RVP7 V25 Release Notes

These notes cover changes made to the RVP7 code since release V24 of 25 March 2002. If you are upgrading from an earlier release, please read those notes also.

Bug Repairs

1. A bug was repaired that has been present since V15 that would cause the powerup receiver noise levels to be initialized incorrectly for three of the four pulsewidths. In particular, only the "current" pulsewidth from the **Mt** menu would startup with exactly the correct noise level. The other incorrect values (usually wrong by a few dB) would persist until such time that a real noise sample was taken for each pulsewidth. Those real noise levels would then remain fine thereafter. We would like to thank the dedicated engineers at MELCO for their time and patience in tracking down this bug.

New Features

1. The CFGPHZ (Configure Phase Modulation) command has been changed so as not to return the *Could not generate phase sequence* error when random phase is requested but the phase control output lines are not configured for phase modulation. This helps work around a bug in IRIS 7.28 in which random phase modulation was being requested for Magnetron systems.

2. The RVP7 is now capable of calibrating the external phase shifter that is attached to its phase control output lines. This automatic procedure is invoked through the **Mz** menu and requires only that the phase shifted COHO waveform be applied to the Burst Input of the RVP7/IFD. This is easy to accomplish because that is the default wiring for normal operation. Amazingly, no other reference signals are needed to perform the calibration. See also Setup Change #1.

3. The reloading of trigger patterns into the trigger generator hardware is now considerably faster. Previously, triggers would be missing for approximately 3.6ms whenever the PRF was changed; this has been reduced to under 100 μsec. The change affects radar systems that could not tolerate even short gaps in the triggers they receive from the RVP7. Note that all of the trigger generator's minimum PRT constraints remain enforced even during this speedup.

4. The RVP7 external trigger input can now be used as a synchronizing clock for internal trigger generation. In this mode a continuous clock is applied to the trigger input line, for the purpose of quantizing the precise start time of each pulse. This is different from the normal external trigger case in that the trigger rate is still chosen internally by the RVP7; but each pulse will not begin until an edge of the input clock is seen. This feature is useful when the RVP7/IFD is phase locked to a reference clock, but the RVP7/Main triggers must be further locked to some submultiple of that reference clock. See also Setup Change #3.

5. The RVP7 now supports ten (rather than six) user-programmable triggers as a special order option. Please contact SIGMET for more information about obtaining this feature if you really need it.

Setup Changes

1. The **Mz** (Transmitter Phase) menu now includes new features to calibrate the external phase shifting hardware. The normal **Mz** command behaves exactly as before, but two new variants are introduced.

   Typing **Mzv** will view the existing phase calibration table, producing a printout such as the following:

```
                    Working Table of Phase Angles
                    -----------------------------
Hex:Angle (Binary)  Hex:Angle (Binary)  Hex:Angle (Binary)  Hex:Angle (Binary)
------------------  ------------------  ------------------  ------------------
00:  0.00 (  0.00)  10: 22.50 ( 16.00)  20: 45.00 ( 32.00)  30: 67.50 ( 48.00)
40: 90.00 ( 64.00)  50:112.50 ( 80.00)  60:135.00 ( 96.00)  70:157.50 (112.00)
80:180.00 (128.00)  90:202.50 (144.00)  A0:225.00 (160.00)  B0:247.50 (176.00)
C0:270.00 (192.00)  D0:292.50 (208.00)  E0:315.00 (224.00)  F0:337.50 (240.00)
```

   In this example, sixteen equally spaced phase codes have been defined. For each code, its 8-bit Hex value is shown, followed by the phase angle both in degrees (0–360) and binary angle units (0–256). The factory default phases are shown here, which explains all of the perfect round numbers.

