WinRAR habt kann der Punkt durchaus etwas anders lauten.



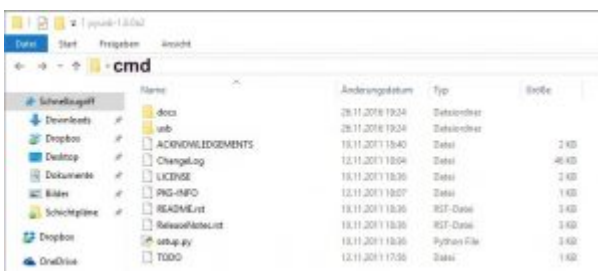
dadurch wird ein neues Verzeichnis mit dem Namen „pyusb-1.0.0a2“ erstellt. Auch hier kann das Verzeichnis später wieder etwas anders lauten wenn ihr eine neuere Version nutzt.

Jetzt wird es wieder etwas kryptischer denn wir müssen mit der Kommandozeile arbeiten.

Wir wechseln mit dem Windows Explorer (nicht Internet Explorer) in das eben durch das

Entpackprogramm erstellte Verzeichnis, darin sollte sich jetzt eine Datei mit dem Namen setup.py befinden.

Jetzt klicken wir in die Adresszeile vom Explorer und geben dort „cmd“ ein und bestätigen die Eingabe mit der ENTER Taste. (leider kann man davon keinen Screenshot machen, deswegen habe ich den zusammen gebastelt)



Nun öffnet sich die Kommandozeile und wir sind automatisch im richtigen Verzeichnis.

Dort geben wir nur folgendes ein:

`python setup.py install`

und drücken wieder ENTER



danach erscheinen einige Zeilen im Kommandofenster, das Fenster jetzt durch einen Klick auf das X oben rechts oder durch die Eingabe von „exit“ schließen

```

C:\Windows\System32\cmd.exe
copying build\lib\usb\debug.py -> C:\Python27\Lib\site-packages\usb
copying build\lib\usb\interop.py -> C:\Python27\Lib\site-packages\usb
copying build\lib\usb\__init__.py -> C:\Python27\Lib\site-packages\usb
byte-compiling C:\Python27\Lib\site-packages\usb\backend\libusb1.py to libusb10
.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\backend\libusb10.py to libusb10
.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\backend\openusb.py to openusb.0
.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\backend\__init__.py to __init
.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\control.py to control.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\core.py to core.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\legacy.py to legacy.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\util.py to util.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\debug.py to debug.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\interop.py to interop.pyc
byte-compiling C:\Python27\Lib\site-packages\usb\__init__.py to __init__.pyc
running install_egg_info
writing C:\Python27\Lib\site-packages\pyusb-1.0.0a2-py2.7-egg-info
C:\Users\Admin\Desktop\MD380-Tools\pyusb-1.0.0a2>

```

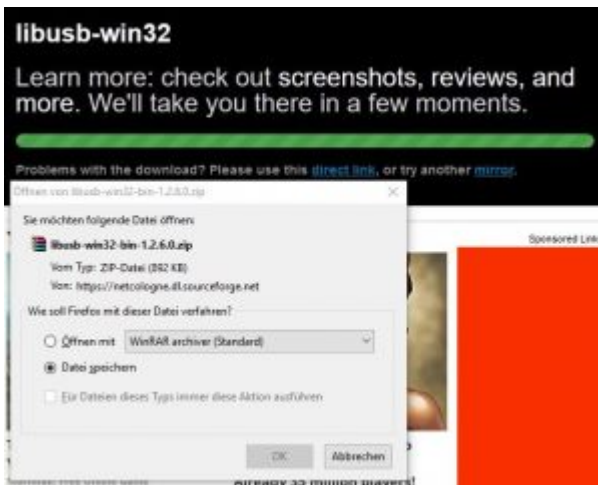
libusb-win32 installieren

Download: <https://sourceforge.net/projects/libusb-win32/>

auf der sich öffnenden Seite wieder auf den Download Button klicken.

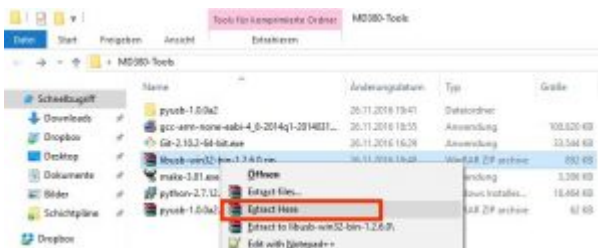


Auch hier sollte der Download wieder automatisch starten, wenn nicht dann wieder unter dem grünen Balken auf den „Direct Link“ klicken.



