MMDVM – Multi-Mode Digital Voice Modem

Das mmdvm ist ein MultiMode Digital Voice Modem und besteht aus einem Rechner einen Arduino DUE und einer Platine für die Ankopplung des Funkgerätes. Die Platine wird an die „9k6“ Buchse angeschlossen, bzw an die „Digital“ Ein/Ausgänge der Betriebsfunkgeräte.
Quelle BM

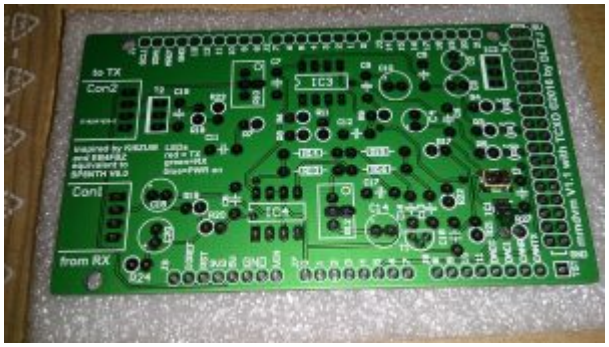
Ich hatte schon vor einer ganzen Weile vom MMDVM Projekt gehört, aber da ich ja bereits einen Repeater habe war das Projekt für mich nicht ganz so interessant.

Jetzt durch den Testbetrieb im BM Netz bin ich durch die OMs in der SysOp Gruppe wieder drauf aufmerksam geworden und habe mir pauschal mal den Bausatz bei DL7TJ bestellt.
Diese Anleitung bezieht sich somit auf genau diesen Bausatz.

Wer den Bausatz beziehen möchte kann sich bei DL7TJ per E-Mail melden: mmdvm@t-online.de oder sich hier in der Facebookgruppe melden sofern man einen FB Account hat: https://www.facebook.com/groups/1187868361237026/?fref=ts

Natürlich sollte jedem klar sein das wenn ihr der Anleitung folgt dies auf eigenes Risiko ist,ich übernehme keine Haftung für Schäden jeglicher Art.

So schaut nur die Platine aus:



~~Als TRX wird ein Yaesu FT 7800 eingesetzt welchen ich hier noch habe. Als 2. TRX muss ich schauen ob mein Icom IC 2820 dafür geeignet ist, eine Databuchse mit 9,6k hat er jedenfalls.~~

~~Für die ersten Tests ist das dann erst einmal ausreichend, falls ich das fertige Projekt dann mal an einen anderen Standort (oder hier bei mir zusätzlich) in den Live-Betrieb schicke dann werde ich wohl ein Betriebsfunkgerät benötigen um die Ausgangsleistung an die 15 Watt ERP anpassen zu können ... oder mit Dämpfungsglied ... da muss man dann halt schauen was besser ist.~~

EDIT: inzwischen sind es 2 Motorola GM340 geworden

OK da mir zum Löten noch ein paar Gerätschaften fehlen die ich erst bestellt habe kann ich mich schon mal mit dem Softwarepart beschäftigen.

Die MMDVM Platine sitzt auf einem Arduino DUE welcher wiederum mit einem Raspberry PI verbunden ist.

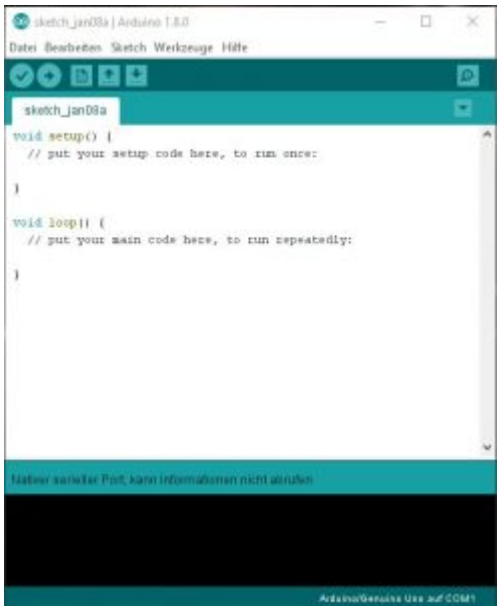
Auf dem Raspberry PI muss später ein Linux Image rauf und die MMDVM Software. Auch den Arduino muss man erst mal programmieren. Während ich mit dem PI schon gearbeitet habe und auch 2 davon hier rum zu liegen habe ist der Arduino dann doch noch komplett neu für mich.

Hierzu laden wir uns hier die Arduino Software runter
https://www.arduino.cc/en/Main/Software
Aktuell ist die Version ARDUINO 1.8.0 aktuell, das kann sich aber später ändern.

