IC-9700 Amplifier/Preamp Sequencer
Version 2.0
© John Price – WA2FZW - March, 2025

Table of Contents

Introduction

This document describes a sequencer used to switch one or two linear amplifiers, antenna mounted receive preamplifiers and switch boxes that allow receiving on an SDR when the radio status changes from transmit to receive and vice-versa.

Version 1.0 of the device was specifically designed to work with the Yaesu FT-891.

In Version 1.4 I added the capability to key one or the other of two amplifiers, but it only supported one preamp and one SDR switch box. Version 1.5 supports two of everything, but eliminates the transmit inhibit capability used in the unit for the Yaesu FT-891.

Version 1.5 is designed specifically for the Icom IC-9700 on any 2 of the 3 bands it can operate on. In my case, I'm only operating on 2 meters and 70 centimeters and thus you'll see that the panel labels on my unit reflect that.

Version 2.0 improves the design when being used for the digital modes. By adding an optional USB to TTL converter, WSJT-X can be told to use that to initiate the sequence via the *RTS* (Request to Send) signal instead of CAT (CI-V) control. The sequencer will switch the preamp followed by the amplifier and finally the radio itself; this accomplishes the same thing as the Yaesu's transmit inhibit did in the original design. You could also use *DTR* (Data Terminal Ready) to accomplish the same thing.

When operating SSB or CW, you can use an optional footswitch to trigger the sequencer which works like the *RTS* signal in that the radio will be keyed last.

If neither the USB to TTL converter or footswitch is installed, the sequencer will work like previous versions; triggered by keying the radio's microphone, transmit button or CAT control. Do not try to operate break-in CW or use VOX control under any circumstances when using the sequencer this way.

The Hardware

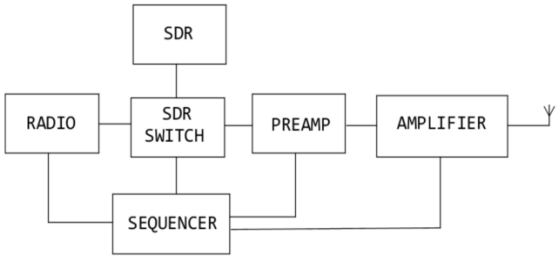
Some Assumptions

- If installed, the *RTS* line on the USB to TTL converter is normally at 5V and goes to ground when *RTS* is activated.
- The radio has a *PTT* line (*SEND* in IC-9700 nomenclature) that goes to ground when the radio is put into transmit mode either manually or via CAT control.
- The sequencer can key the radio by grounding its *PTT* (*SEND*) line (indirectly via the *TX_KEY* GPIO pin and K4) when the sequencer is activated via the *RTS* line.
- The preamplifiers are put in receive mode by powering them from a +12V source. Removing the +12V power switches them to transmit (bypass) mode (fail-safe operation).
- The SDR switch boxes are switched from receive mode to transmit mode by grounding a *PTT* input on the switcher. SW4 and SW5 on the schematic are optional switches to ground the *PTT* line of the SDR switchers to put one or both into permanent transmit (bypass) mode.
- The linear amplifiers are keyed by grounding their keying inputs. Relay contacts are used for this so the amplifier's keying voltage can be pretty much anything. A SPDT switch (SW6) selects which of the two amplifiers is actually keyed. I used a center-off switch which allows neither amplifier to be keyed.
- This sequencer is designed primarily to be used when running the various digital modes.
- J8 connects to an optional RCA jack which is for a footswitch. The footswitch acts like the *RTS* signal and provides a fail-safe method of operating for SSB and CW.

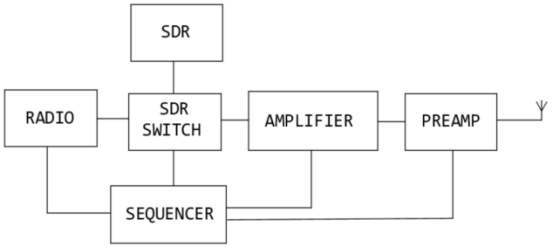
The use of the USB to TTL converter is optional. If not used, the digital mode programs can still trigger the sequence via CAT commands as in previous versions of the sequencer. If you don't use the USB to TTL converter, K5 does not need to be installed.

Equipment Configurations

There are two possible configurations one might use when using an external preamplifier, linear amplifier and/or SDR switch:



