

New South Wales

**Gaming Machine
Communications Protocol
Technical Standard**

Revision 2.4

5 August 2013

Table of Contents

1 New South Wales Gaming Machine Communications Protocol	5
1.1 General	5
Introduction	5
Legislation	5
Supporting Regulations	5
NSW Technical Standards	5
The Casino Liquor and Gaming Control Authority	6
The Office of Liquor, Gaming and Racing	6
Dispensations	6
2 Data Interface Specification	7
2.1 Introduction	7
Subsidiary Equipment to be connected	7
2.2 General Requirements	7
3 Data Block Static and Control Data	9
3.1 Data Block Static Data	9
GMID (Gaming Machine Identification Number)	9
Data Block Identifier	10
Data Block Version Number	10
Base Credit Value	11
Firmware Identification (Program Identification)	11
Percentage Return To Player (Theoretical/Estimated PRTP)	13
Non-SFS Gaming Machine With Theoretical/Estimated PRTP Under 85%	13
Multi-Game Indication (Game Number)	14
Multi-Game Combination Number	14
Gaming Machine Secondary Functions Supported	14
Progressive Levels Supported (Standard Linked Progressive Levels Supported)	15
Probability of Progressive Winning Event	15
Standalone Progressive Levels Supported	15
Set Increment Percentage (Level 1 – 4)	16
Reset Value (Level 1 – 4)	16
Jackpot Limit (Level 1 – 4)	16
Total Standalone Progressive Percentage Return To Player	17
Manufacturer Identification	17
3.2 Data Block Control Data	17
Block Sequence Counter	17
Checksum	18
4 Metering	19
4.1 List of Soft Meters (Electronic Meters)	19
Definition of Soft Meters	20
Turnover	20
Total Wins	20
Cashbox	20
Cancelled Credits	21

Money In	21
Money Out	21
Cash In	22
Cash Out	22
Miscellaneous Accrual	22
Credit	22
Occurrence Meters	23
Power Up	23
Games Played Since Last Power Up	23
Games Played Since Last Door Open	23
Games Played (Stroke)	23
Current Jackpot Value (Level 1 – 4)	24
Accrued Hidden Jackpot Value (Level 1 – 4)	24
Number of Jackpot Resets (Level 1 – 4)	25
Total Value of Jackpots Won (Level 1 – 4)	25
Total Turnover Accumulated by iSAP-GM Since Start Up	25
Total Turnover Accumulated by iSAP-GM Since Last Configuration Change	26
5 Gaming Machine Status Indicators	27
5.1 Machine Status Indicators	27
Non Lockup Status Definitions	28
Idle Mode	28
Game Cycle (Game Play Mode)	29
Power Up	29
Reset	29
CCCE Transfer Completed	30
Normal	30
Jackpot Hit	30
Standalone Progressive Configuration Change	31
Lockup Status – Non Error	31
Large Win	31
Collect Cash	31
Cancel Credit	32
Progressive Win	32
Manufacturer or Game Specific Win	32
Lockup Status – Error/Audit/Test/Power Save	33
Test Mode	33
Audit Mode	33
Main Door(s) Open	33
Security Cage(s) Open	33
Memory Error	34
Self Audit Error	34
Mechanical Meter Disconnect	35
Display Error	35
Cash Input Error	35
Cash Output Error	36
Manufacturer Specific Error	36
Gaming Machine Power Save	36
Subsidiary Equipment Play Suspended	36

Cancel Credit Error	38
6 Communication Ports	39
6.1 Communication ports to be provided	39
6.2 Port Overview	39
Port P1	39
Ports P2, P3, P4, P5, P6	40
General	40
6.3 Port Details. Port P1	41
Hardware	41
CCCE Protocol (Increment/Decrement In Credits)	42
Overview	42
CCCE Transaction Set 1 (T1) (Increment In Credits)	44
CCCE Transaction Set 2 (T2) (Decrement In Credits)	47
6.4 Port Details. Ports P2, P3, P4, P5, P6	49
Hardware	49
Power Good Signal	50
Subsidiary Equipment Function (SEF) Signal	51
6.5 Data Transmission	51
Overview	51
Standard Data Block (SDB)	52
Function Data Block (FDB)	57
Progressive Data Block 1 (PDB1) and Progressive Data Block 2 (PDB2) Pair	60
Manufacturer Data Block (MDB)	69
6.6 Extended CCCE Functions	70
CCCE Command Classes Overview	70
Gaming Machine Status Requirements for CCCE Cents Transfer and Progressive Win CCCE Transfer	72
Other requirements	75
CCCE Command Classes	77
6.7 Machine Lockup Command Set 50H-52H	90
6.8 Message Display Command Set 53H-54H	96
6.9 Program Signature Request Command Set 56H-57H	101
6.10 Non-Cash Prize Ticket Print Command Set 59H-5AH	104
6.11 Peripheral Manufacturer Data Block Version 1.06	107

1 New South Wales Gaming Machine Communications Protocol

1.1 General

Introduction

- 1.1.1 This New South Wales (NSW) Gaming Machine Communications Protocol Technical Standard defines the communications requirements that exist between gaming machines, the Centralised Monitoring System (CMS) and subsidiary equipment.
- 1.1.2 Refer to the Communications Protocol Version 2.4 “Redline” document for details of significant changes between this document and the Communications Protocol Version 2.3.

Legislation

- 1.1.3 The requirements specified in this document are supplementary to and do not take the place of any of the requirements of the *Gaming Machines Act* 2001 (referred to as “the Act”).

Supporting Regulations

- 1.1.4 The requirements specified in this document are supplementary to and do not take the place of any of the requirements of the *Gaming Machines Regulation* (referred to as “the Regulations”).

NSW Technical Standards

- 1.1.5 The following technical standards are used in NSW (see the Department’s website for the latest revisions):
 - a) NSW Appendix to the National Standard;
 - b) NSW Gaming Machine Communications Protocol;
 - c) NSW Jackpot Technical Standard; and
 - d) Ticket In – Ticket Out (TITO) technical standard.

The Independent Liquor and Gaming Authority

- 1.1.6 The Authority has the responsibility of approving gaming equipment and technical standards for gaming equipment to be used for the purposes of gaming in Clubs and Hotels in NSW. State Wide Link jackpot equipment is approved by the Minister.

The Office of Liquor, Gaming and Racing

- 1.1.7 The Office of Liquor, Gaming and Racing (OLGR) has the responsibility of administering the Authority's approval process. Contact the OLGR's Gaming Technology Unit for information regarding technical standards, and submissions for approval of gaming equipment.
- 1.1.8 Matters arising from the testing of gaming equipment which have not been addressed in this document (e.g. due to omissions or new technology) will be resolved by the Authority and its delegates as part of the approval process.
- 1.1.9 Gaming equipment, which is outside the scope of this Communications Protocol standard, may be submitted for evaluation and approval. However approval is at the discretion of the Authority which will assess each application on its own merits. Note that the Gaming Technology Unit should be advised prior to the lodgement of the application, and that 'field trial' approval from the Authority can be sought.

