







Modifications to the Icom IC-M700 HF Marine SSB radio. Mods include extra channels, extended frequency coverage, split on/off control, and ...



[Update README.md](#)
[Elliott Liggett](#) authored 6 months ago

Name	Last commit	Last update
 microphone	added microphone schematic	7 months ago
 ramboard	Removed inverter from SRAM board	7 months ago
 LICENSE	Add LICENSE	7 months ago
 README.md	Update README.md	6 months ago

 [README.md](#)

IC-M700 Mods

The purpose of this repository is to document modifications to the IC-M700 HF Marine SSB radio. These modifications make the radio more user-friendly for amateur radio use.

Mods:

1. MARS/CAP jumper
2. More channels via SRAM expansion
3. Toggle SPLIT behavior for faster programming
4. Microphone gain and speech processor

1: MARS/CAP Jumper (and other jumpers)

To enable transmit on all covered frequencies, make sure that W37 ("W1037") on the Logic Board is installed. If it is cut, solder it back together.

To allow banks A and B to be programmed (sometimes locked out), make sure W33 ("W1033") is NOT installed.

If the 10-key buttons are not working, there is a small switch on the Matrix board, S1228, which can be toggled.

2: SRAM expansion

Schematic:

<https://gitlab.com/eliggett/ic-m700-mods/-/blob/master/ramboard/output/ramboard.pdf>

The ram board stores only the 48 channel, there are not any configuration data on the SRAM, unlike many Icom radios from this era (such as the R71). Therefore, it's totally fine to replace the battery or even replace the entire board, no critical data loss occurs except the 48 channels which you have to program in again.

The CPU constantly reads from the SRAM during power-on. Thus it is possible to modify the SRAM contents and the CPU will try and follow whatever is programmed in. The SRAM address comes from the Transmit state ("AD3" on the RAM Board pinout, which is A2 on the SRAM chip, high for TX), the knobs (bank and channel), and three bits from the CPU, which are cycled in and out to cover all the necessary data. It takes several read cycles to read out a channel and/or program a channel. The Chip Select feature is used so that the CPU can re-use the 3 address and 4 i/o connections to read out the 10-key buttons. Thus, you need to be careful to respect the position of CS (which is called "AD10" in the schematic of the RAM Board), or else you will have difficulty reading out the contents and using the 10-key. The "on" period of the address switching to the UPD444 is about 150 microseconds, and thus almost any SRAM or even EEPROM will easily work as a substitute. Some ram boards from this era, such as is used on the IC-M700TY use the UPD446. The 446 chip has an additional grounded bit, which can be switched high or low to double the total channel count. Chips can be stacked vertically, that is, connected in parallel, to add additional "super banks" of channels, so long as you are careful about terminations and pull-up/down options. The CS (chip select) line has to be pulled low on the selected active SRAM chip.

The modification discussed here adds 5 SRAM chips for 10x the channels. Each SRAM chip is used, first with A10 down and then with A10 up, so that each chip contributes two superbanks. The OEM UPD444 chip was left in place and isn't used, but it could be used if I had more potisions on my encoder.

Circuit description:

"Encoder board": A 11 position encoder is connected to a BCD-decimal decoder (CD4028) IC. The decimal output legs of this IC are combined into consecutive pairs of two using OR gates, such that the first two positions are grouped to "Out 1" and the second two positions are "Out 2" etc. These signals (5 total) are used to activate each of the 5 SRAM chips. The first bit of the encoder, which can be thought of as an "odd/even" bit, toggles with each click of the encoder knob. This signal is used to toggle A10 (the unused bit) of the SRAM. Thus the positions of the encoder knob are:

POS	SRAM	A10
0	1	0
1	1	1
2	2	0
3	2	1
4	3	0
5	3	0
6	4	1
7	4	0
8	5	1
9	5	0
10	NA	NA

With appropriate additional circuitry, position 10 could have been used to activate the OEM SRAM. But 480 channels seemed like enough and it required an additional chip, so I let it go.

The signals from this encoder board are all low-frequency as they come directly from a person rotating a knob. There is not need to be too worried about wire length or dressing on this board.

"SRAM board": On this board, the "out" signals from the encoder board go into NAND gates along with an inverted CS signal. The CS signal is inverted by using one of the extra NAND gates, driving both inputs of the gate to make it into an inverter. The result of a high signal and the high inverted "not CS" signal gives a zero only when both are high, perfect for our SRAM chips which need a low signal to enable their functions.

The new "not CS" signal we create goes from each NAND output to the CS pins of each SRAM, along with a pull-up resistor (47k-55k) to the battery-backup'd voltage used to power the OEM SRAM chip and the bank of SRAM chips. The "bit 0" line from the encoder knob is buffered with an extra OR gate (signal is sent to both inputs), and then goes the A10 of the SRAM bank (all are in parallel).