Configuration 1



Configuration 2

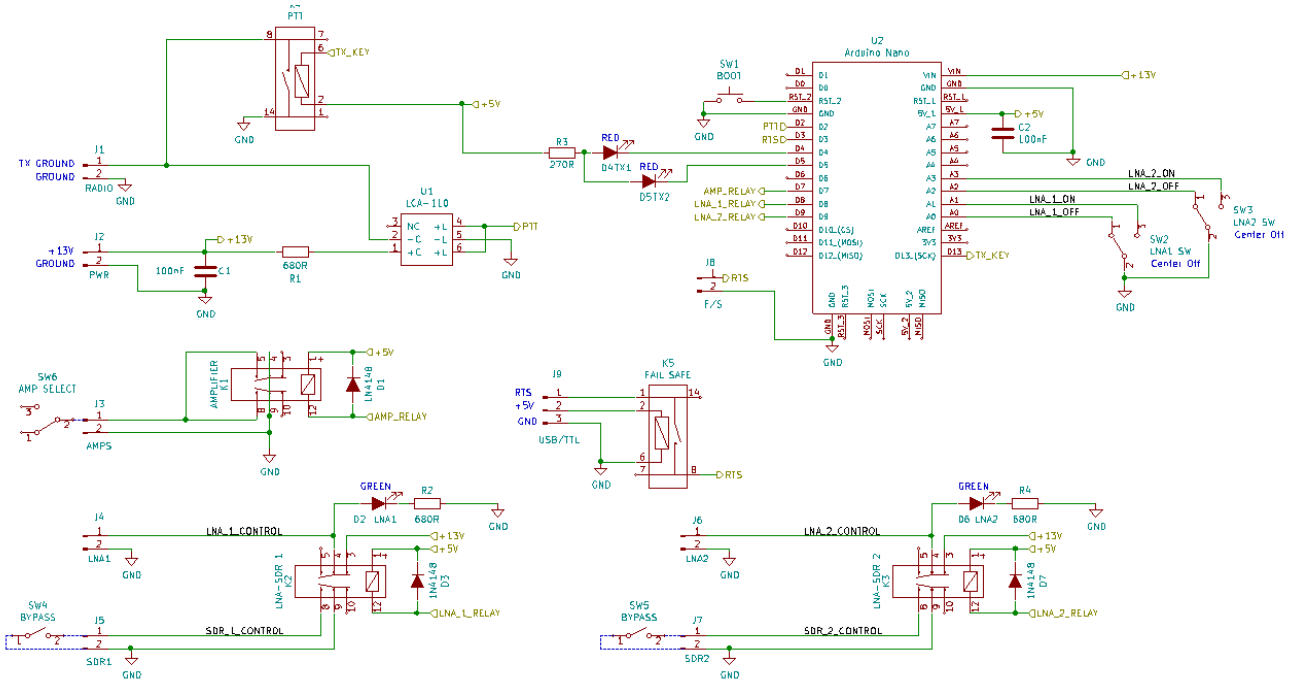
The primary difference is whether the preamp is ahead of or behind the amplifier. If you have preamplifiers that can handle the output power of the amplifier, it's better to put them between the antenna and the amplifier and as close to the antennas as possible.

Any of the major components (except the radio and antenna) are optional. If all you have is a radio and an antenna, you can stop reading because you have no need for this sequencer!

Which configuration you have and what specific equipment will determine how to set the various delay values in the software.

Theory of Operation

Here's the schematic:



Switches SW4, SW5 and SW6 are optional.

Note that although the schematic shows the +13V power source as being from the radio, I power mine from the same 12V power supply that powers the radio. The IC-9700 is capable of 1 amp from the ACC connector and I would be pushing that limit with the relays in the 2 LNAs and the SDR switchers in my configuration. I ran the original version of the sequencer from the radio's power when there was only one preamp being switched with no problems.

When the various pieces of equipment are switched between receive and transmit mode is controlled by the software in the Arduino Nano processor. We'll get to setting the timing later.

In Version 2.0, the receive to transmit sequence can be initiated in three ways. When using WSJT-X (or any of the other digital mode programs), the sequence is initiated via a *RTS* signal provided from an optional USB to TTL converter. The link is for the particular one I used; I suspect there others which will work just as well.

When the *RTS* input goes *LOW*, the processor immediately switches the preamps and SDRs to the transmit mode via K2 and K3. After *LNA_ON_DLY* milliseconds, the selected amplifier is keyed via K1. Then after *PTT_ON_DELAY* milliseconds, the *TX_KEY* GPIO pin is set to *TX_ON* which will put the radio itself into transmit mode by activating the radio's *PTT* line via K4 on the schematic.

The sequence can also be initiated when the radio's *PTT* line goes active; this will be the case when the radio is sent a transmit command via CAT control or the operator keys the microphone. This is indicated by the radio's *PTT* input going to ground. That causes a *LOW* indication on the *PTT* output of U1 on the schematic.

When the transmit sequence is started via the radio's *PTT* line, the process is the same except we do not re-key the radio via the *TX_KEY* GPIO pin. Be very careful when keying the transmitter manually as it is very easy to destroy the preamp and/or SDR. It is much safer to use the optional footswitch for SSB or CW.

The transmit to receive sequence is also a bit different in Version 2.0. There is the (remote) possibility that while the radio was initially put into the transmit mode via the *RTS* signal, the operator also manually keyed the radio. The process will be explained in the description of the *SetReceive* function below.

Preamp Always On or Off Switch Considerations

Two switches (SW4 and SW5) were added in a previous version of the hardware when I started using the IC-9700 for satellite work. When working the satellites, the radio is transmitting on the 70cm band and receiving on 2 meters (or vice-versa). Without the switches, when transmitting on 70cm and listening on 2 meters, the 2 meter preamp would switch off when I transmitted causing me to sometimes no longer be able to hear myself.

Putting the switch for the receive band in the 'ON' position cured the problem.

VERY IMPORTANT: Be very careful when switching back to normal operations to put the switch back to the normal operating position or to the 'OFF' position. I already fried one rather expensive preamp!

When one (or both) of the switches is in the ‘ON’ position the red transmit indicator LED for that band will flash rapidly as an alert when the radio’s *PTT* is not active. When the *PTT* goes active, however, both red LEDs will be on continuously.

Also when the switch is in the ‘ON’ position, the SDR switch will be stuck in receive mode. You can put them into permanent transmit mode by powering them off or putting the (optional) SDR Disable switches (SW1 and SW2) into the ‘DISABLE’ position. Of course you won’t be able to receive on the SDR in this mode.

My SDR switcher also has its own bypass switch.

K5 is a fail-safe relay. I found in testing the initial prototype that when the USB connection to the USB to TTL converter was disconnected, the *RTS* signal went to a *LOW* (active) state which caused everything to switch to transmit mode. The same thing would happen when rebooting the PC.

