YCCC MOAS II Testing

Revision history

1 February 2012 Tests on prototype 1 and prototype 2 complete.

Description

The YCCC MOAS II is a device which controls antenna relays. It communicates with a computer using a USB interface. And it communicates with radios by monitoring the amplifier T/R lines and by controlling the TX Inhibit lines when they are available.

Hardware

The switch hardware is very simple.

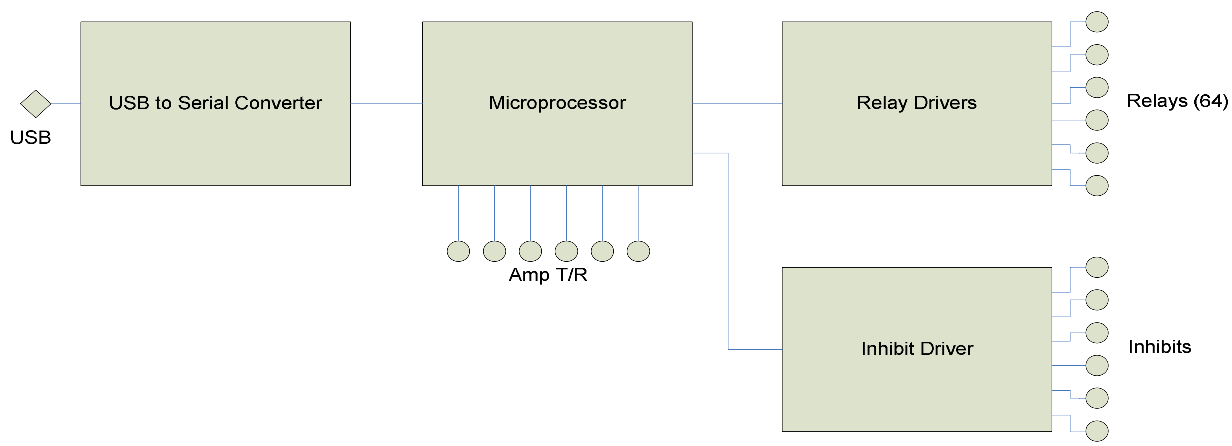

Figure 1. Hardware block diagram

The microprocessor communicates to the PC through a built-in UART and a serial to USB converter. It sends data to the relay drivers and inhibit driver using a three-wire serial protocol.

Types of tests

There are four types of tests:

Correctness

The commands are executed and the results verified to ensure that the right thing happens.

Performance

Timing-critical features are tested or the code is examined to show that the switch meets performance objectives.

Stress

Commands are sent or transmit/receive transitions are simulated at rates well beyond what would happen in actual use. The results are checked to see if the box functions properly under heavy load.

Outputs

Various combinations of driver chips are installed and the outputs are checked to ensure that the combinations work correctly.

Results of tests are shown in Indigo.

Correctness Tests

Each command is tested as described:

Character	Command	Description
!	Antenna	Set an antenna for a station
"	Status	Send status of the box
%	Conflict Table	Set or clear all or part of the conflict table
&	Fast Table	Set or clear all or part of the fast table
'	Ping	Send a ping response showing that the box is connected
(	Inhibit	Inhibit selected stations from transmitting
)	Uninhibit	Allow selected inhibited stations to transmit
*	Set state	Set box state or reset the box
/	Mode	Sets stations to wait or inhibit mode
:	Unit ID	Set or query the unit identifier
[	Inhibit Time	Set the number of milliseconds inhibit will be used when an antenna is switched
\	Receive Delay	Set the time after the amp key line drops before a station goes into receive
]	Forced Receive	Set the time after inhibit is set before a station goes into receive
^	Inhibit Polarity	Sets inhibit outputs high to inhibit or low to inhibit
|	Relay status	Send relay status
~	Inhibit Station	Sets stations to be inhibited when another station transmits

Antenna

Set antennas for receive, transmit, and extra and verify that the relay outputs are set appropriately. Set station 0 antennas and verify that they are set.

Relays were specified for several stations using B, R, T, and X, and they were observed to be set when those stations were in the appropriate states. Station 0 antennas were set and the changes appeared immediately.

Status

Send a status command and verify that the switch returns correct information.

Various antennas were selected. Status commands showed correct antennas and T/R state.

Conflict

Set several antennas to be in conflict. Set antennas and verify that the conflicting antennas cannot be set.

Several antennas were set to conflict. If a station was using one of the antennas and another tried to use it that station was not switched until the antenna became available.