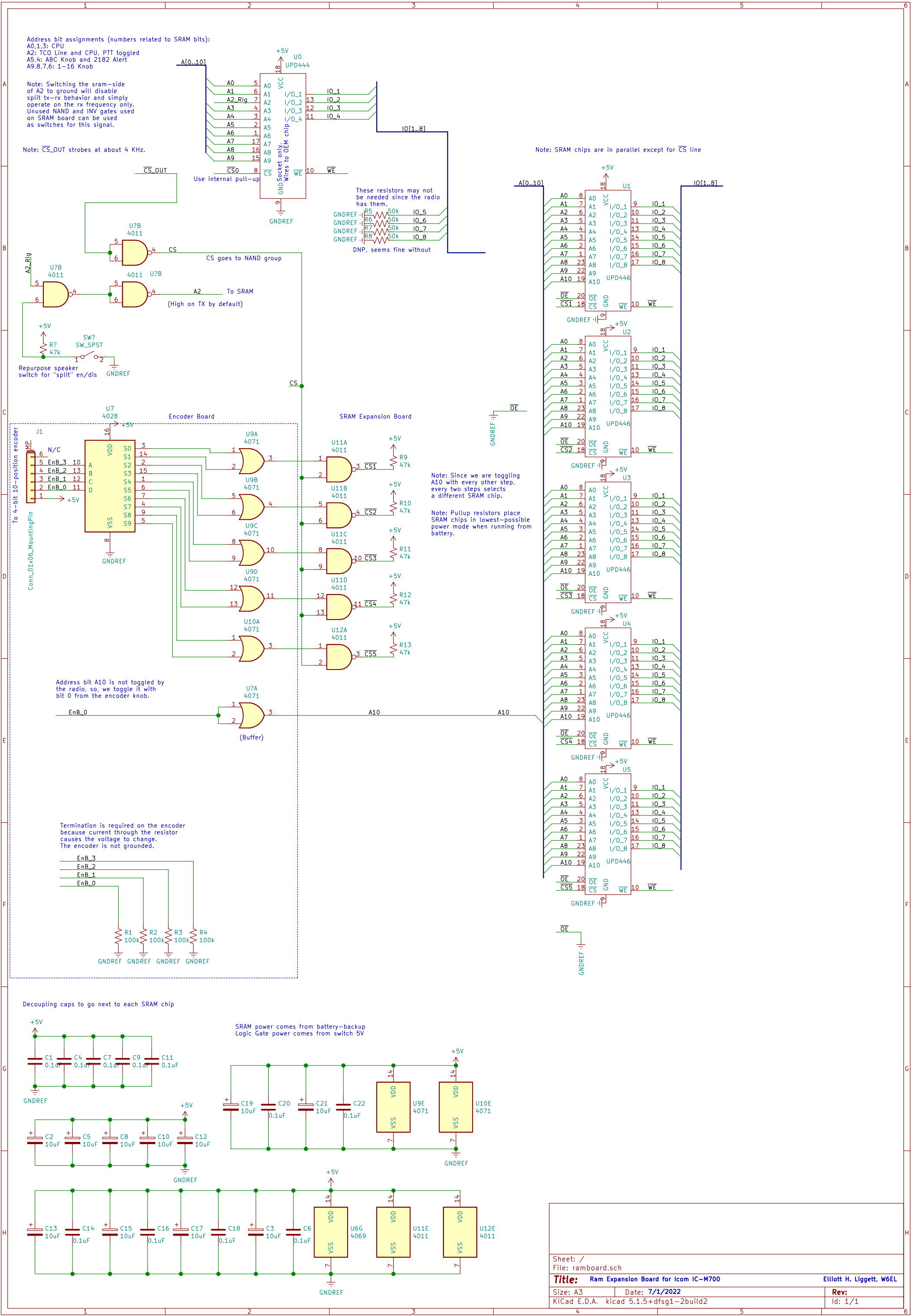
Look at the IC-M700TY service manual to see how to connect the OE pin (grounded I believe) if using the UPD446.