If the 5V output from the USB to TTL converter goes away, K5 disconnects the *RTS* line from the processor. The Arduino’s internal pullup resistor on GPIO pin D3 keeps the *RTS* signal showing as inactive. The footswitch (also connected to D3) will continue to be functional.

The Printed Circuit Board

The hardware consists of a printed circuit board (PCB) containing an Arduino Nano (or clone) processor, five mechanical relays, one solid state relay (SSR), a few resistors, a couple of capacitors and several LEDs and switches (mounted externally).

The board can be powered from the radio if it can conveniently provide 12 to 13 volts and enough amperage however one must be careful not to exceed any current limitations the radio might have (1 amp max for the IC-9700). I powered earlier versions of the sequencer (with a single preamp and no SDR switcher) from the radio with no problems but now use a separate power supply (A wall-wart is fine if it doesn’t make RFI). I use the same power supply that powers the radio.

U1 is an LCA-110 solid state relay (SSR) which provides isolation between the radio’s *PTT* line and the processor. It is required as the normal voltage on the *PTT* line is more than the 5V that the Arduino pins can handle. I measured a bit over 6V on my IC-9700; it may be higher for other radios.

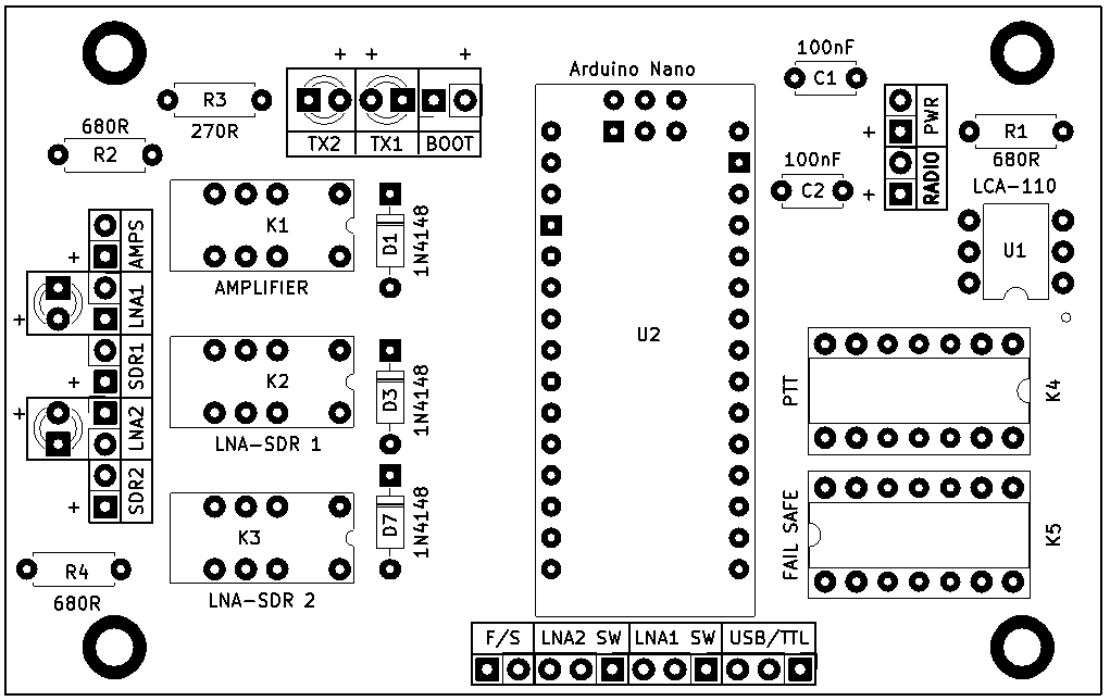
Relay K1 keys the amplifier(s). Relays K2 and K3 provide power to the preamps and ground the *PTT* inputs on the SDR switches in transmit mode. K4 grounds the radio's *PTT* line when the sequence is initiated via the *RTS* signal

The LEDs indicate when the radio's *PTT* line is active (red), and when the preamp is in receive mode (green). The preamp LED is operated by the relay which supplies power to the preamp in receive mode providing a positive indication that power is actually being sent to the preamp (or at least to the connector on the rear panel).

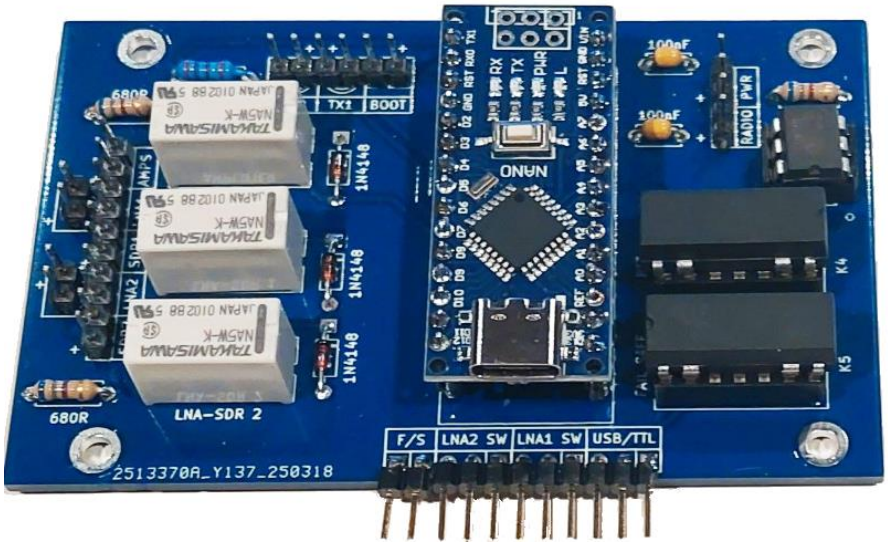
SW4 and SW5 are connected to the PCB. The optional switches to disable the SDRs (SW1 and SW2) are not connected to the PCB, but are simply SPST switches that ground the *PTT* lines to the SDR switchers. I simply connected them across the RCA jacks for the SDR switchers.