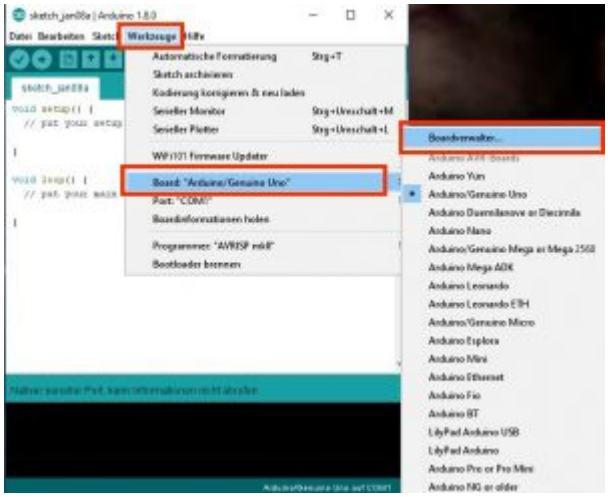
Da ich Windows Benutzer bin nehme ich die Windowsversion (Windows-Installer), zu den anderen Versionen und dem gefriemel bei MAC oder LINUX kann ich somit nichts sagen. Auf der Downloadseite nicht erschrecken, da stehen Geldbeträge aber das sind freiwillige Spenden.

Während der Installation muss man noch 2 Mal die Installation der USB Treiber extra bestätigen.

Nun starten wir die Arduino Software

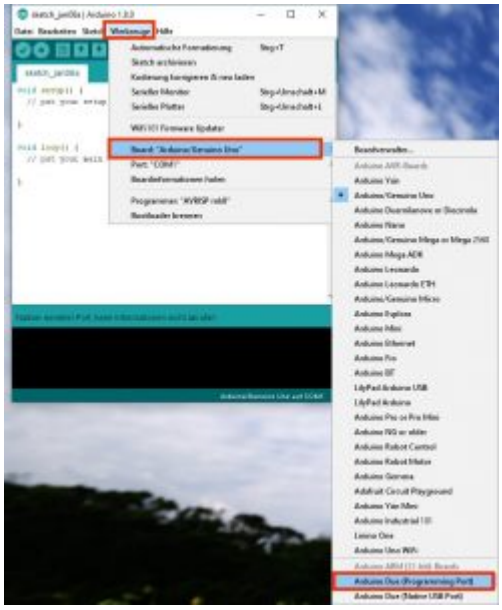


Jetzt klicken wir auf
Werkzeuge –> Board: —> Boardverwalter



und wählen dort das Arduino SAM Board aus.
Hier gibt es 2 Einträge für das SAM Board, wir benötigen den für den Arduino Due. Bei diesem Eintrag klicken wir jetzt auf Installieren.

Als nächstes wählen wir unter
Werkzeuge –> Board: –> Arduino Due (Programming Port)
aus



Als nächstes müssen wir eine Textdatei (platform.txt) anpassen, diese befindet sich im Verzeichnis:
C:\Users\Admin\AppData\Local\Arduino15\packages\arduino\hardware\sam\1.6.10
Admin –> durch euren PC Benutzernamen ersetzen
1.6.10 –> durch die gerade aktuelle Version von SAM ersetzen

Um die Datei jetzt vernünftig bearbeiten zu können benötigen wir einen Texteditor mit Zeilenanzeige und viel wichtiger einen der auch die Zeilenumbrüche darstellt. Dies ist beim normalen Notepad zumindest bei mir nicht der Fall.

Falls ihr nur Notepad habt am besten Notepad++ installieren welches man hier downloaden kann:
https://notepad-plus-plus.org/
Nach der Installation von Notepad++ machen wir einen Rechtsklick auf die Datei platform.txt im oben ausgewählten Verzeichnis und wählen „Edit with Notepad++“ aus.

Jetzt müssen wir in der platform.txt nach diesem Text suchen:

```
## Combine gc-sections, archives, and objects
```

welcher sich in der SAM Version 1.6.10 in Zeile 80 befindet.
Eine Zeile drunter befindet sich eine sehr lange Zeile die bei mir wie folgt ausschaut

```
recipe.c.combine.pattern="{compiler.path}{compiler.c.elf.cmd}"
-mcpu={build.mcu} -mthumb {compiler.c.elf.flags}
"-T{build.variant.path}/{build.ldscript}"
"-Wl,-Map,{build.path}/{build.project_name}.map"
{compiler.c.elf.extra_flags} -o "{build.path}/{build.project_name}.elf"
"-L{build.path}" -Wl,--cref -Wl,--check-sections -Wl,--gc-sections
-Wl,--entry=Reset_Handler -Wl,--unresolved-symbols=report-all
-Wl,--warn-common -Wl,--warn-section-align -Wl,--start-group
{compiler.combine.flags} {object_files}
"{build.variant.path}/{build.variant_system_lib}"
"{build.path}/{archive_file}" -Wl,--end-group -lm -gcc
```

das kann jetzt bei euch natürlich evtl in einer anderen Version wieder anders ausschauen.

Am Ende dieser Zeile (in meinem Fall Zeile 81) sollte folgendes stehen

```
"{build.path}/{archive_file}" -Wl,--end-group -lm -gcc
```

DAVOR fügen wir jetzt den folgenden Code ein

```
"{build.system.path}/CMSIS/CMSIS/Lib/ARM/arm_cortexM3l_math.lib"
```

incl. den Anführungszeichen und so das jeweils zum Code davor und Code danach ein Leerzeichen vorhanden ist.