Fast Table

Set some antenna combinations to be fast and others to be slow. Select these antennas. Verify that when switching between slow antenna combinations the station is temporarily inhibited. Verify that if the antennas chosen for transmit and receive have a slow transition the transmit antenna is used for receive.

Some antenna combinations were set to slow. When switching between these the station was inhibited briefly. A slow combination was selected for transmit and receive and the station used the transmit antenna always.

Ping

Send the ping command and verify the ping reply.

Ping commands were sent and replies were verified.

Inhibit

Inhibit various stations and verify that the inhibit outputs are set.

Inhibit commands were sent and the results were verified.

Uninhibit

Uninhibit various stations and verify that the inhibit outputs are cleared.

Inhibit commands were sent and the results were verified.

Set State

Turn off the resolver and set some antennas. Verify that they re not set by observing the relays and by using the Status command. Turn the resolver on and verify that the antennas are now set.

Reset the box to power on state and verify that all relays and inhibits are cleared and that antenna commands are not accepted.

Set the box to operate mode and verify that antenna commands are accepted and that they change the outputs.

Turn Antenna, Transmit, and Conflict messages and set antennas and assert transmit on one or amplifier T/R lines. Verify that the correct messages are received from the box.

Reset sets the box back to initial state. Relay outputs are cleared. Inhibit outputs are cleared. Antenna commands using B, T, and R produce an error. In operate mode they do not produce errors.

The resolver was turned off and antenna commands sent. They were not processed until it was turned on.

Antenna, transmit, and conflict messages were observed during other testing.

Mode

Set a station in interrupt mode. Verify that if the antenna is changed while it is transmitting that the antenna change occurs. Verify that it does not occur when the station is in wait mode.

Set one station in interrupt mode and set another in wait mode. Create a conflict so that both will be resolved when one stops transmitting. Verify proper behavior.

A station was set in interrupt mode. When the antenna was changed the station went to receive if forced receive delay was set, then changed antennas, and was inhibited for the slow delay time whether or not the transition was supposed to be slow.

A station was set in wait mode. When the antenna was changed the station stayed in transmit and did not change transmit antennas until it went to receive.

One station was set to interrupt and one to wait. They were alternately set in conflict while transmitting. If one station was transmitting when the conflict was resolved the antenna change was done according to that station's mode. If both stations were transmitting the antenna change was resolved in wait mode.

Unit ID

Set the unit ID to a non-zero number and verify that it has been changed.

The unit ID was changed. The box was reset and the unit ID remained at the new value.

Inhibit Time

Set the inhibit time to 9999 ms. Change antennas using a slow transition and verify that the station is inhibited for approximately ten seconds.

The inhibit time was set to 9999 ms. A slow antenna transition was tested. The time is approximately ten seconds.

Receive Delay

Set the receive delay for a station to 9999 ms. Set the station to use different transmit and receive relays with a fast transition. Switch between transmit and receive and verify that the relays change after ten seconds and that if active the receive message also is delivered after ten seconds.

Start transmitting again before the receive delay times out. Return to receive and verify that the box does not go into receive until the full delay occurs.

The time was set to 9999 ms. The station was transmitting. When it stopped the antenna changed after approximately ten seconds.

The station transmitted several times. The antenna changed ten seconds after the end of the last transmission.

Forced Receive

Set a station to interrupt mode. Set the forced receive time to 99 ms. Verify that the station is inhibited for the correct time and then the antenna change occurs.

Set the forced receive time to 0 ms and verify that no inhibit occurs before the antenna is changed.

Verify that all changes are slow transitions even if the fast table shows a fast transition.

This was tested as part of mode testing.

Inhibit Polarity

Change the inhibit polarity for a station and verify that it has changed. Power cycle the unit and verify that stations are inhibited according to their polarity until the unit is set into operate mode.

Inhibit polarity was changed for several stations and the results were verified.

Relay Status

Set some relays using various antenna commands and verify that the status command shows the correct relays.

Several relays were set. The relays status was correct in all cases.

Inhibit Station

Set a station to inhibit another station. Set that station to transmit and verify that the other station is inhibited and cannot transmit.

Set two stations to inhibit each other. Verify that the first one transmitting locks out the other and that if the other transmits it does not cause a reverse lockout or instability. Verify that an attempt to set a station to inhibit itself results in an error.

Two stations were set to inhibit each other. When one station was transmitting the other was inhibited. Setting the other station to transmit did not result in a change of antennas and did not inhibit the transmitting station.

Attempting to set a station to inhibit itself results in an error.