The SPDT switch (SW6) to select which amplifier should be keyed is connected between J3 (AMPS) and the RCA jacks for the amplifier keying lines.

Here's what the printed circuit board layout looks like:

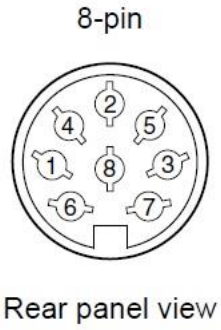


And here’s the assembled circuit board:



Icom IC-9700 Connections

The Icom uses a large size 8 pin DIN connector (*ACC*) for the external connections:



I have a couple versions of the *IC-9700 Basic Operating Manual*, however they all have the same description of the pins on the *ACC* DIN connector. On the original version for the IC-9700, I cut one end off of a DIN cable and hardwired the cable to the PCB; this time I used a DIN jack on the rear panel of the sequencer. The DIN Cable I used to make the connection came from Amazon.

The following table gives the pin usage I used to connect the radio to the sequencer (I used an external power source instead of the 13.8V from the radio as it is only capable of supplying 1 amp):

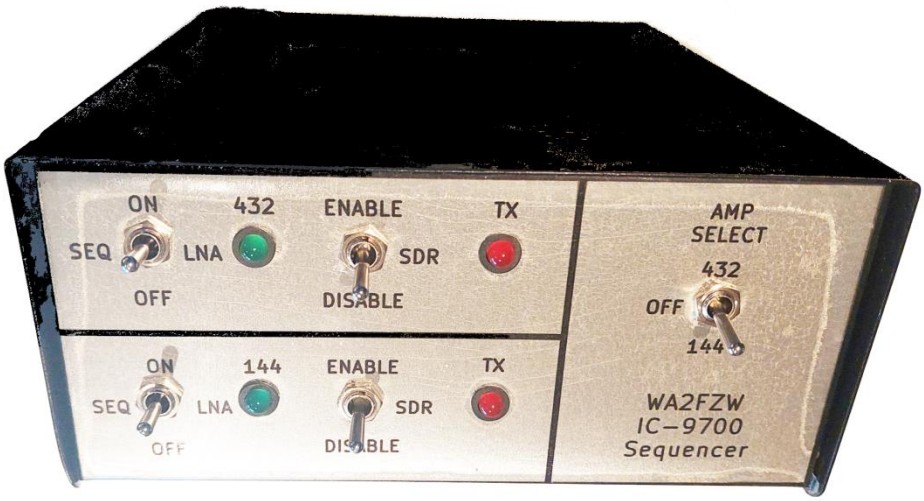
Pin	IC-9700 Nomenclature
2	Ground
3	Send (aka *PTT*)
7	+13.8V @ 1 Amp

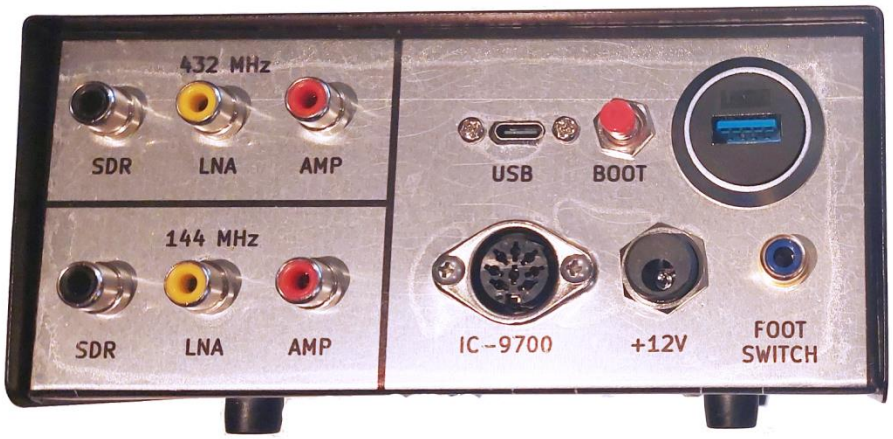
The Icom manual says the *SEND* line is both an input and an output (I verified this). When the radio is keyed via the microphone's *PTT* button, the *TRANSMIT* button on the front panel or via CI-V control, the *SEND* line goes *LOW*.

Putting a *LOW* on the *SEND* line will also cause the radio to go into transmit mode.

Some Construction Notes

I built the unit into an enclosure from Circuit Specialists. Here's what the final assembly looks like (sorry, not the greatest pictures):





The Software

GPIO Pin and Other Symbols Defined

The following are the GPIO pin assignments; for the most part, the symbols used match the net names on the schematic:

PTT	2	*PTT* Indication (interrupt capable)
RTS	3	*RTS* Indication (interrupt capable)
TX1_LED	4	Band #1 transmit indicator LED
TX2_LED	5	Band #2 transmit indicator LED
AMP_RELAY	7	Operates the linear amplifier relay (K2)
LNA1_RELAY	8	Operates the band #1 preamp/SDR relay (K3)
LNA1_RELAY	9	Operates the band #2 preamp/SDR relay (K4)
TX_KEY	13	Operates K1 to ground the PTT line
LNA1_OFF	A0	Preamp for band 1 always off when LOW
LNA1_ON	A1	Preamp for band 1 always on when LOW
LNA2_OFF	A2	Preamp for band 2 always off when LOW
LNA2_ON	A3	Preamp for band 2 always on when LOW

The *DEBUG* symbol can be defined as 0, 1 or 2. When set to 0, no debugging stuff is enabled. Setting *DEBUG* to 1 causes various status messages to be sent to the Arduino IDE's serial monitor. Setting the value to 2 also exaggerates the sequence timing values to facilitate monitoring with meters and/or an oscilloscope for testing.

