

Coder 5.02-0 Release Notes

DATE:

March 15, 2010

BACKGROUND:

Version 5.02-0 is a new release of the Coder/Opsys

WARNING: To use this version of GAP/Coder, you must use GAP 2.18C or higher.

WARNING: To use this version of GAP/Coder, you must use SOS 3.4 or higher.

WARNING: To use this version of GAP/Coder, you must use Toolkit 3.3 update 2 or higher.

NEW FEATURES:

The major changes from 5.01 to 5.02 for the ECM3, MCU and PCMHD controls were:

- **Additional RAM:** 4K extra bytes were made available on the PCMHD (Low Volume and High Volume) through code optimization and more effective use of processor resources
- **PCMHD EEPROM health:** To extend the life of the EEPROM on the PCMHD, an `AUTO_SAVE` field was added to the [PCM STATUS](#) block. This field allows the user to limit the number of writes made to the serial EEPROM by disabling the *auto write to serial EEPROM* feature currently used in the following blocks: [DM7 30 MSG](#), [DM25 MSG](#), [EVENT MGR](#), [NV INC](#), [I NV LOG](#) and [NV LOG](#)
- **[Event Manager](#) / DM Mil Enhancements** for more rigorous compliance with OBD standards:
 - Events have a new attribute, "MIL", that can be used to meet OBD standards
 - It is possible to select either *timer* format or *date/time* format for HMI time values (supported in Toolkit 3.3 update 2 and later)
- J1939 requests for multi-packet messages are now responded to correctly for both global and specific destination addresses

The major change from 5.01 to 5.02 for the Atlas and Micronet controls was:

- **New application start permissive:** Before an application starts, the control verifies that it is compatible with the footprint part number and Operating System version
- J1939 requests for multi-packet messages are now responded to correctly for both global and specific destination addresses

ISSUES:

The following Product Issue Database ("PID") issues and requests were addressed in 5.02-0:

Issue #	Description	Solution
11272	<p>Toolkit tunable upload problem:</p> <p>Toolkit attempts to load tunable files to the control when shutdown is disabled</p>	<p>The control's response to disallowed shutdown and reset requests from Servlink was corrected. Toolkit will only upload tunables properly with controls which contain this fix</p>
12211	<p>Long GAP3 names:</p> <p>Request to allow longer GAP Category and Block names</p>	<p>The new GAP3 editor program allows Category and Block-name lengths of 20 characters. The coder was modified to accommodate this change with increased buffer sizes.</p> <p>On controls on which the SID file is stored, longer names will cause slightly higher flash memory consumption. On control without a stored SID file, the longer names will have no adverse impact on flash consumption</p>
14384	<p>Servlink protocol error:</p> <p>Servlink returns an <i>increment too large</i> error code when an attempt is made to tune a value out of range</p>	<p>The error code was corrected to <i>value out of range</i></p>
14424	<p>Pattern 14 error on engine stop:</p> <p>When Pattern 14 is selected on the PCMHD, the EFI P CORE.RAW_SPEED output initially reads 198 RPM after boot. Once a start attempt is made, the value reads correctly, but when the engine is stopped the value then only goes down to ~6 RPM</p>	<p>Updated TPU_Sync_CmRES.uc and TPU_Sync_CrRES.uc to use an immediate byte when adding to the test byte to set the carry flag. This allows stall to set correctly and raw speed to go to zero when the engine shuts down</p>
14599	<p>Actuator Control calibration problem:</p> <p>ACT_CTRL blocks in an RTN chassis can lose the calibration on channel #1</p>	<p>Corrected the value which the RTN chassis uses to decide when to perform calibration</p>
14696	<p>J1939 Address Claim problem:</p> <p>The J1939_NTW block is supposed to give up its claimed address if another node with higher priority requests the</p>	<p>Corrected the address claiming algorithm in the control code</p> <p>Added a NACK response to poorly formatted messages</p>