Die Datei (bei mir libusb-win32-bin-1.2.6.0.zip) wieder in unserem erstellten Download Ordner speichern.

Da es wieder eine ZIP-Datei ist müssen wir diese auch wieder entpacken, also wieder einen Rechtsklick auf die Datei libusb-win32-bin-1.2.6.0.zip und „Extract Here“ auswählen um die Datei zu entpacken.

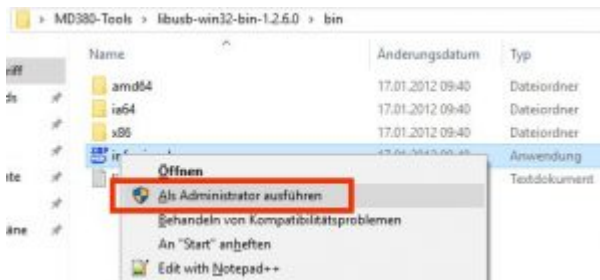


Nun wechseln wir wieder mit dem Windows Explorer in das eben durch das Entpackprogramm erstellte Verzeichnis „libusb-win32-bin-1.2.6.0“ und dann in das sich darin befindende Verzeichnis „bin,,“.

In dem Verzeichnis sollte sich eine Datei mit dem Namen „inf-wizard.exe“ befinden.

Das Programmierkabel in den Computer und das Funkgerät stecken, danach das MD380 bei gedrückter PTT-Taste und gleichzeitig gedrückter oberer Funktionstaste einschalten um es in den Flash-Modus zu schalten.

Auf die Datei „inf-wizard.exe“ machen wir wieder

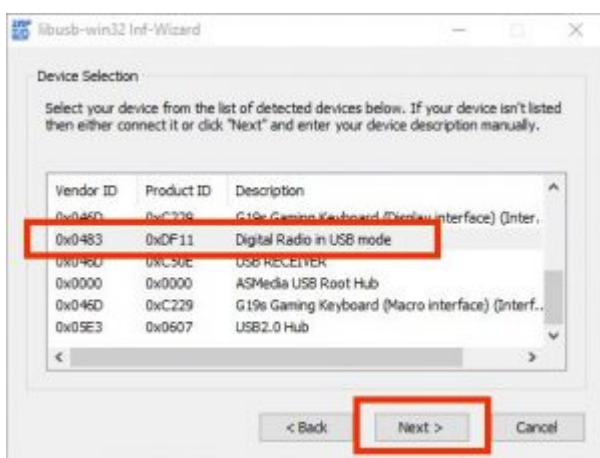


Es öffnet sich folgendes Fenster

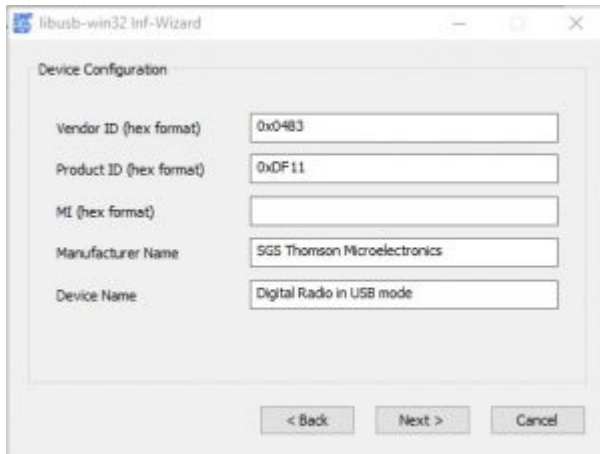


dort klicken wir auf „Next >“ (TRX muss wie oben in rot 'beschrieben verbunden sein)

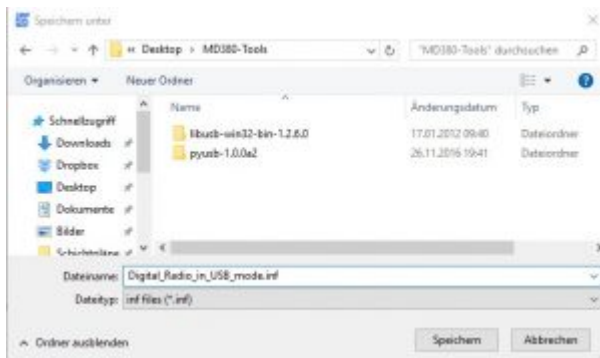
Jetzt wählen wir den Punkt „Digital Radio in USB mode“ aus und klicken auf „Next >“