There are several definitions of states for the various GPIO pins to make the pin states more readable, for example:

```
AMP_ON   LOW    // A LOW on the AMP_RELAY pin keys the linear
AMP_OFF  HIGH
```

There are several global variables:

txState	Current transmit/receive state
oldTxState	Previous transmit/receive state
rtsState	Current state of the *RTS* line
pttState	Current state of the *PTT* line
lna_1_ForcedOff	*true* when switch SW2 is in the always off position
lna_1_ForcedOn	*true* when switch SW2 is in the always on position
lna_2_ForcedOff	*true* when switch SW3 is in the always off position

lna_2_ForcedOn	*true* when switch SW3 is in the always on position
tx_1_LedState	Used in *CheckBlink* to toggle the LED state
tx_2_LedState	Used in *CheckBlink* to toggle the LED state

Timing Settings

There are four symbols in the software that define the timing for the sequencer:

LNA_ON_DELAY	25mS	Time between switching the LNA(s) to transmit mode and keying the amplifier
PTT_ON_DELAY	50ms	Time between keying the amplifier and keying the radio when the sequence is initiated via the *RTS* signal.
PTT_OFF_DELAY	10mS	Used in the *SetReceive* function when it is determined that the PTT line is active because of K4 being energized.
AMP_OFF_DLY	50mS	Delay between turning the amplifier(s) off and returning the preamp and SDR switch to receive mode.

The values shown are those currently set in the software when the *DEBUG* level is 0 or 1. The values may be adjusted to suit your needs based on the specifications of your particular equipment and the equipment configuration. There are alternate settings that are used when the *DEBUG* level is set to 2; I currently have all except *PTT_OFF_DELAY* set to 3 seconds.

Note, the sequencer is designed primarily for digital mode operations. It can be used for SSB operations using manual *PTT* (not *VOX*) and pause slightly between keying the radio and speaking. Using it for CW operation should only be done if the radio is put into transmit manually before operating the CW key; in other words don't try using break-in modes (I fried a $250 LNA that way)! You can also use a footswitch to trigger the transmit sequence. Using the footswitch is the same as triggering the transmit sequence via the *RTS* line which will cause the preamps and linear to be switched before the radio.

Overall Functionality

The software is pretty trivial. The *setup* function sets up all the GPIO pins used and sets them to the proper initial states.

Experimentation has shown that the preamp and amplifier relays will not operate until their GPIO pins are set as *OUTPUT* pins, thus the preamp will startup in transmit (bypass) mode and the amplifier will be off. Note, however that during the reboot process, the radio gets keyed for a few milliseconds, so when rebooting, make sure the preamps and linears are disabled.

The first thing done in setup is to make sure the radio is not in transmit mode unless the operator has it keyed manually; the sequencer cannot do anything about that.

The *loop* function really only does a few things; it first calls the *CheckBlink* function which blinks one or both of the transmit LEDs when one or both of the LNAs are in the always on mode.

Next, the *SwitchPreamp* function is called every 50 milliseconds which may or may not switch the preamps and SDR switches into receive mode. I'll explain that function in more detail momentarily. It is called everytime through the loop so it can detect whether or not one of the LNA switch settings changed.

Next, the *loop* function looks at the states of the *PTT* and *RTS* lines to see if the transmit/receive status needs to be changed and if so, invokes the *SetTransmit* or *SetReceive* to take the appropriate action. Note that the optional footswitch works like the *RTS* signal.

The *SetTransmit* Function

This function takes care of switching things from receive mode to transmit mode.

First the preamps and SDR switchers are switched to transmit mode (unless one or both of the preamps are in the always on mode).

After delaying for *LNA_ON_DLY* milliseconds, the amplifier is keyed; if SW6 is installed, the operator can select which of two amplifiers is keyed.

There are two ways in which the transmit sequence could have been initiated. If it was started because the *PTT* went active, there is nothing else to do except set the *txState* and *oldTxState* variables to *TX_ON*.

However if the sequence was started because the *RTS* went active, we need to activate the *PTT* line to actually put the radio into transmit mode. That is done *PTT_DELAY* milliseconds after the amplifier is keyed.

The *SetReceive* Function

This function handles the transmit to receive sequencing. It is a bit trickier than in previous versions due to the two different methods of putting everything into transmit mode.

First of all, we check the *RTS* status; if it is active, we leave everything in transmit mode and simply return from the function.

If *RTS* is not active, we check to see if the *PTT* is active. If it is active, it could be because the radio was keyed manually or because the software activated it via K4. We assume it's active via K4 so we attempt to turn the *PTT* off and wait up to *PTT_OFF_DELAY* milliseconds. After waiting, we check to see whether or not the *PTT* is still active. If it is, the assumption is that the radio was keyed manually or via a CAT command and not via the *RTS* line. If this is the case, we leave everything in transmit mode and return.

If the *PTT* is now inactive, we start the transmit to receive sequence. The transmit LEDs are turned off and the amplifier is unkeyed. After *LNA_ON_DLY* milliseconds, the preamps and SDR switchers are switched back to receive mode unless one or both switches are in the 'OFF' position.

The *CheckBLink* Function

This function checks the status of the preamp mode switches and if one or both is on, it flashes the transmit LED for the associated band every 100 milliseconds to alert the operator to be very very careful.

