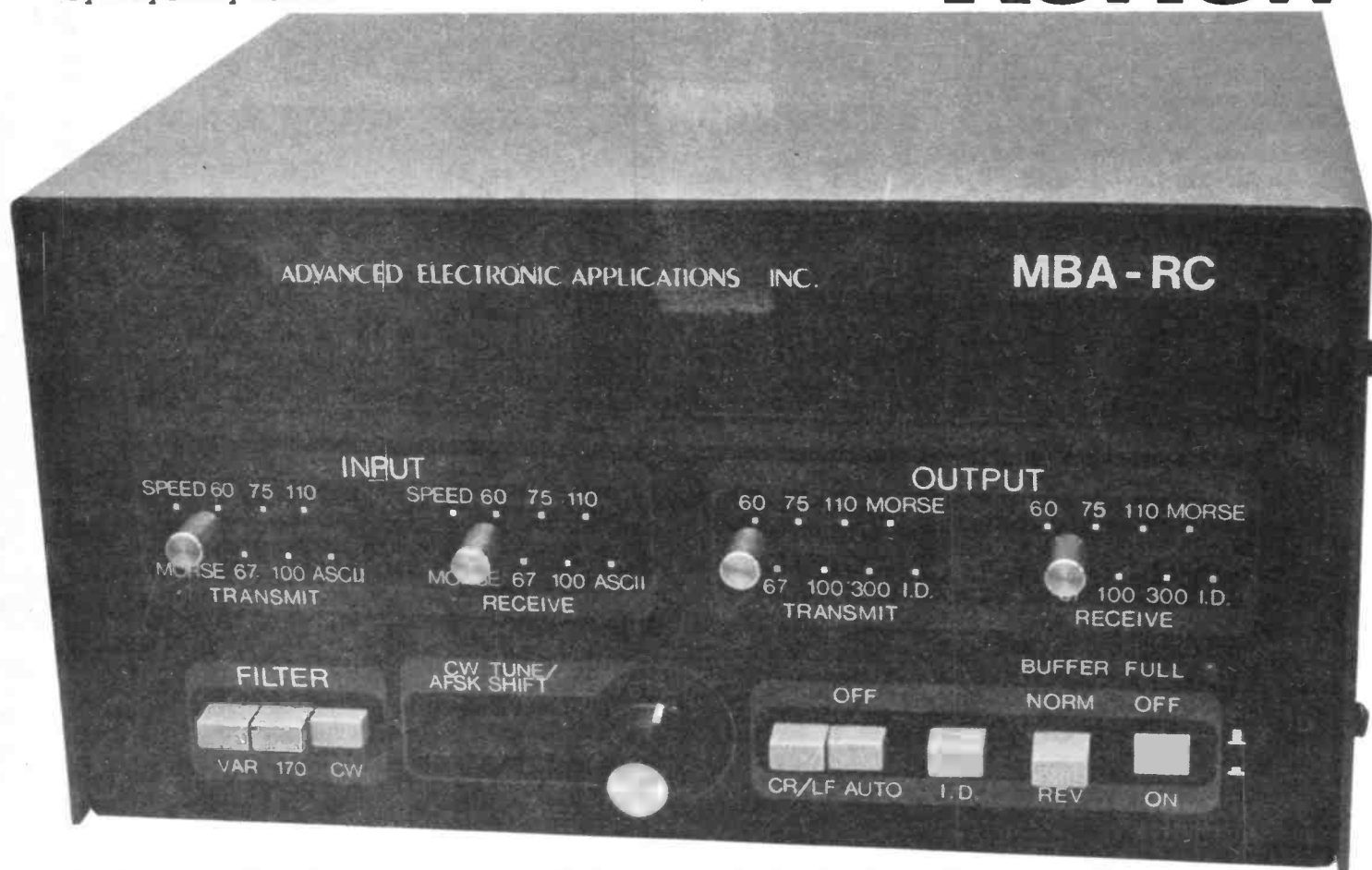


MBA-RC

Review

By Tony Bailey G3WPO



In the very first issue of this magazine, we reviewed the CWR-600 morse/RTTY decoder. This somewhat larger box of tricks from a different stable will do exactly the same thing, with the advantage of having a 32 character alpha-numeric display built in to it, so you don't have to tie it up to a monitor or TV. It is one of a range of units, and is the top-of-the-line version.

However, as it costs somewhere around twice the price of the CWR-600, you might guess that it has a few tricks up its sleeve — and you would be right. For a start, it is also a Transmit as well as Receive unit, coping with RTTY, Morse and ASCII transmissions. So that's what the extra money gets you then?

Well, not quite, the best is yet to come. Frank said to me "Want to review an RTTY decoder?". "Of course, but I can't send RTTY with it, unless it'll hook up to my TRS-80", replies I. "Want a bet", replies Frank, "You've got a Morse key, haven't you?".

RTTY with a morse key?

That's exactly what I said... but it does just that. You will notice from the front panel illustration that there are four separate slide switches, each marked with similar sets of mode information. The left hand side pair sets what happens in the receive (INPUT) mode. Slightly confusing because the left switch is marked Transmit and the next one

Receive. What it means is, "Receive" is the input TO the unit FROM the receiver audio output. i.e., if you are receiving Morse Code, then you set the INPUT Receive switch to "MORSE".

By Transmit it means what YOU are transmitting TO the unit. The reason for this is that these two settings don't have to be the same, as we will demonstrate in a moment. So, if you are inputting 45.5 Baud RTTY from a teletype machine into the unit, then you set the INPUT Transmit switch to "60" (which is 60 wpm RTTY = 45.5 Baud).

Moving onto the OUTPUT side, you will have guessed that this is the transmit side, except why two switches? Well, the OUTPUT Transmit switch is what you would expect — it