Performance

There are two ways to determine performance. One is to examine the code and count the number of cycles. The other is to measure an output using a device such as an oscilloscope or period counter.

There is one overriding performance goal for this box: When a station changes between receive and transmit the relay outputs must change quickly. The worst case time should be less than one half millisecond.

The delay comes from several places. The box receives an interrupt when a T/R line changes state. It then must recalculate the relays by adding the relays for all stations in their current states (transmitting or receiving). It then sends this information to the relay drivers. The relay drivers can only accept the information relatively slowly. And the inhibit information must be recalculated and output to the inhibit driver.

If interrupts are turned off when the T/R line changes state the interrupt routine will not be called. It will start when interrupts are turned on. The interrupt used for T/R line changes is the highest priority interrupt used by the box so it will always be the first interrupt serviced when interrupts are enabled.

This is the list of places where interrupts are turned off during normal operation:

- Pin change interrupt

  The pin change interrupt is serviced when a T/R line changes state.

The code was instrumented to turn on an unused microprocessor output pin at the start of the interrupt service routine and to turn it off at the end of the routine. This time was measured with an oscilloscope to be approximately 80 µs.

Examination of the code shows that if a station is in transmit with a receive delay it can add almost 1 µs to the time measured. Worst case this adds 6 µs to the measured time.

There is also a small amount of time, less than 1 µs, taken to enter and exit the interrupt service routine.

Worst case for this routine is less than 90 µs.

- Timer interrupt

  The timer interrupts every 500 µs. Interrupts are turned off while the timer interrupt is being serviced.

This code was examined and cycles were counted.

When no timers are running this interrupt completes in less than 2 µs. The worst case, with all timers running and expiring at the same time would be 11 µs. In practice the latency would be less because per-station timer expiration updates the relays and this would also pick up the pin change.

- UART input

  The UART interrupts whenever a character is received.

This code was examined and cycles were counted.

The interrupt will always complete in less than 4 µs.

- UART output

  The UART interrupts whenever it is ready to send a character.

This code was examined and cycles were counted.

The interrupt will always complete in less than 3 µs.

- Antenna command processing,

The command processing turns off interrupts while setting the global (station 0) relays.

This code was examined and cycles were counted.

The interrupts will be off for less than 4 µs.

- Status store

  The status store turns off interrupts while adding a character to outgoing UART buffer.

This code was examined and cycles were counted.

The interrupts will be off for less than 1 µs.

- Resolver

  The resolver turns off interrupts moving pending station and relays to current and actual station and relays.

The code was instrumented to turn on an unused microprocessor output pin when the resolver turned off interrupts and to turn the pin on when the resolver would branch to the relay update code as this code would pick up any T/R line changes.

The resolver was turned off, antennas were changed, and then the resolver was turned on. This allowed for simultaneous changes of multiple antennas.

The resolver time depends on how many relays are changed.

1 Change	8 µs
2 Changes	13 µs
4 Changes	23 µs
6 Changes	32 µs
8 Changes	40 µs
10 Changes	48 µs
12 Changes	58 µs

A change was made in the code after the measurements were made which would add less than one µs. Worst case in this code could add another 5 µs if all transitions are slow but the resulting T/R choices are fast. The worst case, which will never happen in actual operation, is about 65 µs.

The resolver also turns off interrupts when copying extra relays.

The code was examined and cycles were counted.

The interrupts will be off for less than 4 µs.

The worst case is when a T/R line changes just after interrupts are turned off. In this case the latency will be that of the pin change interrupt service routine plus the time interrupts are off.

The longest time interrupts are off is during the T/R line change interrupt routine. So the worst case latency would occur if a T/R line changes while the box is already executing the interrupt routine. The routine reads the line status fairly early but not at the beginning of the routine. So the worst case latency is significantly less than 180 µs.

The box exceeds the minimum performance requirement.

Stress

The box should be able to handle more load than it will ever see in normal operation and to degrade smoothly if capabilities are exceeded.

Resolver

The conflict resolver algorithm involves nested loops. The number of loop iterations to check all combinations of antennas is 2^n, where n is the number of pending antenna changes. If a command is received or a T/R line changes the resolver abandons the attempt and tries again. If the resolution takes too long it could never complete.

Also if there are many conflicts but some antennas that can be changed if the resolver takes a long time it would be noticeable to the operators.

The code was instrumented to turn on an unused microprocessor output pin when the resolver was called and to turn it off when the resolver returned.