There is a 2 millisecond delay at the end of the function. That ensures that the next time the function is called the system time won't be on the same 100 millisecond increment.

The *PTT_ISR* and *RTS_ISR* Functions

These are invoked via software interrupts whenever the state of the *PTT* or *RTS* lines changes. Both simply set the state of the input by reading the GPIO pin, however if the *RTS* is already active, *PTT* interrupts are ignored. This is because the software turns the *PTT* on when the sequence is started by *RTS* going active, which also generates an interrupt.

The *SwitchPreamp* Function

This function perhaps deserves a bit more explanation than the other functions. It takes two arguments; the first being which band(s) should be switched (*BAND_1*, *BAND_2* or *BOTH_BANDS*). The second argument is the new preamp and SDR switch state requested (*LNA_RX* or *LNA_TX*).

When the argument is *LNA_TX*, the selected preamp(s) and SDR switch(es) will always be put into transmit (bypass) mode unless the associated LNA switch (SW2 and/or SW3) is in the ‘ON’ position. If the switch is in the ‘ON’ position, both the preamp and SDR switch will remain in receive mode (***BE VERY CAREFUL***).

However, if the argument is *LNA_RX*, the preamp(s) and SDR switch(es) will only be switched into receive (active) mode if the associated preamp enable switch is in the ‘SEQ’ position and the *PTT* is not currently active (*TX_OFF*).

The *Test_LNA* Function

You will see a call to this function commented out at the end of the *setup* function. When the function call is un-commented, it will be invoked. It allows testing of the preamp switch functions. It is a never-ending loop, so when it runs, none of the rest of the sequencer functions will be enabled.

When one of the ‘LNA’ switches is put in the ‘ON’ position, the associated LNA LED will flash. When the switch is in the ‘OFF’ position, the associated LED should go out. The LED should also remain on when the switch is in the ‘SEQ’ position.

You should also see the voltage on the associated ‘LNA’ connectors (if the unit is connected to a 12V power source) following the LED and you should also see contact closure/opening following on the ‘SDR’ connectors provided the associated ‘SDR’ switch is in the ‘ENABLED’ position.

If the ‘SDR’ switch is in in the ‘DISABLED’ position you should see a constant contact closure on the associated ‘SDR’ connector.

It is critical to test everything before actually hooking the sequencer up to external equipment as you can easily destroy a preamp or SDR. The complete testing procedure is documented below.

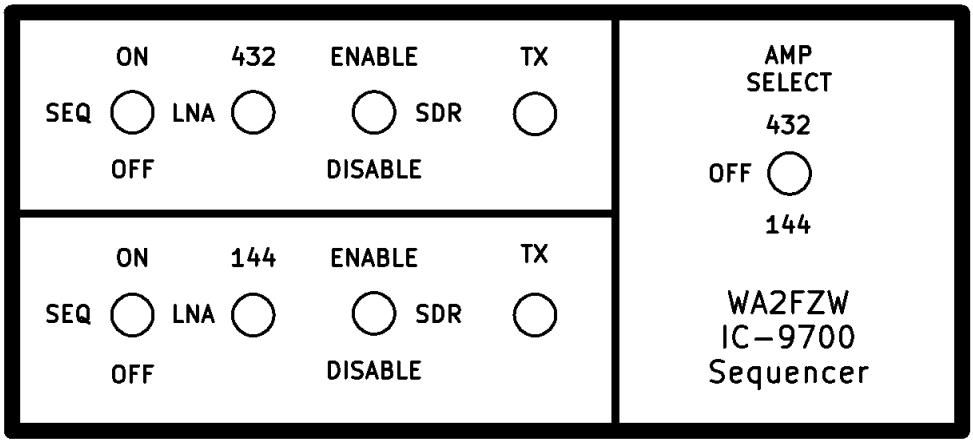
The *ShowPinStatus* Function

When the *DEBUG* level is set to ‘1’ or more, this function reports the status of the LNA switches, the *PTT* line and the *RTS* line at startup.

Some Operational Notes

Front Panel Controls and Indicators

Here’s what the front panel looks like:



In the two areas on the left, from left to right are:

- LNA/SDR ‘ON’, ‘SEQ’, ‘OFF’ switches (SW2 & SW3 on the schematic).
- Green preamp power on (receive mode) LEDs.

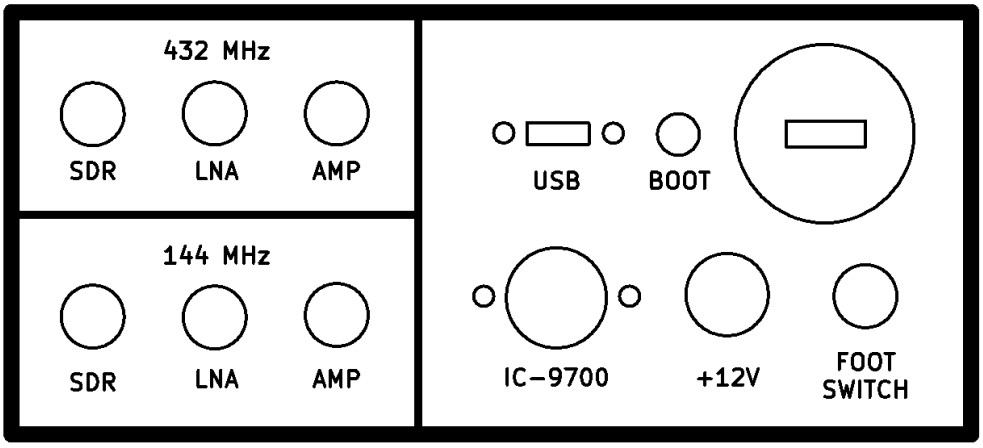
- SDR ‘ENABLE’/’DISABLE’ switches (SW4 and SW5).
- Red transmit indicator LEDs.