Dispensations

- 1.1.10 In cases where new gaming equipment does not meet the technical standards due to new or advanced technology, there may be reason to seek a dispensation. The manufacturer is advised to seek approval for any dispensation early in the development cycle.

Bug-fix submissions, the addition of TITO functionality, (and in some cases) games being migrated to another approved platform, may comply with the technical standard against which the equipment was originally approved, or a later technical standard as authorised by the Authority.

2 Data Interface Specification

2.1 Introduction

This document describes the data communication facilities to be provided by a gaming machine.

Subsidiary Equipment to be connected

2.1.1 The Data Interface Specification will allow communication between a gaming machine and the following classes of subsidiary equipment:

1. Data Gathering or Monitoring Systems
2. Centralised Cash Control Equipment (CCCE)
3. Promotional Equipment
4. Progressive Equipment ("standard" or "mystery" and "linked").

A minimum of six ports are to be provided for connection of subsidiary equipment.

2.2 General Requirements

2.2.1 The gaming machine is a stand-alone device responsible for all aspects of its own functioning.

The gaming machine's secure area (logic areas) may only connect and communicate to devices 'outside' the gaming machine via the six dedicated NSW communication ports (P1 to P6) and only in a manner as described in this document and any OLGR approved technical standards (which include the GTA Ticket Printer, MDB, and TITO standard). 'Outside' devices generally means the CMS-GMIC, player loyalty interface cards, card-based gaming interface cards, CCCE interfaces, jackpot controller interface cards, and in-venue monitoring system interface cards. These devices are located inside the machine but communicate with devices outside the gaming machine.

Video inputs containing 'non-sensitive' data (from sources other than the gaming machine) are allowed to be connected in an approved manner to a gaming machine. For example, a connection of a video stream to enable a jackpot display on the machine's top box. However, the input data stream must not communicate with the machine's secure logic area in any manner.

In certain circumstances (i.e. TITO conversion of older games not capable of supporting a single copy ticket printer) it may be acceptable to use a system interface card to intercept signals between the machine's bill validator unit and the logic area, and a printer. This will be considered for approval on a case by case basis.

Downloadable game functionality will be considered for approval on a case by case basis, on the proviso that:

- a) A business model for the selling of the proposed system and any ongoing licensing type issues has been approved/authorised by the Authority and/or the OLGR;
- b) The proposed system is capable of integration and operation with the CMS business rules (rules pertaining to monitoring, venue verification and authorisation); and
- c) The proposed system is capable of being approved and being able to operate within the confines of the NSW gaming legislation.

- 2.2.2 The standard is to be used for point to point communication between the gaming machine and a subsidiary equipment "interface" that must be located within the secure enclosure of the gaming machine.
- 2.2.3 Only equipment that is evaluated and approved by the Authority may be connected to a port. The equipment connected must correspond to the intended port usage.
- 2.2.4 The data block version must remain fixed for the gaming machine. A change in the data block version number will require a new version of the game software to be created.
- 2.2.5 Gaming machine static/configuration data can only be changed after a gaming machine RAM reset (or cause a RAM Reset to occur).
- 2.2.6 Change of any gaming machine software should only be performed after a gaming RAM reset procedure. If the gaming machine software is changed without a machine RAM reset being performed, the gaming machine will report a memory error as a result of failed static data consistency verification.
- 2.2.7 Once a RAM reset is executed, the gaming machine must terminate any data block output until the memory error and the self-audit error are completely cleared and the machine is completely configured. A gaming machine is configured when it (the machine) is ready to accurately output all relevant data block information (reflecting the identification, the metering and the current state of the machine).
- 2.2.7a On configuration change, a gaming machine is to stop sending data blocks until after the RAM Reset and the machine is completely configured.
- 2.2.8 Any value entered during the machine configuration procedure must be confirmed before being accepted by the gaming machine or the setup of that value is aborted.
- 2.2.9 It is preferable to restore a previously set GMID from an EEPROM (or equivalent) upon a gaming machine RAM reset.
- 2.2.10 All gaming machine configuration data must reset to zero after the RAM reset, except for the default value.
- 2.2.11 All gaming machine meters must be reset to zero when the RAM reset procedure is executed.
- 2.2.12 The first standard data block (SDB) output after the gaming machine RAM reset / Data Configuration Procedure must have all meters set to zero. One SDB with such meter value must be output.

3 Data Block Static and Control Data

3.1 Data Block Static Data

Gaming machine "static" data, which is reported in the data blocks such as SDB, FDB, PDB1, PDB2 and MDB, is defined as:

GMID (Machine Number)
Data Block Identifier
Data Block Version Number
Base Credit Value
Firmware Identification (Program Identification)
Percentage Return to Player (It is static for single games only.)
Multi-Game Indication (Game Number – It is static for single games only.)
Multi-Game Combination Number
Gaming Machines Secondary Functions Supported
Progressive Levels supported (Standard Linked Progressive Levels supported)
Probability of a Progressive Winning Event
Standalone Progressive Levels Supported
Set Increment Percentage (Level 1 – 4)
Reset Value (Level 1 – 4)
Jackpot Limit (Level 1 – 4)
Total Theoretical Standalone Progressive Percentage Return to Player
Manufacturer Identification (for MDB only)

(Refer to sections below for details)

GMID (Gaming Machine Identification Number)

- 3.1.1 GMID is a six digit number allocated by the Authority to each gaming machine. The technique used to set this number must bear in mind the requirement of byte for byte compatibility between the game program and the "master program" held by the Authority as well as the need for flexibility and security to change the GMID if necessary. It will not be permissible to store this parameter in any game program storage devices as this will violate the byte for byte compatibility requirement. For example, an acceptable technique is that the GMID be stored on "switches" or in EEPROM, which can be independently verified. To set or verify the GMID, access must be gained to a sealed logic cage.

Prior to commissioning, written advice stating the GMID number must be received from the Authority.

A gaming machine is not permitted to operate without a valid GMID number.

Note: Zero GMID or non-BCD coded GMID are not valid GMIDs.

This clause replaces section 3.18.7 of the National Standard.

Data Block Identifier

- 3.1.2 The Data Block Identifier is a two digit code that is transmitted in byte 2 of all data blocks. The Data Block Identifier permits any subsidiary equipment that is attached to a gaming machine, via communication ports P1 to P6, to identify the type of data block being transmitted, thus facilitating the interpretation of data being received.

