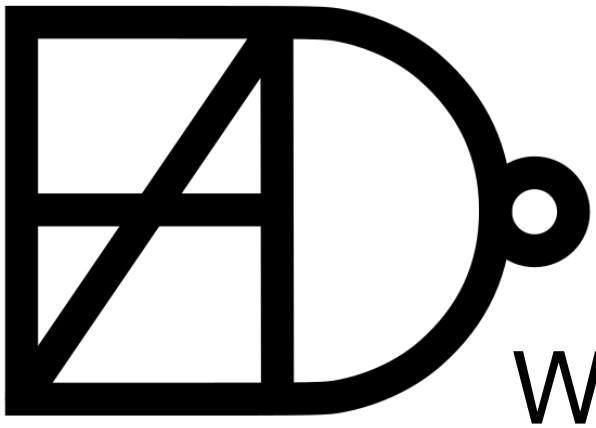
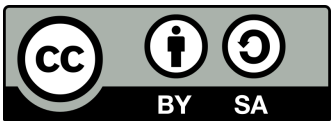


WA0EDA

STM32-DVM-MTR2K v2.0c/d Product Update

Revision: 20230121

Table Of Contents



Copyright © 2023 The Regents of the K0USY Group
This document is licensed under the CC BY-SA 4.0
https://creativecommons.org/licenses/by-sa/4.0/

The WA0EDA Skunkworks may be contacted at:
5505 Plymouth Drive
Lawrence, KS 66049
n0mjs@me.com
(telephone support not provided)

Disclaimers and Conditions of Use

3rd Party Software Disclaimer

The STM32-DVM-MTR2K contains various 3rd party softwares redistributed under various versions of the GPL. WA0EDA and the Regents of the K0USY Group make no claims or warranties regarding such software and we are not responsible for its use. Using the STM32-DVM-MTR2K requires acceptance of and compliance with the license terms of included 3rd party software.

Use at Your Own Risk

The STM32-DVM-MTR2K, as a whole unit, is not UL (or similarly) listed, and has not been tested for FCC Part 15 (or other) compliance. By using the STM32-DVM-MTR2K, the user agrees to indemnify the WA0EDA Club, the Regents of the K0USY Group, Cortney T. Buffington, N0MJS and other affiliated entities against liability resulting in its use. The STM32-DVM-MTR2K is intended for use in experimental and educational environments consistent with the amateur radio service.

Configuration Disclaimer

WA0EDA provides configuration advice, but is not responsible or liable for providing comprehensive or authoritative information about MMDVM or the MTR2000. Configurations presented in this manual represent how we set up MMDVM and the MTR2000 with the STM32-DVM-MTR2K. This manual does not attempt to cover alternative configurations, and is not an authoritative source of information for MMDVM or the MTR2000.

Update

As a mature product the STM32-DVM-MTR2K has undergone a few changes since Version 2.0c was released. This update explains the differences in versions and provides schematic and board layouts for the v2.0d version.

The version 2.0c product manual should be considered the primary source document, and only specific items mentioned in this update are changed.

I/O Processor Changes

Due to challenges in the supply chain for the original ATmega328P/PA chip used in the original v2.0c modem, the I/O processor was updated on most boards with serial numbers larger than 20210100 to an ATmega328PB. The 328PB is superior to the 328P/PA part in most ways, but does not program as easily in the Arduino ecosystem.

Modems with the ATmega328PB contain the Optiboot bootloader rather than the “original” Arduino bootloader. Optiboot is both faster and smaller, but requires different settings to update/program firmware. If using the original “update_io.sh” script the line:

```
avrdude -c arduino -p ATMEGA328P -P /dev/ttyS2 -b 57600 -U flash:w:./"$1"
```

Must be changed to:

```
avrdude -c arduino -p ATMEGA328PB -P /dev/ttyS2 -b 115200 -U flash:w:./"$1"
```

Starting with OS images dated 2023-01-21 and newer, two scripts are provided in the “mmdvm” user’s home directory rather than the older “update_io.sh”. They are:

- update_io_pa.sh
- update_io_pb.sh

Each for their respective part.