In my build, the top section (labeled 432) are the BAND 2 items and the bottom section (labeled 144) are the BAND 1 items.

The section on the right side contains the amplifier select switch (SW6). I used the center-off option so that both amplifiers can be disabled.

Rear Panel Connections

Here’s what my rear panel looks like:



In the sections to the left are the connectors for the SDR switchers, preamps and amplifiers. I used RCA jacks. As on the front panel, the top part (labeled 432 MHz) are the connections for BAND 2 and the bottom section (labeled 144 MHz) is for the BAND 1 connections.

The section to the right contains:

- A type ‘C’ USB connector which allows one to re-program the unit without having to take it apart.
- The ‘BOOT’ momentary pushbutton (SW1 on the schematic).
- A type ‘A’ USB connector which is the USB connection for the USB to TTL converter used to generate the *RTS* signal.
- The 8-pin DIN jack for the connection to the IC-9700.

- The 12V power connector.
- An RCA jack for the footswitch.

The type ‘A’ USB connector is actually a panel mounted type ‘A’ to ‘A’ USB connector. The USB to TTL converter plugs into the rear of the unit. It came from Amazon.

SDR ENABLE/DISABLE Switches

If you are using an SDR switcher and have installed the switch to disable the SDR switcher as described above it is best to disable the SDR when not actually using it. If you’re using my switcher or a different one with a splitter, there is a 3dB loss due to the splitter.

There are several such switchers available. The one I am using is of my own design. It has an internal splitter and grounds the input to the SDR in transmit mode.

Another good one is from Paul (N2EME). Unlike mine, Paul’s requires an external splitter to be able to receive simultaneously on the radio and the SDR.

There is a switcher available somewhere on Amazon and MFJ also has one (although they’re out of business, their stuff seems to still be available). Paul and I have tested these and they are not very good; I don’t recommend using wither one.

LED Indications

Green LEDs – On indicates power is being supplied to the preamp for that band (or at least to the connector on the unit), which means it is in receive mode. If the LED is off, the preamp is in transmit (bypass) mode.

Red LEDs – Both on indicates the radio's *PTT* is active either because the radio was keyed manually or via a CAT command or as a result of the transmit sequence being activated via the *RTS* line or a footswitch. Flashing is a warning that the preamp switch for the band is in the always 'ON' mode and transmitting on that band could damage the preamp. This feature is primarily for the IC-9700 where in satellite mode, you are transmitting on one band and receiving on a different band; each band on the IC-9700 has its own antenna feed.

Note that putting one of the preamp switches into the 'ON' position will also put the associated SDR switch into permanent receive mode and transmitting could destroy the SDR. Whenever the preamp switch is in 'ON' you should 'DISABLE' the SDR switcher.

Preamp Overload Protection

Both my preamps contain circuitry to protect them from strong nearby signals (in-band or out-of-band). If you use a preamp without such protection, it is important that the 'ENABLE'/'DISABLE' switch be used to put yours into bypass mode before using a different nearby transmitter and/or antenna. I do this even though my preamp has the protection circuit.

Testing

I strongly suggest testing everything without any amplifiers, preamps or SDRs connected to the unit. It is ok to connect the unit to the radio for testing, or you could use a jumper wire to simulate the radio's *PTT*. The unit would have to be connected to a 12V source. The only test equipment needed is a volt/ohm meter.

Preamp 'ON'/'OFF'/'SEQ' Switches (SW2 & SW3)

With power applied to the unit and these switches in the 'SEQ' position, both green LEDs should be lit and you should measure +12V on the LNA connectors (if the *PTT* is not active). The SDR connectors should show an open condition.

With the radio's *PTT* active, and these switches are in the 'SEQ' position, the green LEDs should go out there should be no voltage on the 'LNA' connectors. The 'SDR' connections should show a contact closure.

Moving the switches to the 'OFF' position should cause the green LEDs to go out and there should be no voltage on the 'LNA' connectors regardless of the *PTT* state. The 'SDR' connections should show a contact closure.

With the switches to the 'ON' position (*PTT* inactive) the green LEDs sill be illuminated and there should be +12V on the 'LNA' connectors. The red LEDs should be flashing. The 'LNA' connections should show an open condition.

The *Test LNA* function call at the end of the *setup* function in the software can be enabled to facilitate testing the 'LNA' switches.

SDR 'ENABLE'/'DISABLE' Switches

With these switches in the 'DISABLE' position, the SDR connectors should show a contact closure regardless of the state of the preamp switches or the radio's *PTT*.

When these switches are in the 'ENABLE' position, the SDR connections should show open when the *PTT* is inactive and should show a contact closure when the *PTT* is active.

If you implemented the option to have the preamp switches also disable the SDRs when in the 'ON' position (as mentioned above), the SDR connections should show a contact closure when the preamp switches are in the 'ON' position.

PTT Tests

With the preamp switches in the 'SEQ' position and the SDR switches in the 'ENABLE' position, the green LEDs should be on and the red LEDs should be off when the radio's *PTT* is inactive. There should be +12V on the 'LNA' connectors and open conditions on the 'SDR' connections.

With the switches still set as above, when the *PTT* is active, the green LEDs should go out and both red LEDs should light. There should be no voltage on the preamp connectors and the SDR connectors should indicate a contact closure.

One thing to be aware of; if one of the preamp switches is in the 'ON' position, the associated red LED should be flashing when the *PTT* is inactive. When the *PTT* goes active, however both red LEDs will be lit continuously.