No work:	1.5 µs
2 Conflicts:	35 µs
11 Conflicts:	40 ms

The resolver is called in the main loop but returns quickly if there are no antenna changes to process. Most changes do not cause conflicts.

One example of a change which does cause a conflict is when two stations are swapping antennas – after the first station selects the antenna being used by the second station it will be in conflict until the second station selects the antenna being used by the first station. If both transmit and receive antennas are being changed there will be two conflicts. The time spent trying to resolve this is insignificant.

The worst case would be eleven conflicts which cannot be resolved. If this happens the resolver will spend 40 ms trying to resolve the conflicts. Even in very busy conditions there will be times of at least a twentieth of a second when no station is switching between transmit and receive or changing antennas.

This would also be the worst case time required to switch one antenna if there are ten conflicts. 40ms is below what people perceive as a delay so the switch would appear instantaneous in all cases.

Stations 1 Transmit, 2-5 transmit and receive, and 6 transmit were set in conflict such that station 6 receive had the resource needed to resolve it. Station 1 receive antenna was changed and the output relay changes were observed. A LED was set to turn on or off when a command was received. Without the LED it was not possible to perceive the delay. With the LED it was possible to see a very short delay.

And ten or eleven conflicts will only occur if someone is deliberately attempting to cause it or if the controlling software is severely faulty.

Command Buffer

If the box is fed commands non-stop it should be able to handle them without dropping any commands.

At 9600 baud characters arrive at about 1 ms per character. Characters are taken from the UART and placed in a circular 128 byte buffer. In order for this buffer to overflow 128 characters must be in the buffer not processed by the command handler.

Code examination shows no long loops in the command processing. Stress tests have involved sending 60,000 commands to the box with no pauses. No issues have been observed.

Status Buffer

Some commands are shorter than the output they generate. If these commands are sent continuously the output buffer should overflow. The box may not send some responses but all responses sent should be complete. Similarly, if transmit/receive messages are being sent by the box and a station changes transmit/receive state frequently enough messages may be lost but those sent should be complete.

And in no cases should this degrade performance of the box, either in terms of switching antennas or general command handling.

1,000 status requests were sent to the box in a tight loop. 197 responses were returned. All were formatted correctly. 10,000 status request were sent and some returned data was truncated. This is believed to be due to a limitation of the test program which probably cannot handle 140,000 queued messages.

Transmit/Receive

The box must respond quickly and correctly to transmit/receive status changes at a rate beyond what would occur at a busy station.

A square wave generator was connected to an antenna input through a transistor. Transmit and receive were set to two different stations with different antennas. An oscilloscope was connected to the relay output for the receive station.

The frequency of the generator was increased until the oscilloscope started to show jitter in the output. This occurred at approximately 6400 Hz. This represents 12,800 T/R changes per second. The generator does not have quite a perfect 50% duty cycle so the maximum might be slightly higher. Occasional commands were sent, these were processed with no noticeable effect. Status messages from the box were lost, but this was expected. All received status messages were correct.

The generator frequency was reduced to 1000 Hz. A set of commands to set another station's antennas to one antenna and relays and back to another were sent in a loop 10,000 times, for a total of 20,000 commands. An oscilloscope was connected to the relay output that was being changed. The generator was connected and disconnected. An occasional jitter could be seen. The time between antenna changes is about 6 ms, which is due to the serial line speed, and the jitter was a small fraction of a millisecond. The jitter is probably due to the interrupts being off briefly when the antenna change is stored.

Outputs Tests

Each group of eight outputs can be set for either source or sink, or may not be set up at all. Various combinations must be tested.

A test plug with eight LEDs as built. It was used to confirm proper switch operation.

There are 512 possible useful combinations of source and sink outputs. This is too many to test. A few cases were selected involving source only, sink only, source following sink and sink following source. The following table shows what was tested.

1 – 8	9 – 16	17 – 24	25 – 32	33 – 40	41 – 48	49 – 56	57 – 64
Source							
Source	Source	Source	Source				
Source	Source	Source	Source	Source	Source	Source	Source
Source	Source	Source	Source	Source	Source	Source	Sink
Sink	Source	Source	Source	Source	Source	Source	Sink
Sink	Sink	Source	Source	Source	Source	Source	Source
Sink	Sink	Sink	Sink	Source	Source	Source	Source
Sink	Sink	Sink	Sink	Sink	Sink	Sink	Sink

A firmware bug was found and fixed during this testing. The change had no effect on any timing.

All test cases were successful.

Inhibit output and T/R input function was also verified.