Hardware Changes in v2.0d

The v2.0d PCB contains minor changes:

- Additional jumper pads to select the STM32F405 or STM32F446 processors
- FM mode LED
- Additional pads for alternative DC-DC converter
- LED enable jumper removed

Processor Jumper Pads

A change in the silicon of the STM32F4xx series microcontrollers changed the behavior of entering bootloader mode (necessary to update MMDVM firmware). In this case, additional jumpers JP14 and JP15 were added to the "upper left" of the STM32F4xx modem processor. They are labelled "405" and "446" respectively, and one or the other should be bridged based on the part used (eg. If an STM32F405 is used, the JP14, "405" will be solder-bridged).

FM Mode LED

Analog FM support is now mature in MMVDM, and an "FM" mode LED was added to the v2.0d PCB. Earlier versions of the STM32-DVM-MTR2K support FM if configured to do so. The only change is the addition of a mode LED to indicate FM operation.

DC-DC Converter

Supply chain issues have, at times, affected the availability of quality "mini360" DC-DC converters (main 5V DC regulator) for the board. Additional pads were added to alternatively mount an LM2596-based module, also a popular and "generic" part, just like the mini360. This module has been tested extensively and performs as well or better than the mini360, though it is significantly larger and marginally less efficient due to a lower switching frequency.

LED Enable Jumper

It was determined that there was no real necessity to disable the MMDVM LEDs on the modem, and therefore the LED Enable jumper of previous versions was eliminated. This also gave room to spread out the LEDs on the v2.0d revision to separate power and activity, the mode LEDs, and the Status LEDs

FM Mode

In order to operate in FM mode, WA0EDA recommends using the analog handshake facilities included to operate with analog repeat built into the MTR2000, but reconfigured to pass COS status to the MMDVM. This requires modifying the wildcard tables in the station to send COS signal to "GPIO2", and jumping (vertically) the right two pins of JP8 "Analog Handshake Enable". By configuring COS in MMDVM.ini to be a true COS and not a lockout, this provides a stable COS signal to MMDVM's built-in analog FM control routines.

MMDVMhost and MMDVM

WA0EDA does not provide documentation for MMDVMHost and MMDVM beyond what is included in the main manual. These are both 3rd party, open source software products under constant development, and contain their own documentation. MMDVM firmware may be compiled from the main MMDVM repository for the WA0EDA modems directly, as we include updates to the MMDVM repository to support the firmware for our modems.

Software and Utilities

WA0EDA's OS Image for the embedded NanoPi NEO single-board computer contains versions of MMDVMHost, MMDVM firmware and IO firmware. WA0EDA does not provide support for the used of our modems with PiStar. PiStar works, and many of our customers choose it. We do not support it because we've never used it.

In the mmdvm user's home directory, a listing should look similar to that in figure 1 below.

```
mmdvm@stm32-dvm-mtr2k:~$ ls
io_firmware           mmdvm-2022-08-18  supporting-2022-08-18  update_io_pb.sh  WiringNP
mk_mmdvm_build_env.sh  mmdvm-2023-01-10  update_io_pa.sh        update_modem.sh
mmdvm@stm32-dvm-mtr2k:~$
```

Figure 1: /home/mmdvm directory listing

Directories labelled "mmdvm-{yyyy-mm-dd}" contain a precompiled version of MMDVMhost, mmdvm firmware for the processor for both the STM32F405 and STM32F446 chips, and the base MMDVM.ini file with options for the MMDVMHost build.

The directory "supporting-{yyyy-mm-dd}" contains additional software and data files used in the STM32-DVM-MTR2K. Some files, like INI files are examples and will need to be changed for the individual user's needs.

Files ending in .service may be used with systemd to automatically stop/start the services at boot time.

WA0EDA, by convention uses /usr/local/bin for binaries, /usr/local/etc for non executables, /etc for daemon .ini files, and /var/log/mmdvm for log files.

It is expected that users who use WA0EDA's OS image have basic UNIX CLI skills and understand how to copy, move, edit files, change permissions, etc. Support for basic UNIX CLI is not provided. Users who are uncomfortable with the UNIX CLI should consider using PiStar instead.

It is also probable that at some point, you will need to download and compile MMDVM or MMDVMHost yourself. WA0EDA modems require no special experience to accomplish this, but some additional libraries are needed, we recommend running the "mk_mmdvm_build_env.sh" script prior to attempting to compile MMDMV firmware as an easy way to install dependencies.