	<p>same address. This is not functioning: the node with the lower priority is not giving up its address</p> <p>In addition, controls will not respond to improperly formatted address claim request messages. The message is required to have 3 bytes of data, if the wrong number of bytes is sent, the message is thrown away. The control should respond with a NACK indicating the incorrect message will not be processed</p>	
14698	<p>Event Manager flash inefficiency:</p> <p>The application consumes too much flash (~340bytes) for each HMI_PT block connected to the EVENT_MGR block</p>	Corrected bug which caused redundant code to be added to the control program
14705	<p>EE available size error:</p> <p>The amount of EE memory reported available (EE_AVAIL) is off by 1032 bytes if the application contains an I_NV_LOG block</p>	Corrected computation
14709	<p>DM25_MSG block compile problem:</p> <p>There is an issue when you have an EVENT_MGR block with an HMI_PT and a DM_MANGR block with a DM25_MSG block. If there are more event repeat groups in the EVENT_MGR than there are DM_MSG repeat groups in the DM_MANGR, the coder will exit with a general protection fault</p>	Added correct variable structure for nested loop used to build the DM25_MSG freeze frame SID entry in function ProcessSIDFreezeFrame
14710	<p>WRITE_I block problem:</p> <p>The Error and Not Supported inputs on the WRITE_I block are not looked at when data is being read in for messages like DM25_MSG. They need to act as they are described in help for the WRITE_I block</p>	Added support for ERROR and NOT_SUPP inputs to the DM25_MSG and DM7_30_MSG code
14711	DM25_MSG repeat group limit wrong:	Changed RPTdtc limit in DM25_MSG block from 99 to 400 to

	<p>The RPTdtc group is limited to 99 repeats. The max number of repeats needs to be increased to match the max number of repeat groups allowed in the EVENT_MGR RPTin group. The DM25_MSG repeat group is the list of events that will toggle a freeze frame. This is possibly every event in the EVENT_MGR block</p>	<p>match max events allowed in the EVENT_MGR block</p>
14712	<p>DM25 Message Help:</p> <p>Update DM25_MSG Block Help topic to more clearly identify what causes a freeze frame to be stored and when it will be cleared. Example question from applications:</p> <p><i>Do you automatically clear the appropriate freeze frame when the event is deleted from the log either due to self clearing or from a DM or Toolkit clear? So if there is only one freeze frame you clear it out if the event that caused the freeze frame is gone?</i></p> <p>It was also requested that we add the repeat limits to help for planning purposes</p>	<p>Updated Help without adding repeat limits. The repeat limits may be viewed by looking at the field help in GAP 3.0</p>
14714	<p>DM25_MSG HMI display issue:</p> <p>The freeze frame component that is created in Toolkit needs to display integer data as unsigned. This needs to be added to the SID file. The data type defaults to signed</p>	<p>Changed SID file freeze frame data attribute to unsigned</p>
14719	<p>Event Manager enhancement:</p> <p>The EVENT_MGR block needs to be changed so that it ignores a specific event once occurrence count hits 127. No writes to EE, etc. A reset will still clear the fault as it does today. Need to make available the number of times the event manager writes to the EE during a given cycle for application testing. Need to clarify sufficient resolution, number of</p>	<p>Changed the EVENT_MGR block to clamp occurrence count to 127. Timers will continue to update, however event information for that event occurring after the occurrence count reaches 127 will not be saved to the EEPROM</p> <p>Implemented new EE save strategy enhancement which uses the AUTO_SAVE input on the</p>