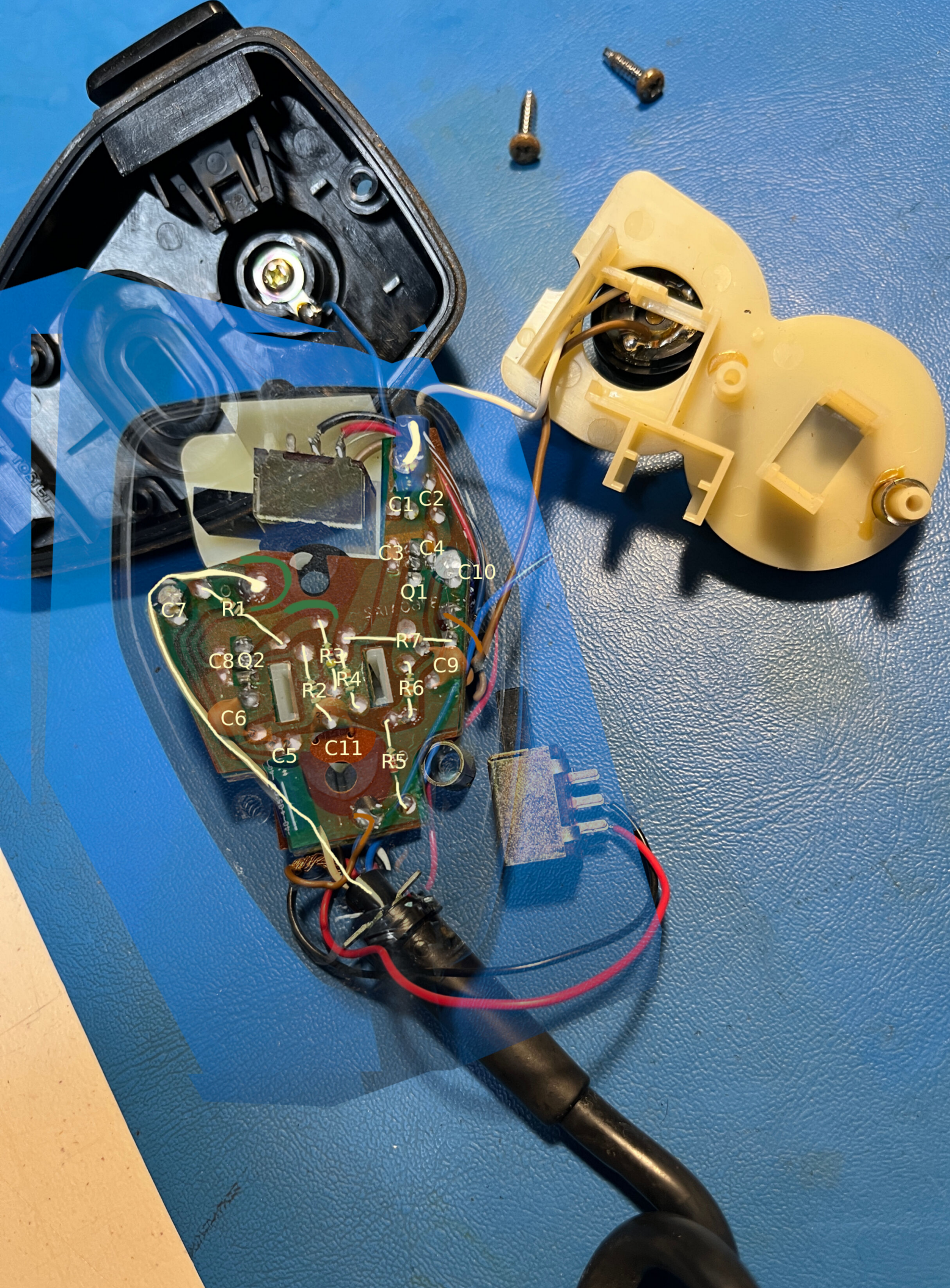
3: Split behavior

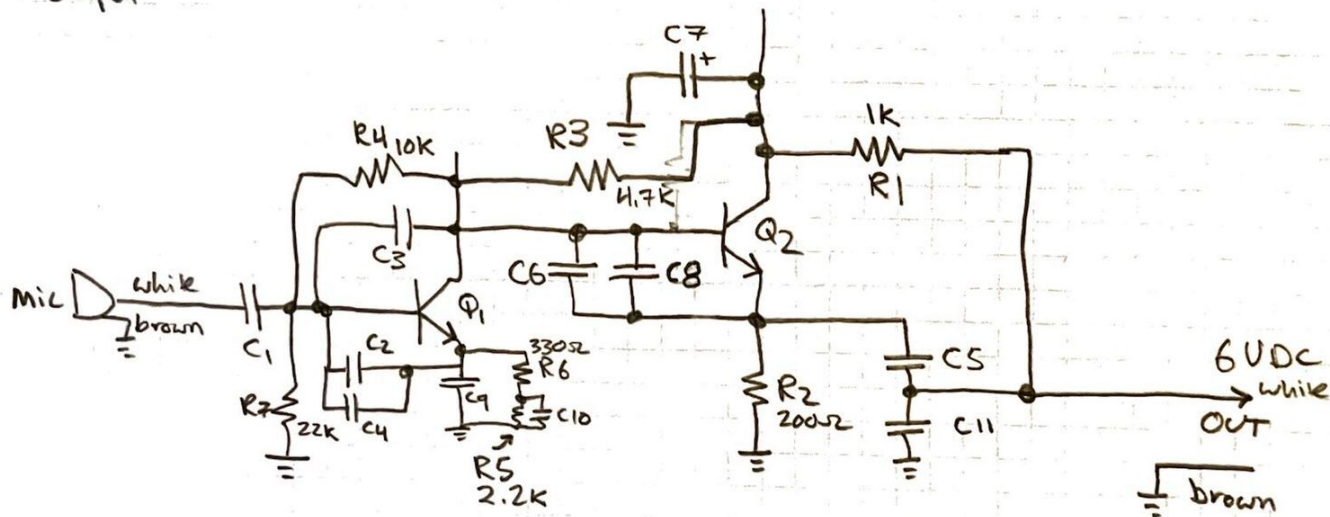
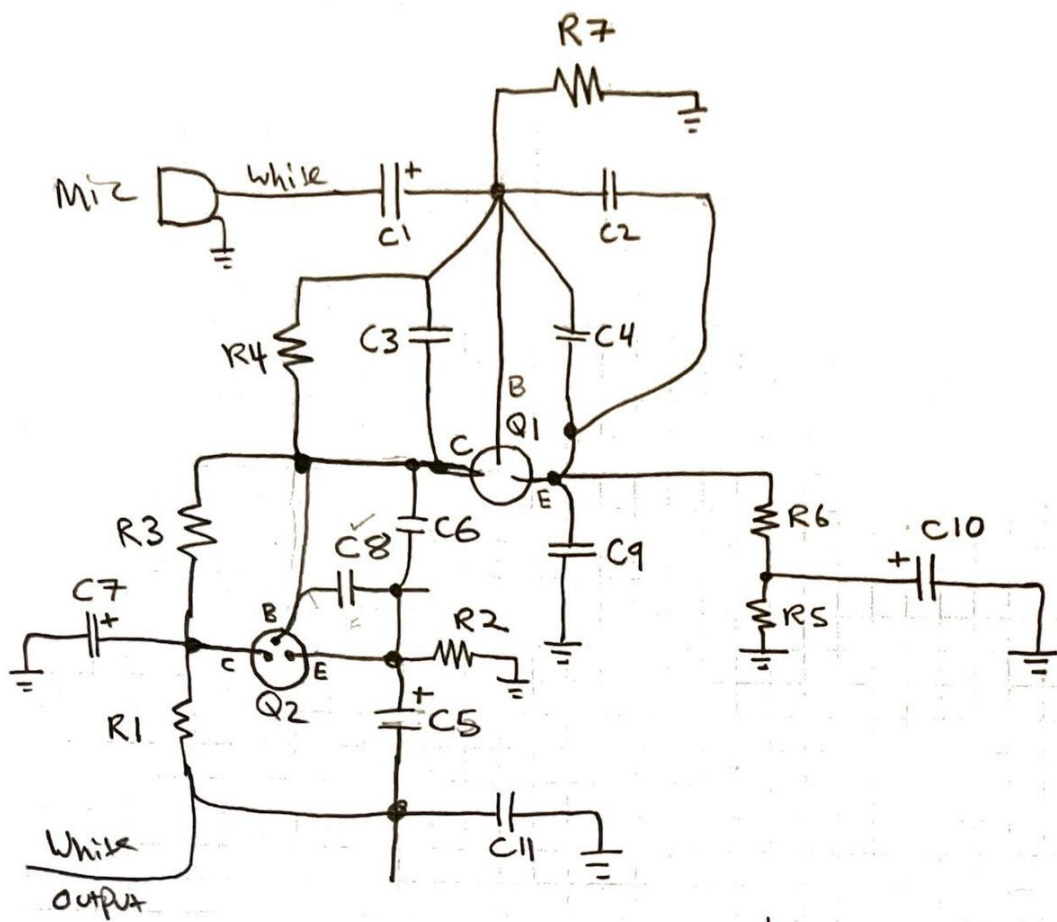
If you are tired of programming both transmit and receive every time, simply cut AD3 as it goes from the Logic Board to the RAM Board. Ground the RAM board side and now the RX frequency will always be used as the TX and RX frequency. A toggle switch can be used to alter this behavior.

4: Microphone

The microphone directory shows the schematic of the OEM microphone, which I could not find anywhere. I believe the gain is about 10x. If you substitute the microphone, as I had to do since mine was broken, you will need to increase the mic gain or possibly add an extra stage of amplification. Since SSB benefits greatly from speech compression, I added a mic pre and speech compressor. I got a fully-populated board from ebay with a MAX4466 mic-pre chip and SSM2167 compressor/gate chip.







$Q_1, Q_2 = 2SD 662, \beta \approx 220$

Icom IC-M700 Microphone