V2.0d Jumpers and Connections

STM32-DVM-MTR2K Connections:

ID	Name	Purpose	Type	Description
J1	MTR2K J2	MTR2000	96 Pin (3x32) Eurocard	Connects STM32-DVM-MTR2K to the MTR2000
J2	AVR_SPI_ICSP	Programming	.1” Pin Header 2x3	In-circuit SPI programmer for the ATmega328P
J3	I/O SERIAL	Programming	.1” Pin Header 1x6	Serial communication/programming for the ATmega328P
JP10	RESET	Programming	.1” Pin Header 1x2	Used to reset the MODEM, used with boot loader mode
JP11*	AVR RESET	Programming	.1” Pin Header 1x2	Connects ATmega328P reset to the station reset line
JP2	EXT PWR +7-15V	Testing/ Programming	.1” Pin Header 1x2	External power for the STM32-DVM-MTR2K while not inserted into an MTR2000
JP3	STM32 ICSP	Programming	.1” Pin Header 2x3	In-circuit SPI programmer for the STM32F4xx
JP4	MODEM DISP	Peripheral	.1” Pin Header 1x4	Serial “repeater” connection via the MODEM
JP5	MODEM SERIAL	Testing/ Programming	.1” Pin Header 1x5	Exposes serial (/dev/ttyS1) connection between the MODEM (STM32F4xx) and HOST (NanoPi NEO)
JP6	SBC DISP	Peripheral	.1” Pin Header 1x4	Serial (/dev/ttyS2) connection from the Host (NanoPi NEO) for serial display, peripheral or communication with the ATmega328P I/O processor
JP7**	I/O SERIAL EN.	Peripheral/ Programming	.1” Pin Header 2x2	Connects NanoPi NEO (/dev/ttyS2) to the ATmega328P I/O processor serial port
JP8***	ANALOG HANDSHAKE ENABLE	Peripheral	.1” Pin Header 2x2	Used as listed in primary manual, or INHIBIT_R to GPO_2 may be used to send COS (GPO_2) to the MMDVM INHIBIT_R input.
JP9	BOOT	Programming	.1” Pin Header 1x2	Used to place the MODEM into boot loader mode
JP12	405/446	Chip Selection	Solder Pad	Solder jumper center pad to 405 or 446 pads
JP13	446	Chip Selection	Solder Pad	Solder jumper if 446 used
JP14	405	Chip Selection	Solder Pad	Solder jumper if 405 used
JP15	446	Chip Selection	Solder Pad	Solder jumper if 446 used
LSP1	3.3V DC	Testpoint	Solder Pad	3.3V supply rail
LSP2	5.0V DC	Testpoint	Solder Pad	5.0v supply rail
LSP3	GND	Testpoint	Solder Pad	Ground bus
TP1	RX Buffer Feedback	Testpoint	Solder Pad	RX Buffer feedback loop test point
TP2	RX Buffer Output	Testpoint	Solder Pad	Output of RX audio buffer used for calibration/ testing
TP3	RX to MODEM	Testpoint	Solder Pad	RX audio as presented to the MODEM ADC
TP5	TX Buffer Output	Testpoint	Solder Pad	DAC output after buffer amplifier
TP6	TX to Repeater	Testpoint	Solder Pad	TX audio output to the exciter
TP7	RSSI	Testpoint	Solder Pad	RSSI from receiver, as supplied to the MODEM ADC