The following Data Block Identifiers are valid for transmission by the gaming machine in its output data blocks:

<u>Data Block Identifier</u>	<u>Data Block Type</u>
00	Standard Data Block
01	Function Data Block
10	Progressive Data Block 1
11	Progressive Data Block 2
22	Manufacturer Data Block
A1h	CCCE Increment in Credits, Phase 1 Echo
A3h	CCCE Increment in Cents, Phase 1 Echo
B1h	CCCE Decrement in Credits, Phase 1 Echo
B3h	CCCE Decrement in Cents, Phase 1 Echo
80h – 8Fh	Progressive Win Notification with CCCE Pay, Echo
90h – 9Fh	Progressive Win Notification with Hand Pay, Echo
C0h – CFh	Standard Progressive Win Payment, Phase 1 Echo
D0h – DFh	Mystery Progressive Win Payment, Phase 1 Echo
70h, 71h	Ticket In Information (TII), TII Phase 1 Echo
51h, 52h	Machine Lockup Command Echo, Player Acknowledgement Receipt
54h	Message Display Command Echo
56h, 57h	Program Signature Request Command Echo, Program Signature Result
5Ah	Non-Cash Prize Ticket Print Command Echo

Data Block Version Number

- 3.1.3 This is a four (4) digit number permanently stored with the gaming machine software program e.g. stored in program EPROM. Each data block (i.e. SDB, FDB etc.) will have a separate data block version number. A data block version number will be altered whenever a revision is made to the data block, i.e. a new version is introduced.

The storage locations and the technique used to store the information must be consistent for all gaming machines (pertaining to a particular family of platforms) produced by the manufacturer. The stored information must be verifiable using standard commercially available techniques.

Base Credit Value

- 3.1.4 For a single-denomination gaming machine, the base credit value that is reported in the Standard Data Block, is a two-byte Binary Coded Decimal code (BCD) representing the monetary value attached to each credit and expressed in cents.

The base credit value may only have one of the following values: 1c, 2c, 4c, 5c, 10c, 20c, 25c, 50c, \$1 and \$2.

- 3.1.4a For a multi-denomination gaming machine, the base credit value that is reported in the Standard Data Block, is a two-byte Binary Coded Decimal code (BCD) representing one cent at all times. Any player's selected denomination is not to be transmitted in the Standard Data Block.

Each multi-denomination combination that a multi-denomination game may support, must consist of two or more different denominations with respect to 1c, 2c, 4c, 5c, 10c, 20c, 25c, 50c, \$1 and \$2, and the combination must not be prohibited by the 'Gaming Machine Prohibited Features Register'.

Note: A multi-denomination gaming machine is not to support any symbol-driven progressive feature/prize of which the winning probability (or the hit rate) will vary as different denominations are selected by players.

A Program ID Combination (PIDC) is permitted to support either the multi- or single-denomination, but not both. If a single software chipset contains configurable multi- and single-denomination functions, then depending on whether a multi- or single-denomination is being configured for operation, an appropriate (but different) PIDC must be transmitted in the SDB by the gaming machine.

Firmware Identification (Program Identification)

- 3.1.5 All firmware defining the gaming machine must be identified.

The complete set of the program and data information that is stored in the gaming machine and is related to the security of the gaming machine needs to be identified.

There are four (4) program identifiers allocated for this purpose and the manufacturer must ensure that all of the gaming machine firmware can be identified using these four identifiers.

Each program identifier will have a unique eight character identifier allocated by the manufacturer. The identifier is used to identify the gaming machine program or data that can be executed or accessed by the processor or any other fixed data that is accessed by the processor either directly or indirectly, and is related to security of the gaming machine.

Note that each binary file submitted to the Authority for approval must be byte for byte identical with the corresponding software / firmware component installed in the field and it must contain the unique 8 character identifier stored as an ASCII string.

Each program identifier will be left justified and the unused characters will contain 20H (spaces). For example, a program identifier of 301221A representing a Game specific

program is interpreted as 301221A(spaces) and therefore transmitted as “2041 3132 3231 3033” with respect to SDB bytes 88 to 95.

The storage locations (e.g. memory addresses) must be consistent in all gaming machines (pertaining to a particular family of platforms) produced by the manufacturer.

Note: Reference section 2.4.8 of the National Standard.

3.1.5a A “unique” Program ID Combination (PIDC), which is the concatenation of the four program identifiers, must be reported in the SDB, SDB Bytes 88 – 119 for all “executable” program components. If more than one program identifier is used, then the following reporting order must be followed:

- Game specific / application software, SDB bytes 88 – 95.
- Game shell / game base / game kernel / system software, SDB bytes 96 – 103.
- Communication / IO software, SDB bytes 104 – 111.
- Miscellaneous / sound / graphics etc. software, SDB bytes 112 – 119.

Note: If a single software chipset may be configured to operate as a standalone non-progressive game or a standalone progressive game, then depending on whether a standalone non-progressive or progressive game is being configured for operation, an appropriate (but different) PIDC must be transmitted in the SDB by the gaming machine to satisfy with the CMS business rules.

A software chipset that is capable of operating as a standard linked progressive game, must not permit any other machine configurations (to operate as a standalone non-progressive game or a standalone progressive game).

Manufacturers should note that for machine designs using a single software chipset which permits multiple machine configurations, should any components of the software chipset have a problem, then all machines with that chipset will need to be retrofitted in the field.

3.1.5b Multi Terminal Gaming Machine (MTGM) with distributed main station (central unit) and terminal software implementation must include both main station and terminal software program identifiers in the PIDC reported in the SDBs.

Main station software shall be reported as system software in SDB bytes 96 – 103. Terminal software shall be reported as game specific software in SDB bytes 88 – 95.

3.1.5c The PIDC (program identifiers 1 – 4) reported by the integrated standalone progressive gaming machine in the PDB1 bytes 72 – 103, shall “mirror” the PIDC reported in the SDB bytes 88 – 119.

Percentage Return To Player (Theoretical/Estimated PRTP)

- 3.1.6 The theoretical/estimated percentage return that is used for administrative purposes to identify the game in the electronic communication (SDB) output by the gaming machine is the minimum PRTP.

Games that implement a feature that introduces an element of choice by the player, that impacts upon the PRTP of the game, must provide an “auto hold” facility, or alternatively “strategy advice” to the player. The PRTP reported in the SDB must be the PRTP calculated using the “auto hold” strategy or “strategy advice”, which in this case is considered to be the minimum PRTP. MTGM ‘table’ games are exempt.

During the Game Cycle the PRTP of the game being played is to be reported in the SDB, otherwise the PRTP of the last game selected is to be reported. Note that after a RAM reset when no games been selected, the PRTP of the game with the lowest Game Number is to be reported.

The PRTP reported in the SDB will reflect the theoretical return to player for all prizes that are accounted for by the “Total Wins” meter of the gaming machine. (i.e. Theoretical return for all prizes that are added to the “Total Wins” meter.)