Jetzt speichern wir die Datei platform.txt und beenden Notepad++

Als nächstes benötigen wir die MMDVM Software, diese bekommt man hier:
https://github.com/g4klx/MMDVM/archive/master.zip

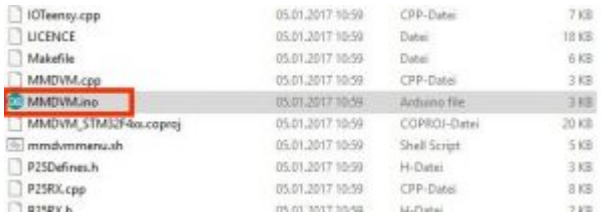
Wir speichern diese Datei nun auf unserem PC im Ordner Amateurfunk oder wo auch immer, der Verzeichnisname ist dabei nicht.

Jetzt entpacken wir die Datei, wer kein Entpackungsprogramm hat bekommt hier WinRAR kostenlos und in deutsch (nur ohne Originalverpackung *HI*)
https://www.winrar.de/downld.php

Als nächstes benennen wir den Ordner MMDVM-master in MMDVM um, dieser Schritt ist notwendig da sonst das Projekt nicht korrekt geöffnet werden kann.

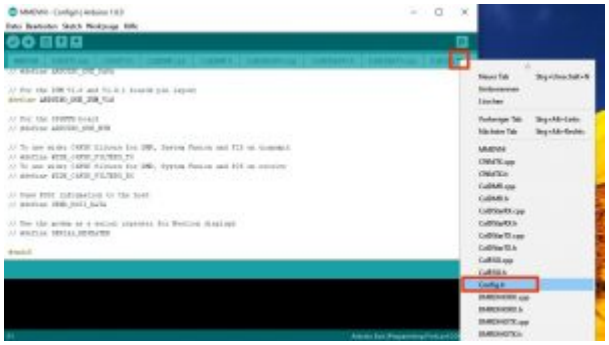
Jetzt wechseln wir in den Ordner MMDVM und starten unser Projekt mit einen Doppelklick auf die Datei MMDVM.ino.

Wer die Dateiendungen ausgeschaltet hat, es handelt sich dabei um diese Datei:



worauf sich die Arduino Software öffnet welche wir vorhin eingerichtet haben.

Als nächstes müssen wir die Datei Config.h bearbeiten, dazu klicken wie bei den „TABs“ ganz rechts auf den Pfeil nach unten (siehe Screenshot) und wählen dann Config.h aus.



Jetzt suchen wir darin nach

```
#define ARDUINO_DUE_ZUM_V10
```

und ersetzen es durch

```
// #define ARDUINO_DUE_ZUM_V10
```

als nächstes suchen wir nach

```
// #define ARDUINO_DUE_NTH
```

und ersetzen es durch

```
#define ARDUINO_DUE_NTH
```

sollte danach dann so aussehen

Jetzt passen wir noch die TCXO Frequenz an und suchen in der Config.h nach

```
// #define EXTERNAL_OSC 19200000
```

und ersetzen es durch

```
#define EXTERNAL_OSC 19200000
```

sollten dann so aussehen

Als nächstes klicken wir auf
Sketch –> Überprüfen/Kompilieren

jetzt arbeitet das Programm einige Sekunden, also einfach etwas warten.

Sobald das Programm fertig ist erhalten wir diese Meldung (kann bei euch natürlich leicht abweichen)

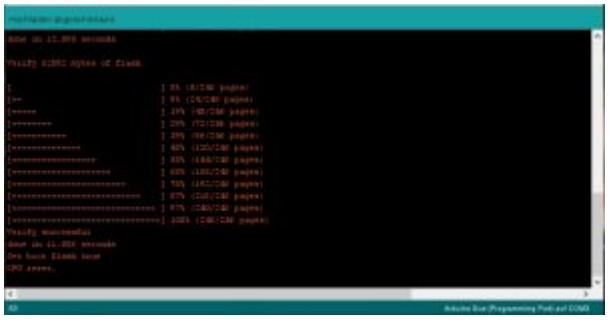
Jetzt verbinden wir den Arduino Due mit dem „Programming Port„, das ist der Port bei der „Dicken schwarzen Buchse“ und einem USB Port am PC.

Nun geht es darum unser „MMDVM Programm“ auf den Arduino Due zu laden. Dazu klicken wir oben links auf den Pfeil nach rechts.

Sollten wir hier folgenden Fehler bekommen

dann bitte die Einstellungen für den COM Port überprüfen so das auch der Programming Port ausgewählt ist.

Danach sollte das „Hochladen“ durchlaufen, dieser Vorgang dauert auch einige Sekunden und wird in etwa wie folgt angezeigt sofern er beendet ist.



Das war es dann mit der Einrichtung des Arduino Due.

Ich werde die Anleitung erweitern sobald ich weiter dran arbeite.