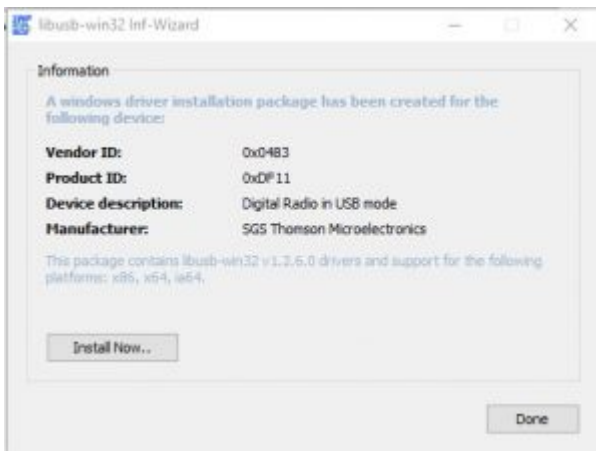
die nächste Seite bestätigen wir auch einfach mit „Next >“



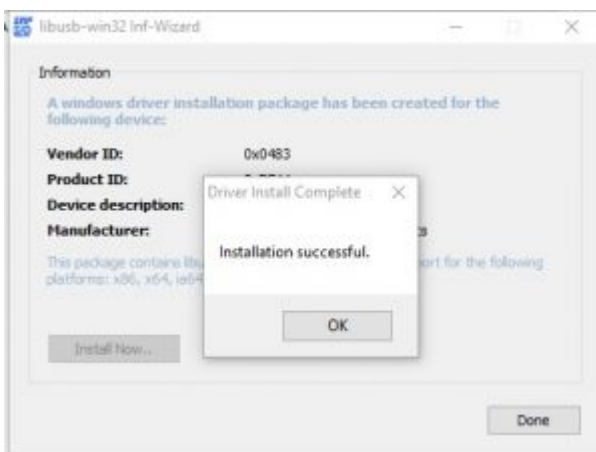
Jetzt wechseln wir im „Speichern unter“ Dialog in unseren erstellten Download Ordner und speichern dort die Datei `Digital_Radio_in_USB_mode.inf` (*parallel dazu wird auch die Datei `Digital_Radio_in_USB_mode.cat` erstellt*) durch einen Klick auf den Button „Speichern“



danach erscheint folgendes Fenster wo wir auf „Install Now..“ klicken.



Wenn die Installation erfolgreich war erscheint folgendes Fenster



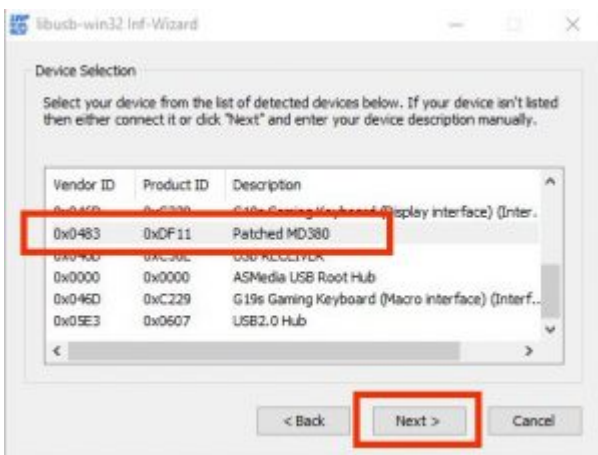
Solltet ihr hier eine Fehlermeldung bekommen, dazu gehört auch das gelbe Ausrufezeichen, dann habt ihr das Programm vermutlich nicht als Administrator gestartet.

Nun beenden wir das Programm mit einem Klick auf den OK Button.

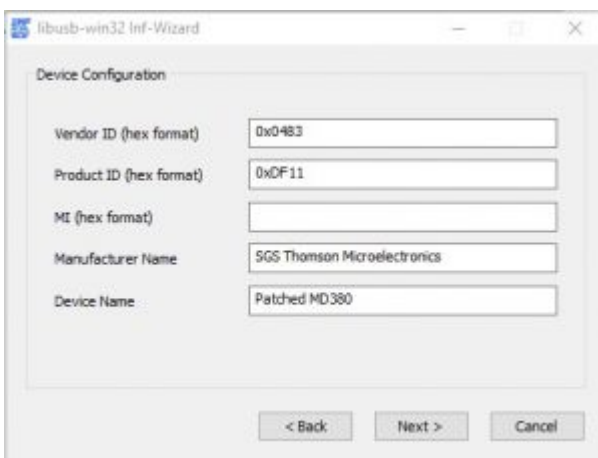
Jetzt schalten wir den TRX wieder aus und direkt wieder ein, dieses mal aber OHNE die PTT und den oberen Button gedrückt zu halten, also einfach nur ganz normal einschalten wie auch im normalen Betrieb, dadurch schalten wir das MD380 in den USB Modus.