Note: For standard linked progressive games, the sum of the PRTP reported in the SDB output by the gaming machine and the progressive PRTP reported in the PDB1 output by the standard link controller must be 85% or above.

This clause defines additional requirement(s) for sections 3.9.15 and 3.9.16 of the National Standard.

Non-SFS Gaming Machine With Theoretical/Estimated PRTP Under 85%

- 3.1.6a A gaming machine that does not support any secondary function and is configured to operate at a theoretical/estimated (Game) PRTP of less than 85%, is permitted on the proviso that:

- a) It (the above-mentioned gaming machine) is connected to a mystery link controller that supports the extended CCCE command class ‘E’ or command ‘62h’;
- b) The theoretical Game PRTP of the gaming machine together with the theoretical progressive PRTP contribution of the mystery link controller provides a total PRTP of 85% or above; and
- c) It (the above-mentioned gaming machine) must be capable of accepting and processing the extended CCCE command class ‘E’ or ‘command ‘62h’ and use the above-mentioned CCCE command input signals from the mystery link controller as a ‘heartbeat’ monitor to ensure that the machine is connected to the link. If the heartbeat signals are not being detected for more than 10 seconds, the machine must disable itself by not allowing any new game play to commence.

While the machine is in an “idle, play suspended” state, it must display an appropriate on-screen error message to inform the player.

The machine must automatically exit the “idle, play suspended” state and return to a normal playable state when a valid heartbeat signal is received from the mystery link controller (the mystery link controller’s interface card) unless any other lockup condition(s) is/are detected.

Note: A non-SFS gaming machine is referred to a gaming machine that does not support any secondary function and it is not for use with any standard link.

The mystery link controller that supports the extended CCCE command class 'E' and/or command '62h', must ensure that the theoretical Game PRTP of each non-SFS gaming machine connected to it, together with the theoretical mystery link PRTP contribution, provides a total PRTP of 85% or above. Otherwise (if the sum of the PRTP reported in the SDB output by the connected gaming machine and the progressive PRTP reported in the PDB1 output by the mystery link controller is less than 85%), the controller must disable the detected gaming machine from play.

Multi-Game Indication (Game Number)

- 3.1.7 The Game Number must be reported in the SDB byte 15. If there is no multi-game software support and only a single game can be played, the Game Number is set to 00. In a multi-game selection, each game is allocated a Game Number from 1 to 16 and it is reported in the SDB byte 15.

During the Game Cycle and Game Idle Mode, the Game Number (in decimal) of the component game being played is to be reported in the SDB byte 15.

When in the Game Selection Screen, the number of (component) games in the MGGM (i.e. the configured multi-game combination) must be reported with the offset value of "20".

For example, if the number of component games in the configured multi-game combination is 8, then the value (i.e. the number of component games) reported in the SDB byte 15 will be 28 (the sum of 08 and 20). If the number of component games in the configured multi-game combination is 16, then the value reported in the SDB byte 15 will be 36 (the sum of 16 and 20).

In multi-game software installation, the default Game Number must not be set to 00.

Multi-Game Combination Number

- 3.1.8 Multi-Game Combination Number (the identification number of the configured multi-game combination) must be reported in the SDB byte 16. Each multi-game combination must have a unique theoretical return percentage to player for that game specification number. There may be up to 8 multi-game combinations defined for any game specification number.

Gaming Machine Secondary Functions Supported

- 3.1.9 This is a one byte number, the bits of which represent the secondary functions supported by the gaming machine.

Secondary functions are those approved functions supported by the gaming machine that permit the use of other prize generating equipment which is activated by a defined "winning event" occurring within the machine.

Presently the least significant bit (i.e. bit 0 = 1) indicates that the machine needs to be connected to a Standard Linked Progressive Jackpot System and must output FDB. Other bits are reserved for future use and must contain the value zero (0). The byte is to be permanently stored with the game program (e.g. game PROM).

Progressive Levels Supported (Standard Linked Progressive Levels Supported)

3.1.10 This is a one byte number (BCD) indicating the number of progressive levels supported by the gaming machine. A maximum of four levels are permitted.

The number is to be permanently stored with the game program (e.g. game PROM). The following data is to be stored:

No progressive support,	00
1 level	01
2 levels	02
3 levels	03
4 levels	04

The storage location (e.g. memory address) and technique used to store the information must be consistent for all gaming machines (pertaining to a particular family of platforms) produced by the manufacturer. The stored information must be verifiable using standard commercially available techniques.

This is valid for gaming machines having the "progressive bit" enabled in "secondary functions supported" byte (standard links) of the SDB. The number of progressive levels supported will be output in the FDB.

Probability of Progressive Winning Event

3.1.11 Where a gaming machine is designed to operate in conjunction with a "Standard" Linked Progressive Jackpot System, the probability of occurrence of the progressive prize winning event(s) must be stored with the game program and communicated in the FDB.

The probability of progressive winning event for jackpot levels 1, 2, 3 and 4 are correspondingly reported in the FDB bytes 27 – 33, 34 – 40, 41 – 47 and 48 – 54.

Note: The probability of progressive winning event is the unconditional probability of triggering a jackpot winning event for each jackpot level, based upon 1 credit and 1 line bet.

Standalone Progressive Levels Supported

3.1.12 This is a one byte BCD number indicating the number of standalone progressive levels supported by the integrated standalone progressive gaming machine. A maximum of four levels is permitted.

The number is to be permanently stored with the game program (e.g. game PROM). The following data is to be stored:

No progressive support,	00
1 level	01
2 levels	02
3 levels	03
4 levels	04

The storage location (e.g. memory address) and technique used to store the information must be consistent for all integrated standalone progressive gaming machines (pertaining to a particular family of platforms) produced by the manufacturer. The stored information must be verifiable using standard commercially available techniques.

The number of standalone progressive levels supported is to be reported in the PDB1 output by the integrated standalone progressive gaming machine.

Set Increment Percentage (Level 1 – 4)

- 3.1.13 This is the percentage of the turnover of the integrated standalone progressive gaming machine, that is added to the pool of funds available for standalone progressive jackpots. (The set increment percentage is also known as contribution percentage.)

The set increment percentage will have a value for each standalone progressive level supported.

It is to be reported in the PDB1 output by the integrated standalone progressive gaming machine.

Note: The definition of INCREMENT meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Reset Value (Level 1 – 4)

- 3.1.14 This is the minimum increment-free amount, which is added to the standalone progressive prize pool, after a standalone Jackpot reset. A separate reset value is applicable for each standalone progressive level supported.

The reset value for each standalone progressive level is to be reported in the PDB2 output by the integrated standalone progressive gaming machine.

