

## RCP02 V15 Release Notes

These notes cover changes made to the RCP02 code since release V14 of 2 September 1998. If you are upgrading from an earlier release, please read those notes also.

### Bug Repairs

1. Repaired an error in the setting of elevation limit switch status when the optional Andrew-Kintec pedestal option is selected. An incorrect bit was being checked in the ACU antenna status block.
2. Repaired a bug in which IP module selections would not be recognized when they were typed in.

### New Features

1. It is now possible to write logic equations in which status variables appear on the left-hand side. The meaning given to such assignments is that the working value of the status variable is modified from its default "requested" value, i.e., the value being assigned from whatever hardware line or external condition is normally attached to the status bit. In this sense, the modification of a status variable is identical to the modification of a control variable. In both cases, when the variable appears on the right-hand side, it refers to its default "requested" value.
2. The "Monitor Status" command now has subcategories very similar to the "Monitor Control" command. You may view both the requested and qualified versions of the primary and auxiliary status bits, as well as the direct hardware inputs themselves. The distinction between requested and qualified status bits now exists because the status bits can appear on the left side of logic equations.
3. The error message that is printed when an ambiguous variable name is typed into a logic equation now includes a list of all of the possible matches. This can help you identify how to type the variable uniquely.
4. The logic equation editor now prints a warning upon exit if multiple assignments are being made to the same variable. For normal status and control variables this almost certainly indicates an error, since only the final assignment will have any lasting effect. For local variables, however, multiple sequential assignments are meaningful since an assignment on one line may be referenced on a subsequent line. For now, the warning is printed even for local variables since there is still a chance that that is not what you had intended to do. Spurious local variable warnings can be eliminated by choosing a unique set of numbered variables to use (i.e., by not reusing them within the overall set of equations).
5. Error messages from within the equation editor no longer cover up the original typed-in line that produced the message.

6. When the optional Andrew–Kintec pedestal option is selected, the following status lines are derived from serial data from the ACU (rather than from electrical inputs at the RCP02 backpanel): LOCAL, SERVOPWR, ELIMLO, ELIMHI, and INTERLOCK. These bits are now qualified by the Enable/Disable questions in the "status" setup menu, i.e., each bit must be enabled in that menu in order to be active. Previously, the Andrew bits were always active and could not be disabled.

## Notes and Examples

1. In understanding the operation of the timer variables described in the last release, it may be helpful to note that the output of an "extend" timer is logically equivalent to the negation of the output of a "retard" timer whose input is also negated. Although these two timer classes are merely inverted-logic duals of each other, it is still conceptually useful to have both the "retard" and "extend" concepts. An analogy is that "AND" and "OR" are both useful logic concepts, even though an OR-gate is merely an AND-gate with inverted inputs and outputs.
2. When writing sets of logic equations for the RCP02, keep in mind that assignments to most types of variables can not be referenced as such on subsequent lines. When control and status variables appear on the right side of an equation, they *always* refer to their original requested value. Assignments made on the left will modify the variable's effective working value; but the original requested value still remains unchanged. This is why it is never correct to make more than one assignment to the same control or status variable, and why the pair of equations:

```
EQ00: cpw0 = cpw1
EQ01: cpw1 = cpw0
```

would swap the two pulsewidth control lines without the use of the temporary intermediate variable that would normally be required for sequential assignments. The only variables that can be referenced immediately after being assigned are the local variables V[0:15]. Thus, the pair of equations:

```
EQ00: v0 = v1
EQ01: v1 = v0
```

would *not* swap the two local variables, but instead, would leave both set to the original value of V1 (probably not useful).