Jetzt durchlaufen wir die ganze Prozedur noch einmal, also wieder ein Rechtsklick auf die Datei „inf-wizard.exe“ und wählen „Als Administrator ausführen“ aus.

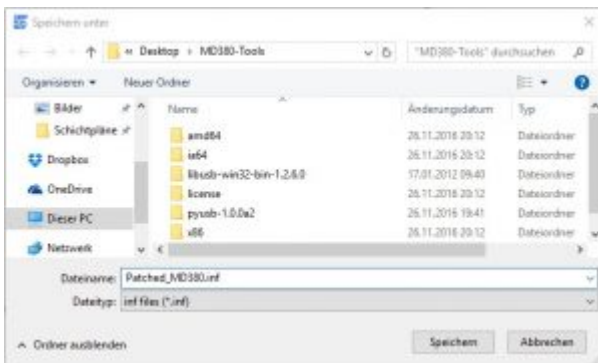
Als erstes wieder „Next >“ auswählen, danach aber jetzt den Punkt „Patched MD380“ auswählen und auf „Next >“ klicken.



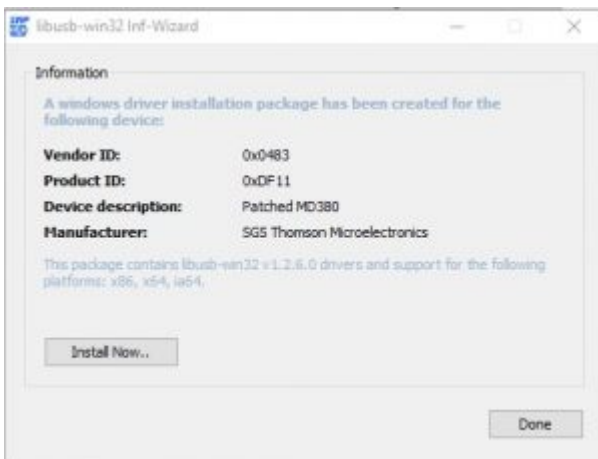
Im nächsten Fenster wieder auf „Next >“ klicken



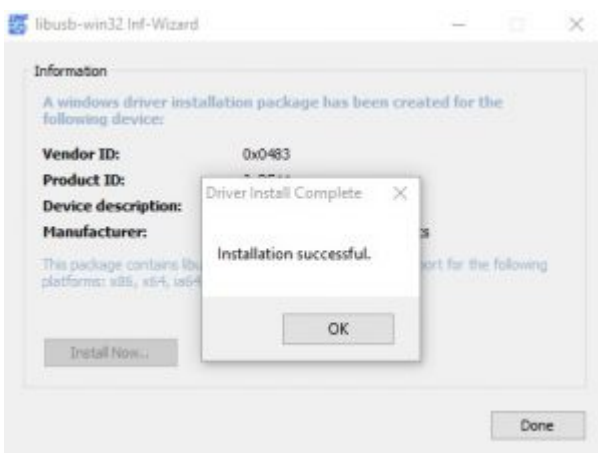
und wie beim letzten mal auch wieder zum speichern in unseren erstellten Download Ordner wechseln und diesmal die Datei „Patched_MD380.inf“ (parallel dazu wird wieder die Datei Patched_MD380.cat erstellt) durch einen Klick auf den Button „Speichern“ speichern.



Jetzt wieder auf „Install Now..“ klicken



und auch dieses mal wird die erfolgreiche Installation mit folgender Meldung angezeigt



Solltet ihr hier eine Fehlermeldung bekommen, dazu gehört auch das gelbe Ausrufezeichen, dann habt ihr das Programm vermutlich nicht als Administrator gestartet.

Nun beenden wir das Programm mit einem Klick auf den OK Button und schalten den TRX wieder aus.

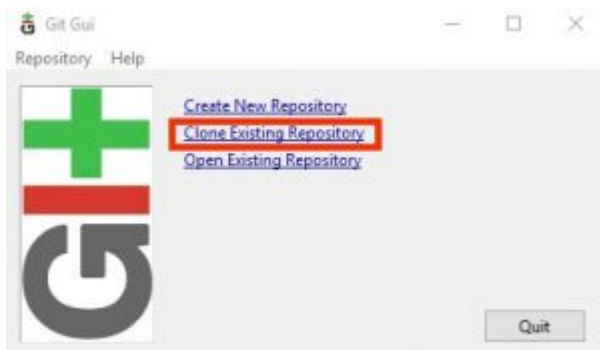
Lokale Repository-Kopie erstellen

Hiermit laden wir den „Entwicklercode“ aus dem Internet auf unseren lokalen PC runter um daraus die Firmware zu erstellen. (kompilieren) Zu diesem Zweck haben wir das Programm Git installiert.