Note: The definition of STARTUP meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Jackpot Limit (Level 1 – 4)

- 3.1.15 The jackpot limit of a standalone progressive level for transmission in the PDB2 by the gaming machine can be up to \$10,000 provided that the maximum standalone progressive prize limit for a game element as listed in section 2.1.4a of the NSW Jackpot Standard is satisfied.

Note: The definition of CEILING meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Total Standalone Progressive Percentage Return To Player

3.1.16 For mystery standalone progressive jackpot machines, the theoretical/estimated standalone progressive percentage return must be calculated using the configuration parameters such as set increment percentage, reset value and jackpot limit that are set in the integrated standalone progressive gaming machine (iSAP-GM).

For standard standalone progressive jackpot machines, the theoretical/estimated standalone progressive percentage return must be calculated using the configuration parameters such as set increment percentage, reset value, probability of progressive winning event and base credit value that are set in the iSAP-GM.

The total theoretical standalone progressive percentage return to player for the standalone progressive jackpots, of which the win amounts are added to the "Total Wins" meter of the iSAP-GM, is to be reported in the PDB1.

Manufacturer Identification

3.1.17 Manufacturer ID is a unique two-character code permanently allocated to each dealer by the Authority. This code is to be stored in the gaming machine program storage device. The number will be stored as two (2) ASCII bytes.

This manufacturer ID is reported in the MDB, MDB byte 3.

3.2 Data Block Control Data

Data block "control" data, which is reported in data blocks such as SDB, FDB, PDB1, PDB2 and MDB, is defined as:

1. Block Sequence Counter (for SDB, FDB, PDB1 and PDB2 only)
2. Checksum

Block Sequence Counter

3.2.1 This counter is incremented only if any data in the data block has been modified when it is compared to the previously transmitted block. i.e. If the same block is being repeated the counter remains unchanged. BCD code is to be used. This will give a range of 0 to 99 which is automatically "wrapped around" to 0.

The data block sequence counter must always increment by one in the same data block if there is any change in the data block.

The counter is initialised to zero whenever a power up or reset takes place.

Checksum

- 3.2.2 In data blocks such as SDB, FDB, PDB1, PDB2 and MDB, of which each has 128 bytes (bytes 1 to 128), the checksum is generated by modulo 2 addition applied to bytes 2 to 126. The checksum is to be split into two bytes each containing one nibble. i.e. 0Xh (checksum lower nibble) and X0h (checksum upper nibble) are respectively reported in bytes 127 and 128. This is required to avoid generating the unique start of block character (FF).

In CCCE commands such as “CCCE Increment In Credits command class A – Phase 1 Echo” and “CCCE Decrement In Credits command class B – Phase 1 Echo” of which each has 10 bytes (bytes 1 to 10), the checksum is generated by modulo 2 addition applied to bytes 2 to 8. The checksum is to be split into two bytes each containing one nibble. i.e. 0Xh (checksum lower nibble) and X0h (checksum upper nibble) are respectively reported in bytes 9 and 10.

4 Metering

4.1 List of Soft Meters (Electronic Meters)

Meters to be provided as “soft” (electronic meters) include:

1. Turnover
2. Total Wins
3. Cashbox
4. Cancelled Credits
5. Money In
6. Money Out
7. Cash In
8. Cash Out
9. Credit
10. Miscellaneous Accrual (if required)
11. Occurrence meter 1
12. Occurrence meter 2
13. Occurrence meter 3
14. Occurrence meter 4
15. Power Up
16. Games Played Since Last Power Up (stroke since last power up)
17. Games Played Since Last Door Open (stroke since last door open)
18. Games Played (Stroke)
19. Current Jackpot Value (Level 1 – 4)
20. Accrued Hidden Jackpot Value (Level 1 – 4)
21. Number of Jackpots Resets (Level 1 – 4)
22. Total Value of All Jackpots Won (Level 1 – 4)
23. Total Turnover Accumulated by iSAP-GM Since Start-Up
24. Total Turnover Accumulated by iSAP-GM Since Last Configuration Change

The soft meters such as “Turnover”, “Total Wins”, “Cashbox”, “Cancelled Credits”, “Money In”, “Money Out”, “Cash In”, “Cash Out”, “Credit”, “Miscellaneous Accrual”, “Current Jackpot Value”, “Accrued Hidden Jackpot Value”, “Total Value of All Jackpots Won”, “Total Turnover Accumulated by iSAP-GM Since Start-Up” and “Total Turnover Accumulated by iSAP-GM Since Last Configuration Change” are to increment in units of one cent (\$0.01).

The soft meters such as “Occurrence meter 1”, “Occurrence meter 2”, “Occurrence meter 3” and “Occurrence meter 4” are only required for gaming machine supporting secondary function (i.e. connected to standard linked progressive jackpot system).

The soft meters such as “Current Jackpot Value (Level 1 – 4)”, “Accrued Hidden Jackpot Value (Level 1 – 4)”, “Number of Jackpots Resets (Level 1 – 4)”, “Total Value of All Jackpots Won (Level 1 – 4)”, “Total Turnover Accumulated by iSAP-GM Since Start-Up” and “Total Turnover Accumulated by iSAP-GM Since Last Configuration Change” are only required for gaming machine supporting integrated standalone progressive jackpots.

Note: "iSAP-GM", standing for integrated Standalone Progressive Gaming Machine, is a standalone gaming machine with integrated standalone progressive controller/functions. Hence, an iSAP-GM must meet all the requirements defined for a gaming machine as well as those defined for integrated standalone progressive controller and functions.

Definition of Soft Meters

Turnover

- 4.1.1 This meter represents the cumulative total of all money bet on the gaming machine. The money bet must be decremented from the credit meter.

Incrementation of the turnover meter should take place when the bet is accepted by the gaming machine and the credit meter is decremented.

Note: The SDB Turnover and Stroke meters must be updated and reported in the same SDB upon commencement of a game play. For games where the rules permit betting of additional credits (i.e. Blackjack insurance, double down and pair splitting), the SDB Turnover meter is allowed to increment as per each additional bet made during the game play while the Stroke meter remains unchanged.

The definition of TURNOVER meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Total Wins

- 4.1.2 This meter represents the cumulative total of all wins generated by the gaming machine and transferred to the credit meter.

Incrementation will take place at the completion of a game play in which a valid win has occurred. Under some circumstances, the meter may be incremented before the completion of the game play e.g. "partial double up" where a portion of the currently displayed win is transferred to the credit meter.

Note: The SDB Total Win and Credit meters must be updated and reported in the same SDB upon confirmation of winnings (i.e. after the final double-up attempt).

The definition of TOTAL WINS meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Cashbox

- 4.1.3 This meter represents the cumulative total of the monetary value of cash, which is actually deposited in any "cash box" associated with the gaming machine. (i.e. coins and notes to cashbox)

Incrementation takes place when cash is detected entering a cashbox.