	times each shelf is written to, number of times each byte in each shelf, number of times event manager causes the EE to update..	PCM STATUS block to disable automatic saving to the serial EEprom
14735	<p>Momentary Boolean block optimization problem:</p> <p>The MOMNTARY B block needs to be excluded from optimization. If an HMI_PT is connected with any ram optimization priority except 0 and the outputs are not connected to anything, the coder program exits</p>	Added the MOMNTARY B block to the list of blocks which are excluded from RAM optimization
14741	<p>DM7_30_MSG block ID overflow:</p> <p>The DM7 30 MSG block has an issue when it creates the IDTab array to hold the ID_X values defined to reference the associated event manager. The array is of type uint8 but must reference ids from 12-1000</p>	Changed IDTab array created for DM7 30 MSG block from type uint8 to type uint16
14752	<p>Missing Block Help topics:</p> <p>No Block Help available for B32 TO I or I TO 32B blocks</p>	Added requested topics to Block Help
14753	<p>J1939 BAM incorrect response to RQST:</p> <p>In coder 5.01 J1939_BAM replies incorrectly to a RQST message. J1939_BAM should reply with BAM if the RQST message is sent to the global address (255) and reply with CTS/RTS if the RQST message is sent to a destination specific address. J1939_BAM seems to filter on the source address of the RQST, not the destination address of the RQST. The result is that J1939_BAM always responds with CTS/RTS, unless the source address of the RQST is set to 255 (which would violate J1939 rules). This means that the originator of the RQST no longer receives data from J1939_BAM</p>	Corrected implementation of requested messages

14765	<p>Fieldbus Master reset trigger behavior:</p> <p>The FBUS_M->RST input is currently an “edge” triggered input. It looks like in versions before 4.03 it was “level” triggered (as the help states) and 4.03 changed it to make it “edge” triggered. The Block Help for FBUS_M should describe the RESET as edge triggered</p>	Updated Help document to reflect that RST is an edge rather than level triggered event in this version
14780	<p>Remove CAN_RD_WR Block:</p> <p>The CAN_RD_WR block topic is still listed in the help document, but is not allowed by the coder program</p>	Removed the CAN_RD_WR block from the Block Help and template
14887	<p>CAN_READ Block Help incorrect:</p> <p>Remove references to the obsolete “MSG_TYPE” field</p>	Block Help updated
14794	<p>DM25 MSG Block freeze frame issue:</p> <p>If the event that caused a freeze frame is cleared and there are other active events logged, the newest active event causes another freeze frame to be triggered. This is deceptive as it will capture data at the time the one event was cleared and make it look like it was captured at the time the next newest event went active. What should happen is: event 1 goes true causes freeze frame event 2 goes true causes new freeze frame event 3 goes true causes new freeze frame event 3 goes away, freeze frame goes away. What is happening: event 1 goes true causes freeze frame event 2 goes true causes new freeze frame event 3 goes true causes new freeze frame event 3 goes away, freeze frame goes away and a new one is triggered by event 2 even though it was already active when event 3 happened.</p>	Freeze frame behavior corrected
14797	<p>J1939 PDU1 response problem:</p> <p>If a message from, e.g., the J1939 BAM block or the CAN_WRITE block set to</p>	Added PGN filtering for PDU type 1

	<p>"on request", the J1939_BAM block will not respond to a PDU1 type request unless the message number has the SA byte == 0. if MSG_NMBR= 256, PF = 1 SA = 0. I can request 0x0100 and it responds. If I set it to MSG_NMBR 257 PF =1 SA = 1 and request 0x0100 I get a NACK (it should respond), if I request 0x0101, I get a NACK (this is correct). The J1939 stack should be masking off the SA portion of the MSG_NMB input when trying to match a request. The Request PGN (in the incoming RQST) must have the SA = 0 for a response to happen, but to be friendly, the coder side should be smart enough to respond to a matching PF</p>	
14816	<p>SYNCHRO Block Help incomplete: SYNCHRO block help does not describe the fields SLIP_F_REF or DWELL</p>	Added descriptions for these fields to the Block Help
14844	<p>DM7_30_MSG latch problem: DM7 30 MSG latched values do not persist through power cycles</p>	Modified the UpdateDM30Block function so that the Logged flag is read from the EE and set in RAM for valid latched data
14852	<p>SYS_INFO Block Help incomplete: The Block Help for the SYS INFO block needs to mention that a FLASH_OPT value of 4 is needed for applications with the DATA LOG M block</p>	<p>Added the following to the SYS INFO block help topic: NOTE1: FLASH_OPT "Level 4" must be used when using the DATA LOG M block</p>
14885	<p>MODBUS_S Block Help incomplete: MODBUS S Block Help topic: The help does not specify that a different NET_ADDR must be used for each port, even though they are on the same slave</p>	Changed the MODBUS S Block Help topic to say that in TCP mode, the NET_ADDR_ field value must be unique within the entire GAP application
14887	<p>CAN_READ Block Help confusing The CAN READ Block Help topic has some confusing redundant information about repeat groups</p>	Removed redundant information from the CAN READ Block Help topic
14903	IC SPRK Blocks RAM optimization	Added these blocks to the list of