Jetzt klicken wir unten links auf das Windows Logo, scrollen dann zu G und öffnen den Eintrag „Git“ und wählen dort dann „Git GUI“ aus.



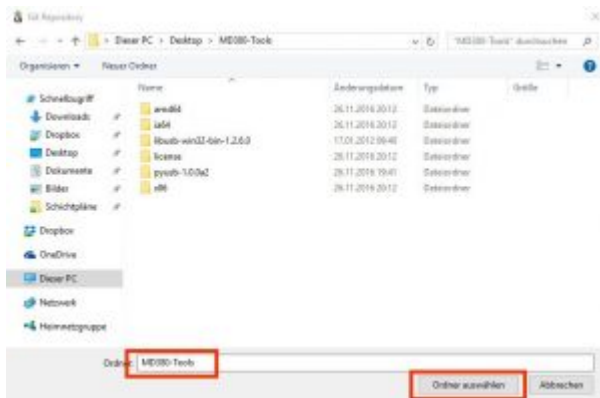
danach startet das Programm Git, dort klicken wir dann auf „Clone Existing Repository“ (das ist der mittlere Eintrag)



Bei Source Location tragen wir folgendes ein:
<http://github.com/travisgoodspeed/md380tools>

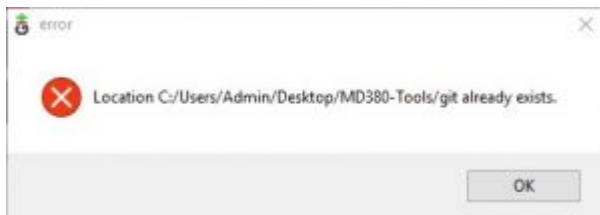
Bei Target Directory tragen wir unseren lokalen Ort ein wo die Dateien hin geladen werden sollen. Dazu klicken wir rechts bei Target Directory auf den Button Browse und wechseln wieder in unseren erstellten Download Ordner.

Jetzt stellen wir sicher Ordner auch ausgewählt ist, wenn nicht dann markieren wir den Ordner mit einem einfachen Mausklick und klicken dann auf den Button „Ordner auswählen“



Nachdem wir den Button „Ordner auswählen“ geklickt haben schreiben wir hinter der Verzeichnisangabe bei der „Target Directory“ noch ein „/git“

(das Programm ist da irgendwie fehlerhaft, man kann keine existierenden Verzeichnisse auswählen)

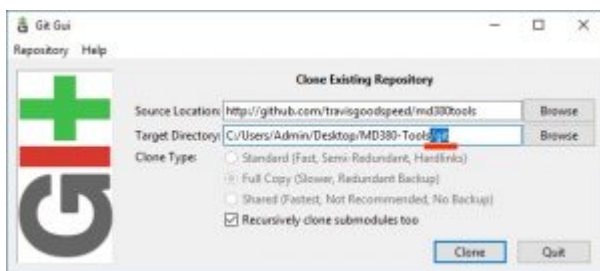


Also falls ihr die oben stehende Fehlermeldung bekommt einfach in der Target Directory das Verzeichnis ändern auf einen Ordner den es noch nicht gibt.

Das Ganze schaut dann in etwa so aus

`C:/Users/Admin/Desktop/MD380-Tools/git`

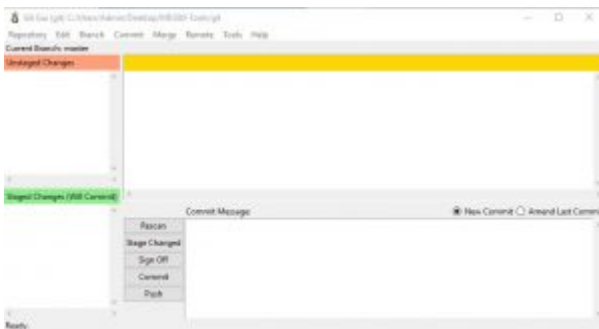
natürlich lautet eure Target Directory anders. Jetzt klicken wir auf den Button „Clone“



und das Programm Git lädt die Dateien runter, was je nach Internetverbindung einige Zeit dauern kann.



Irgendwann erscheint dann dieses Fenster was wir dann mit einem Klick auf das X rechts oben wieder beenden können.



So das war es mit der Installation gewesen, waren ein Haufen Programme und auch viel Arbeit.

.... Anleitung ist noch nicht fertig ...