Note: The definition of CASH BOX meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Cancelled Credits

- 4.1.4 This meter represents the cumulative total of all credits paid out by means of a "book payout" and/or a "ticket". The "ticket" component applies where a ticket printer is integrated into the gaming machine and does not make use of the money transfer facilities available on CCCE port P1.

Incrementation takes place when the "key off" procedure is executed by an authorised person or when the machine deduces that a ticket has been "successfully" dispensed from the device.

Note: The SDB Cancelled Credits and Credit meters must be updated and reported in the same SDB.

The definition of CANCELLED CREDITS meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Money In

- 4.1.5 This meter represents the cumulative total of the value of money transferred into the gaming machine from Centralised Cash Control Equipment (CCCE).

Incrementation takes place at the conclusion of a transaction at the same time the credit meter is incremented and the CCCE is notified of a successful transaction.

Note: The SDB Money In and Credit meters must be updated and reported in the same SDB.

The definitions of MONEY IN and CASHLESS IN meters, as it is stated in section 3.3.16 of the National Standard, are replaced with that defined in this clause.

Money Out

- 4.1.6 This meter represents the cumulative total of the value of money transferred out of the gaming machine to Centralised Cash Control Equipment (CCCE).

Incrementation takes place at the conclusion of a transaction at the same time that the credit meter is decremented and the CCCE is notified of a successful transaction.

Note: The SDB Money Out and Credit meters must be updated and reported in the same SDB.

The definitions of MONEY OUT and CASHLESS OUT meters, as it is stated in section 3.3.16 of the National Standard, are replaced with that defined in this clause.

Cash In

- 4.1.7 This meter represents the cumulative total of the value of all cash (notes and coins) inserted and accepted by the gaming machine.

Incrementation takes place only when a valid cash input is detected.

Note: The definitions of COINS IN and BANKNOTES IN meters, as it is stated in section 3.3.16 of the National Standard, are replaced with that defined in this clause.

Cash Out

- 4.1.8 This meter represents the cumulative total of the value of all coins output from the gaming machine.

Incrementation takes place when a valid coin is output by the gaming machine.

Excess credit paid out is not added to this meter.

Note: The definition of COINS OUT meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Miscellaneous Accrual

- 4.1.9 This meter is intended for a machine that implements special stand alone functions where prize money is accrued "in the background" according to some approved method and is transferred to the credit and win meters when a certain "winning event" occurs.

Credit

- 4.1.10 This meter represents the current accumulated credit balance that can be redeemed by the player. The credit meter is incremented at the same time as the Cash In, Money In or Win meters are incremented.

It is decremented at the same time that the Cash Out, Money Out, Turnover or Cancelled Credit meters are incremented.

Occurrence Meters

- 4.1.11 A gaming machine approved for connection to a "standard progressive system" must be capable of storing four (4) occurrence meters (one for each progressive level).

The "highest" (in terms of maximum prize) level is designated as level 1 and the "lowest" is level 4. Each meter represents the cumulative total of progressive "hits" at each level and will be called Occurrence -1, Occurrence -2, Occurrence -3, and Occurrence -4.

The appropriate occurrence meter is to be incremented at the same time that a valid progressive jackpot win occurs. The "progressive jackpot hit" status condition is turned on and the win level is also set at this time and thus all three pieces of information are available for transmission to the progressive equipment.

Power Up

- 4.1.12 This meter indicates the number of times the machine has been switched on since the last RAM reset.

Incrementation takes place each time the machine is switched on after the power was off.

Games Played Since Last Power Up

- 4.1.13 This meter indicates the number of games played since last power up.

Incrementation takes place each time the machine commences a valid new game play. It is reset to zero whenever a power up occurs.

Games Played Since Last Door Open

- 4.1.14 This meter indicates the number of games played since last door open (i.e. the main door).

Incrementation takes place each time the machine commences a valid new game play. It is reset to zero whenever the main door open occurs.

Games Played (Stroke)

- 4.1.15 This meter represents the cumulative total of games played.

Incrementation takes place each time the gaming machine commences a valid new game play that is not related to the outcome of a previous play. It is not reset to zero like the "games played since last door open" and "games played since last power up" meters.

Note: The SDB Turnover and Stroke meters must be updated and reported in the same SDB upon commencement of a game play. For games where the rules permit betting of additional credits (i.e. Blackjack insurance, double down and pair splitting), the SDB Turnover meter is allowed to increment as per each additional bet made during the game play while the Stroke meter remains unchanged.

The definition of GAMES PLAYED meter, as it is stated in section 3.3.16 of the National Standard, is replaced with that defined in this clause.

Current Jackpot Value (Level 1 – 4)

4.1.16 This meter indicates the standalone progressive “reset” value plus the current accrued standalone progressive Jackpot Value, available to be won as it is accrued since the last standalone jackpot reset (separate value for each level). It is expressed in terms of dollars and cents.

Incrementation takes place when a valid turnover contribution is accepted by the integrated standalone progressive gaming machine. This meter will restart with the "reset" and any "hidden" values after a standalone progressive jackpot reset.

It is to be reported in the PDB1 output by the integrated standalone progressive gaming machine.

Note: The definition of CURRENT VALUE meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Accrued Hidden Jackpot Value (Level 1 – 4)

4.1.17 This meter indicates the value of a standalone progressive jackpot accruing concurrently with the current standalone progressive jackpot or accruing after the current standalone progressive jackpot reaches the prescribed limit, but unavailable to the player until the current jackpot has been won, at which time all or a preset portion of the hidden jackpot is transferred to the new current standalone progressive jackpot. Separate meter is required for each level. It is expressed in terms of dollars and cents.

It is to be reported in the PDB2 output by the integrated standalone progressive gaming machine.

Note: The definition of OVERFLOW meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Number of Jackpot Resets (Level 1 – 4)

- 4.1.18 This is a cumulative total representing the number of times the Standalone Progressive Jackpot has been reset. Separate total is required for each level.

Incrementation takes place when a valid Standalone Progressive Jackpot hit is reset.

It is to be reported in the PDB2 output by the integrated standalone progressive gaming machine.

Note: The definition of HITS meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Total Value of Jackpots Won (Level 1 – 4)

- 4.1.19 This is a cumulative total representing the value of Standalone Progressive Jackpots won. Separate total is required for each level. It is expressed in terms of dollars and cents.

Incrementation takes place once a valid standalone progressive jackpot is hit and the reset is completed.

It is to be reported in the PDB2 output by the integrated standalone progressive gaming machine.

Note: The definition of WINS meter, as it is stated in section 3.3.20 of the National Standard, is replaced with that defined in this clause.