With the *PTT* inactive the amplifier connections for both amplifiers should show an open condition. When the *PTT* goes active, the connection for the selected amplifier should show a contact closure. Make sure to check this for both amplifiers. As mentioned earlier, one could use a center-off SPDT switch so that neither amplifier is keyed.

DEBUG Output

Setting the DEBUG symbol to a value of 1 (or higher) causes several messages to be sent to the Arduino IDE's serial monitor.

When the program is started, you will see the following messages showing the states of the LNA switches and the *RTS* and *PTT* lines:

```
WA2FZW - Amplifier/Preamp Sequencer - Version 2.0
LNA 1 Status = Always On
LNA 2 Status = Sequenced
RTS Status = Inactive
PTT Status = Inactive
```

When the sequence is initiated via the *RTS* or *PTT* line going active you will see a sequence of messages like the following. In this case, the sequence was initiated via the *RTS* line. The numbers are the readings from the *millis()* function at the time of the event and the text at the right are comments which are not part of the output:

```
Time: 4145          Time sequence was initiated
pttState = 1        PTT is inactive
rtsState = 0        RTS is active
LNAs in TX: 4146    LNAs switched to TX mode 1mS after RTS active
Amp On 4170         Amp keyed 24 mS later
PTT On 4220         PTT activated via K4 50mS later
```

When the *RTS* line was released, the following messages were produced:

Time: 7458	Time *RTS* released
pttState = 1	*PTT* is actually active (see note below)
rtsState = 1	*RTS* is inactive
PTT Off 7460	*PTT* released 2mS after sequence start
Amp Off 7510	Amp un-keyed 50mS later
LNAs in RX: 7535	LNAs switched to RX 25mS later

If *DEBUG* is set to 2 or higher, the times are exaggerated to facilitate watching the actual outputs with a meter or scope.

When the sequence was initiated via the *RTS* line, we do not show the *PTT* as being active. Doing so caused a lockup condition where the *PTT* wouldn't be released when *RTS* went inactive.

Known Bugs and Glitches

Booting the Processor

When the processor is booted, the various relays switch themselves into the proper receive mode, but very briefly switch to transmit mode. There is nothing that can be done about it as that's just how the Arduino GPIO pins work. When initializing them

Because the radio will very briefly (milliseconds) be put into transmit mode, one should disable the preamps and amplifiers using the front panel switches when rebooting the processor.

Possible Enhancements

The one thing I thought of (after having the PCBs made) would be to use a DPDT (not center off) switch for the amplifier selection and having the other side of the switch connected to a couple of GPIO pins (similar to how the preamp switches work) to tell the processor which band is being used for transmit.

With such a modification the transmit LEDs could actually show which band is in transmit mode rather than both of them being lit during transmit on either band. The software modifications would be fairly simple.

Suggestion Box

I welcome any suggestions for further improvements. Please feel free to email me at WA2FZW@ARRL.net.

Bill of Materials

Here is a list of the parts you will need and in many cases, links to where you can get the less common parts.

There are hyperlinks to Mouser Electronics for some of the parts. Mouser has implemented a Captcha-like guard against robots. Thus, you can't simply click on the links, but if you copy the links and paste them into your browser they work (at least they did when I wrote this).

There are also hyperlinks for some items available from Jameco Electronics. Jameco has recently instituted a $25 minimum order policy, so if you order from them, find some other stuff you need!

The PCB		Gerber files are available on Github.
R1, R2 & R4	680R 1/4W	
R3	270R 1/4W	
C1, C2	100nF 50V	
U1	LCA-110 SSR	6-pin DIP package; available from Mouser and other common suppliers
U2	Arduino Nano	I used clones which are much cheaper than genuine Arduinos.
D1, D3 & D7	1N4148	
D2, D4, D5 & D8	General purpose LEDs – 20mA	Pick whatever colors you like. I used red for the TX indicators and green for the preamp indicators. The documentation assumes these colors. The PCB is designed so that the LEDs can be installed on the PCB or connected externally.
SW1	Momentary push button	

SW2 & SW3	SPDT center off toggle switches	
SW4 & SW5	SPST Toggle Switch	
SW6	SPDT toggle switch	Not on the PCB, but connected between J3 (*AMPS*) and the amplifier connectors to select one of two amplifiers. A center-off switch can be used to select neither amplifier.
K1 – K3	Fujitsu NA5W-K DPDT relay	Available from Jameco
K4 & K5	NTE R56-1D.5-6 Reed Relay	I'm told NTE is going out of business. An equivalent is the Littelfuse HE721A0500; available from Mouser or Digikey.
J1 – J9	Various size 2.45mm male pin headers	The PCB layout allows some of these to be installed as larger headers. Also, there are pin headers for SW1, SW2 & SW3. Use vertical headers except for the ones for SW2, SW3, J8 and J9 which need to be horizontal ones.
IC-9700 Connection	8-pin large female DIN receptacle	Available from Jameco. Connects to J1 and J2 if 13.8V from the radio is used to power the unit.
8-pin DIN cable	From IC-9700 to the sequencer	Available from Amazon and other places.
Power Connector		Whatever you like. I used this one from Amazon.
Amplifier, LNA, SDR Switch, etc. Connections	RCA and/or coaxial female receptacles	However many you need to connect all your equipment. Use whatever you like as connectors for the amplifier, preamp, SDR switch and power connectors.

USB to TTL converter		The one I used came from eBay. There are others available which should also work.
Penglin USB 3.0 Panel Mount USB Pass Through Connector		Available from Amazon.
Enclosure		I used one from Circuit Specialists; 6.7” x 5.9” x 2.8”.