	<p>error:</p> <p>The IC1100SPRK and IC100 SPRK blocks may or may not function correctly if the SYS_INFO RAM_OPT (ram optimization level) is greater than 1</p>	blocks which are excluded from RAM optimization
14904	<p>PCMHD EE pri fault and EE sec fault problems:</p> <p>Beginning in coder 5.01, these outputs also report errors in the parallel EEPROM. A checksum was added to the parallel EE to verify data validity. On some controls, this checksum may not always get reported correctly and the pri and/or sec fault might be set. This problem may go away after a power cycle</p>	A change was made so that 4 individual EEPROM reads are made in the code to the EEPROM and the checksum is then built and used as a comparison. Timing was verified on the scope to prove that this method provides plenty of margin
<none>	<p>LATCH1 Block Help:</p> <p>Missing description of SEL_ field</p>	Added missing information to Block Help
<none>	<p>AO PWM FLT Block Help:</p> <p>Open and Short Fault information is not accurate</p>	Corrected Block Help
<none>	<p>CAN P STAT Block Help:</p> <p>Need information about redundant CPU operation</p>	Added missing information to Block Help

UNSUPPORTED BLOCKS:

The following is a list of blocks which are available in the Template and GAP Editor, but are not supported in the 5.02-0 release version of the coder (compiler):

- [CAL ID](#)
- [CO IC1100](#)
- [COIL CNFG](#)
- [DM23 MSG](#)
- [DM27 MSG](#)
- [DM28 MSG](#)
- [DM29 MSG](#)
- [HDDIO](#)
- [I TO NAN](#)

- [J1939 TJET](#)
- [PC104 ATL](#)
- [RATIO LIM2](#)
- [TEMP HIST](#)

- Tunable inputs for the [SYNC INJ](#) block are only available to Woodward developers in this version

PART NUMBERS:

Version 5.02-0
 Gap/Coder Part Number: 9927-1792
 Master Kit: 9927-1333, 8928-1088, 1796-1068

Template Changes in 5.02

NEW BLOCKS

The following blocks were added new to this template:

- [DM23 MSG](#)
- [DM27 MSG](#)
- [DM28 MSG](#)
- [DM29 MSG](#)
- [HDDIO](#)

DELETED BLOCKS

The following blocks were deleted going from the 5.01 template to the 5.02 template:

- CAN_RD_WR

MODIFIED BLOCKS

The following blocks' input and/or output fields were modified going from the 5.01 template to the 5.02 template

<u>Block Name</u>	<u>Description of Template Changes</u>
AO 4 20	B_ENABLE and SD fields were made compliant with on-line change requirements (not available in many coder versions)
DM MANGR	Added DM1_2_MIL field. Added support for new DM messages (23,27,28,29)
LL VALUES	The variable allocation count fields (LL_?_CNT) were made compliant with on-line change requirements (not available in many coder versions)

<u>SYNC INJ</u>	The MIN_ADV, MAX_ADV and CYL fields were changed to allow variable parameters (not available in many coder versions)
<u>MIL</u>	The FLASH_MODE field was changed to default to TRUE instead of FALSE
<u>PCM_STATUS</u>	Added AUTO_SAVE field to allow (by disabling AUTO_SAVE) an application controlled strategy to reduce the number of saves to the EEprom
<u>EVENT MGR</u>	Added MIL_STAT Boolean output field Added TIME_DISP, MIL_CNTDWN and MIL_TIME Integer input fields
<u>DM25 MSG</u>	Added TRIG_TYPE constant Integer input field to allow freeze frame data to be triggered by PENDING as well as ACTIVE events Changed RPTdtc repeat group to allow up to 400 ID_ fields (from 90)