Total Turnover Accumulated by iSAP-GM Since Start Up

- 4.1.20 This is a cumulative total representing the value of turnover accumulated by the integrated standalone progressive gaming machine (iSAP-GM) since its start up. It is expressed in terms of dollars and cents.

Incrementation takes place when a valid turnover contribution to the standalone jackpot pool [i.e. current/hidden standalone jackpot value (level 1 – 4)] is accepted by the iSAP-GM.

It is to be reported in the PDB2 output by the iSAP-GM.

Total Turnover Accumulated by iSAP-GM Since Last Configuration Change

- 4.1.21 This is a cumulative total representing the value of turnover accumulated by the integrated standalone progressive gaming machine (iSAP-GM) since the last (standalone progressive) configuration change. It is expressed in terms of dollars and cents.

Incrementation takes place when a valid turnover contribution to the standalone jackpot pool [i.e. current/hidden standalone jackpot value (level 1 – 4)] is accepted by the iSAP-GM.

It is to be reported in the PDB2 output by the iSAP-GM.

5 Gaming Machine Status Indicators

5.1 Machine Status Indicators

Non Lockup Status

1. Idle
2. Game Cycle
3. Power Up
4. Reset
5. CCCE Transfer Completed
6. Normal
7. Jackpot Hit
8. Standalone Progressive Configuration Change

Lockup Status - Non Error

1. Non Progressive Win or "Large" Win or "Large Accumulated Credit" (optional)
2. Collect Cash
3. Cancel Credit
4. Progressive Win
5. Manufacturer or Game Specific Win

Lockup Status - Error/Audit/Test/Power Save

1. Test Mode
2. Audit Mode
3. Main Door(s) Open
4. Security Cage(s) Open
5. Memory Error
6. Self Audit Error
7. Meter Disconnect
8. Display Error
9. Cash Input Error
10. Cash Output Error
11. Manufacturer Specific Errors
12. Gaming Machine Power Save
13. Subsidiary Equipment Play Suspended
14. Cancel Credit Error

In addition to the above which are communicated through the SDB/FDB/PDB1/PDB2 data blocks, a separate indicator "Gaming Machine Power Good" (output on pins 6 & 7 of each port) will be available.

The machine status bits in the SDB must, as far as reasonably possible, reflect the actual status of the machine at the time the block is being sent. The metering information contained in the SDB/FDB/PDB1/PDB2 data blocks must also adhere to this principle and should represent the updated values corresponding to any actions

performed as well as being consistent with the electronic metering information displayed on the audit screen.

As an example, if the cancel credit status bit is set, that is, the machine is awaiting a cancel credit (book pay) operation and then the cancel credit is actually performed, the following SDB will have the status bit cleared along with updated Cancelled Credit and Credit meters.

As another example, the metering information transmitted in the SDB and displayed on audit screen must pass the "self audit test" at all times unless the gaming machine is actually in a self audit error state, that is, the Self Audit status bit is set. This requirement may mean that the manufacturer needs to employ some form of "lockout or filtering " so that any intermediate meter update activity cannot potentially result in a "self audit error" condition being diagnosed by equipment or persons when this condition does not apply or has not actually been detected by the gaming machine.

Note that in some cases, a status indicator can cover a number of conditions e.g. cash error covers various types of coin faults such as coin in jam, yo-yo etc.

The exact nature of the error(s) is provided by the gaming machine display when the error condition is detected.

The status conditions relating to integrated standalone progressive jackpot functions such as Normal and Jackpot Hit, must always be reported in the PDB1 output by an iSAP-GM. The "Manufacturer Specific Error" indicator reported in the PDB1 byte 13 bits #4 and #5 are to be filled with '0' if not used. Other status bits of the PDB1 bytes 11, 12 and 13 are to be filled with "0" if not used. (The gaming machine status conditions such as "Reset", "Power Up", "Audit Mode", "Meter Disconnect", "Security Cage Open", "Memory Error", "Power Save" and "Test Mode" are not required to be reported in the PDB1 as they are always reported in the SDB.) The PDB1 byte 12 bits #0, #2 and #3 are not applicable to any iSAP-GM.

Non Lockup Status Definitions

Idle Mode

- 5.1.1 This is the normal non playing mode i.e. game playing mode is completed and the gaming machine is waiting for commencement of next game.

This status indicator is set when the machine enters idle mode and cleared when the machine commences a valid game play mode (i.e. game play).

This status indicator is reported on the communications ports.

Every time the gaming machine is in the idle mode, it must output at least one SDB reporting the idle status of the machine (i.e. the SDB idle flag being set to 1 and SDB Game Cycle flag set to 0).

Note: The text "game cycle" is also used to refer to the game play mode.

This clause defines additional requirement(s) for section 3.9.8 of the National Standard.

Game Cycle (Game Play Mode)

- 5.1.2 This is the active game cycle that commences with the press of a button and then a bet (decrement of credit meter by the amount bet) and concludes when all transactions for the game have been completed. A valid game cycle may not commence until a self audit check is performed, mechanical meters connect status is checked and all ports are checked for a "subsidiary equipment function" signal fail condition.

This status indicator is set when a valid game cycle commences and cleared when the idle mode is entered. Idle and Game Cycle status indicators should never be the same.

This status indicator is reported on the communications ports.

Note: The above-mentioned game cycle refers to the game play (mode) that commences with wagering of a bet and concludes when all transactions for the game have been completed. This term has different meaning to that defined in the glossary of the National Standard.

In the NS glossary, the term "game cycle" refers to the total number of possible outcomes of a game.

This clause defines additional requirement(s) for sections 3.9.6 and 3.9.7 of the National Standard.

Power Up

- 5.1.3 The power up status condition is set when power up is detected and is cleared when the first valid new game play after the power up is commenced.

This status indicator is reported on the communications ports and the display.

Note: This clause defines additional requirement(s) for section 3.4.1 of the National Standard.

Reset

- 5.1.4 The purpose of reset is to indicate any abnormal conditions which result in the program being "restarted" and as a result of this process cause significant game state or gaming machine parameters to be partially or completely lost.

The following events summarise typical actions that would lead to the reset status condition being set:

1. The processor(s) is "reset" by a hardware action. Normally this will mean that a "reset" line is activated e.g. a "watchdog timer" causes an incorrectly operating program to restart by means of the "reset" line.
2. The program performs a "cold start or initialisation procedure" and the significant current game state parameters are partially or completely lost.
3. Memory is cleared or it is restored from known "uncorrupted" values.

The reset status condition is set when a reset is executed and is cleared at the commencement of the first valid game play after the reset.

This status condition is reported on the communications ports and the display.

CCCE Transfer Completed

- 5.1.5 This status indicator is set when a valid transfer has taken place between the CCCE and the gaming machine. A valid transfer will mean that the gaming machine Credit and the Money In or Money Out meters have been adjusted appropriately.