   Typing **Mzc** will perform an automatic calibration of the phase control hardware. The complete set of valid codes are tested, and phase shifts relative to the idle code are reported. Each measurement requires 100 pulses of data, and the trigger ***must*** remain at a fixed PRF for the duration of the measurements. Progress messages are printed, and the final report will resemble:

```
                      Table of Measured Phase Angles
                   (Jitter – Min:0.41, Avg:0.48, Max:0.58)
                   ---------------------------------------
Hex:Angle (Binary)  Hex:Angle (Binary)  Hex:Angle (Binary)  Hex:Angle (Binary)
------------------  ------------------  ------------------  ------------------
00:359.99 (255.99)  10: 20.66 ( 14.69)  20: 45.26 ( 32.19)  30: 67.23 ( 47.80)
40: 89.71 ( 63.79)  50:114.48 ( 81.41)  60:134.73 ( 95.81)  70:157.38 (111.92)
80:180.68 (128.48)  90:201.39 (143.21)  A0:226.04 (160.74)  B0:248.10 (176.43)
C0:270.68 (192.48)  D0:295.04 (209.81)  E0:315.52 (224.37)  F0:337.86 (240.25)
```

   These are the actual measured results of a 4-bit phase modulator. The table of angles has the same interpretation as for the **Mzv** printout. The measurement errors are summarized in the header, and show the minimum, average, and maximum phase jitter (in degrees) for the entire procedure. The minimum jitter is useful as a test of basic system clock stability, since it most likely results from the iteration that compared the idle phase code with itself. In this case, the value of 0.41 degrees is really a measure of the phase stability of the signal generator being used for the bench test.

   As a special case, you may append a “j” to the phase calibration command, which will modify the printout to show the individual phase jitter (in degrees) for each angle, rather than the binary equivalent of those angles. Thus, typing **Mzcj** will perform a phase

calibration and display the individual angle jitters for your inspection. The detailed jitter information is only available in this one printout; it is not retained, even if you decide to save this set of calibration angles.

After the table is printed you will be prompted with the question: *Keep this calibration?* If the values seem reasonable, then you may overwrite the current working angles by answering *Yes*. Otherwise, the new values are discarded and the RVP7 continues to run with its old angle table. Note that you must still use the **save** command if you want the kept angle table to persist beyond the next power cycle.

2. The **Mc** question to configure the PHOUT lines now reads:

```
PHOUT[7:0] usage– 0:Off, 1:XMTPhase, 2:DigAFC, 3:XMTPh+CLKs : 0
```

The difference is in item #3 — when you choose the special clock output option on *PHOUT<7:6>*, the transmitter phase control will continue to run normally on the six remaining *PHOUT<5:0>* lines. Previously those six lines were inactive when clock output was selected.

3. The **Mt** question to select external triggers now reads:

```
Trigger Method – 0:Internal, 1:ExtPretrig, 2:ExtSync : 0
```

The “0” and “1” choices are identical to the previous “internal” and “external” trigger options. However, choice #2 is new, and is used to select the trigger synchronization mode described in New Feature #4. Making this choice also properly requires that the trigger PRT quantization be set to match the rate of the synchronizing clock.

As an example, suppose that the RVP7/IFD uses a 36MHz sampling clock that is phase locked to a 10MHz external reference. With just this much in place, the RVP7/Main internally synthesized PRTs will be guaranteed free of jitter relative to the 10MHz reference. However, the relative phase between the triggers and the reference clock will be entirely random, and will even change each time the PRF or pulsewidth is changed. This is because the triggers are generated at the 36MHz rate, and the 10MHz absolute phase information never leaves the RVP7/IFD PLL circuitry.

The situation changes if a submultiple of the 10MHz reference is also applied to the RVP7/Main trigger input, and the “ExtSync” menu option is selected. Now the trigger generator will wait for an edge of the TrigIN clock prior to starting each trigger pulse. Triggers are therefore produced at a programmable rate, just as for normal internal triggers, but with an absolute starting phase that is locked to an external reference. If a 1MHz (Ref/10) TrigIN clock is applied, then the corresponding PRT quantization should be ***((36 x AQ) + 24)*** clocks, so that requested trigger periods are rounded to the nearest Nμsec plus 0.75μsec, where N is an integer. This leaves 0.25μsec to resync to the next 1MHz TrigIN clock edge on each pulse.