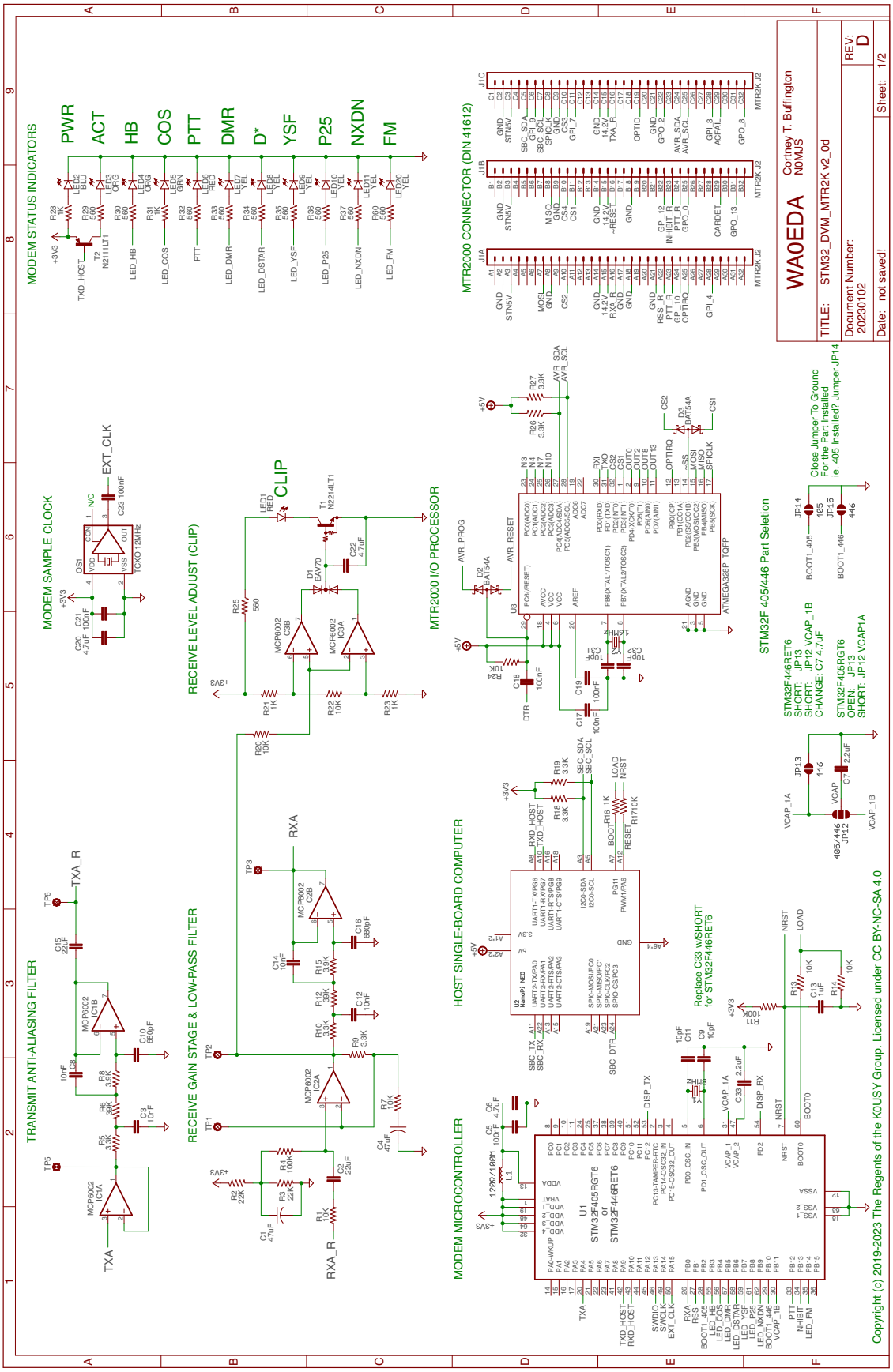
* JP11 AVR RESET: When closed, connects the hardware reset pin of the ATmega328P microcontroller to the Station Control Module external reset line. This is an output from the SCM that resets connected peripherals when the SCM itself resets. In order to ensure I/O synchronization, this jumper should not be removed. This line does not reset the ST32F446RET6 (MODEM) or the NanoPi NEO (Host).

** JP7 IO SERIAL EN.: Connects asynchronous port /dev/ttyS2 on the NanoPi NEO to the asynchronous port on the ATmega328P (through a 3.3v to 5.0v level converter). Jumpers must be removed to use NanoPi NEO /dev/ttyS2 with an MMDVM display (such as Nextion). Similarly, an MMDVM display must not be connected when using this port to update firmware on the ATmega328P, or for using the NanoPi NEO to read and/or set GPIO lines on the MTR2000 (future feature).

*** ANALOG HANDSHAKE ENABLE: When closed, connects PTT_R to GPI_3 and INHIBIT_R to GPO_2. Used with wildcard GPIO features for in-cabinet repeat knockdown and modem inhibit for analog+digital operation. GPI_3 and GPO_2 will not be available for general purpose use when connected. GPO_2 and INHIBIT_R may also be used to provide a COS signal from the station to the MMDVM modem for analog FM mode in MMDVM directly

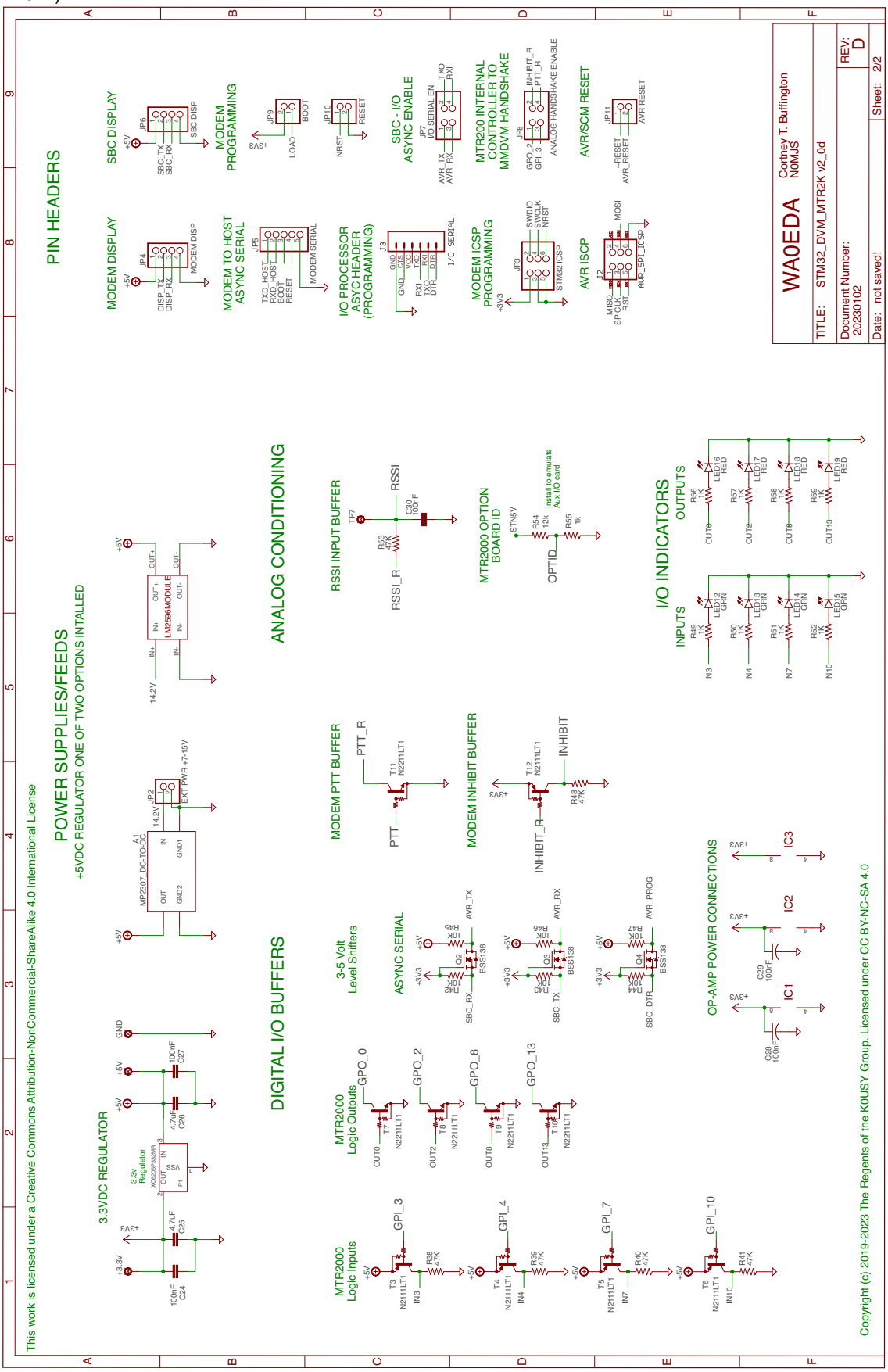
V2.0d Schematic Diagram

(page 1 of 2)



V2.0d Schematic Diagram

(page 2 of 2)



V2.0d PCB Layout Diagram