This status condition is cleared at the commencement of the first valid game play after the transfer. In cases where more than one valid transfer has taken place before a game play commences, the status condition will remain set until a game play commences.

The display must indicate details of the last valid transfer transaction with the CCCE.

This status condition is reported on communications ports and on the display.

Normal

- 5.1.6 This is the normal function mode i.e. turnover inputs are accepted from the integrated standalone progressive gaming machine, the integrated standalone progressive prize pool is incremented as a result of the turnover inputs and the awarding of the integrated standalone progressive prizes is not impeded.

This status condition is turned on when the above conditions apply and is turned off by conditions which prevent the integrated standalone progressive gaming machine from operating normally e.g. "memory error", "security cage open", "power save" and "test mode", "audit mode", "main door open" status and detection of abnormal conditions.

If a normal status condition does not apply, the integrated standalone progressive gaming machine must suspend its "game play" operation.

This status condition is reported in the PDB1 (PDB1 Byte 11 bit #0) output by the integrated standalone progressive gaming machine.

Jackpot Hit

- 5.1.7 The integrated standalone progressive gaming machine has detected a valid jackpot win in any of the standalone progressive levels supported.

The status indicator is set when the standalone progressive jackpot is hit and is cleared once the jackpot is reset.

If more than one integrated standalone progressive jackpot is waiting to be reset (multiple hits), the status bit remains set until all the standalone jackpots are reset.

This status condition is reported in the PDB1 (PDB1 Byte 11 bit #1) output by the integrated standalone progressive gaming machine.

Standalone Progressive Configuration Change

- 5.1.8 Standalone Progressive configuration change may only occur after a RAM Reset (or cause a RAM Reset to occur).

Lockup Status – Non Error

Large Win

- 5.1.9 Manufacturers have the option to implement the lockup of a machine when a large win greater than or equal to [LARGEWIN] occurs.

If implemented, the status bit is set when the machine detects a large win being equal to or greater than [LARGEWIN]. It is cleared when the appropriate "key-off" procedure is executed.

If not implemented, the status bit shall be zero at all times.

This status condition is reported on the communications ports and on the display.

Note: The value of [LARGEWIN] is not specified at the present.

This clause defines additional requirement(s) for section 3.3.6 of the National Standard. The requirement for Substantial Win, as it is stated in section 3.16.7 of the National Standard, is replaced with that defined in this clause and clause 2.4.63 of the NSW Appendix to the National Standard.

Collect Cash

- 5.1.10 Refers to a situation where the player elects to redeem her/his accumulated credits, which results in the payment being made by means of a hopper payout.

The status condition is turned on when the collect cash process is activated (hopper pay) and turned off when the process is completed.

This status condition is reported on the communication ports and display.

Note: This clause defines additional requirement(s) for section 3.7.4 of the National Standard.

Cancel Credit

- 5.1.11 Refers to a situation where the player elects to redeem her/his accumulated credits, which results in the payment being made by means of a cancel credit payment (book pay) or by way of a "ticket" being issued by the gaming machine.

The status condition is turned on when the gaming machine enters the cancel credit mode and turned off when the appropriate "key off" procedure is executed, a valid ticket has been printed and issued, or the cancel credit mode is cancelled by a player.

This status condition is reported on the communication ports and display.

Note: This clause defines additional requirement(s) for section 3.7.2 of the National Standard.

Progressive Win

- 5.1.12 This is valid for gaming machines having the "progressive bit" enabled in the "secondary functions supported" byte (standard linked progressive function) of the SDB.

This status condition is turned on when the gaming machine detects a valid standard linked progressive win.

It is turned off when one of the following conditions is satisfied:

- a) The gaming machine detects that a valid manual Jackpot Reset key-switch procedure is executed; or
- b) The gaming machine successfully accepts and processes a CCCE progressive win payment transfer utilising the CCCE commands A1/A2 while the machine is in a "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" state or the CCCE command class C while the machine is in a "game play, standard linked progressive win" state.

Note that the progressive win level and occurrence meter are also displayed and output from the communication ports by means of the FDB.

Note: This status indicator causes a "lockup" condition but is not the result of any error or audit condition.

Manufacturer or Game Specific Win

- 5.1.13 This status condition will reflect any other type of "win" that causes the gaming machine to lockup.

The status condition is turned on when the win is detected and is turned off when the appropriate "key off" procedure is executed.

This status condition is reported on the display and the communications ports.

Lockup Status – Error/Audit/Test/Power Save

Test Mode

- 5.1.14 The test mode status condition is turned on whenever the gaming machine enters the test mode and is turned off when this mode is exited.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for section 3.4.2 of the National Standard.

Audit Mode

- 5.1.15 The Audit Mode status condition is turned on whenever the gaming machine enters the audit mode and is turned off when this mode is exited.

This status condition is reported on the display and the communications ports.

Main Door(s) Open

- 5.1.16 Main door open refers to any monitored external door being opened.

This status condition is turned on when a door open is detected and is turned off when a "door closed" status is established.

In the case of main door open sensors disagreeing on the condition detected, the main door open status condition will be turned on, however the display will indicate a "detector mismatch" condition.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for section 3.5.1 of the National Standard.

Security Cage(s) Open

- 5.1.17 This refers to the logic cage and/or cage enclosing the hard meters being opened.

The status condition is turned on when the open condition of the security cage is detected and is turned off when a closed condition of the cage is detected.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for logic area open as it is stated in section 3.16.6 of the National Standard.

Memory Error

5.1.18 In addition to the detection of corrupted memory, the gaming machine must also implement a technique to detect a change in game program(s). When detected, a change will have the same status as a memory error and result in a RAM reset having to be performed. The gaming machine program must have a mechanism to help ensure that any "option" settings that apply with the new program are correctly set.

This status condition is turned on when a memory error is detected and is turned off when a memory reset procedure is performed by an authorised person.

This status condition is reported on the display and the communications ports.

Note that if the memory error bit is set (memory error is detected), only Standard Data Blocks (SDB) are output. No other data block is output (including MDB).

Note: This clause defines additional requirement(s) for section 3.2.6 of the National Standard.

Self Audit Error

5.1.19 This refers to a situation where the electronic meters fail the "self audit check" relationship, which is described as:

$$\text{CREDIT} = [\text{CASH IN} + \text{MONEY IN} + \text{TOTAL WINS} - \text{CASH OUT} \\ - \text{MONEY OUT} - \text{CANCELLED CREDIT} - \text{TURNOVER}] \% 2^{32}$$

Where: % is the modulus operator (to handle meter roll over).

The self audit check must be performed at the following times:

1. The start of every game cycle (a bet cannot be accepted and the game cycle cannot commence unless the self audit is correct).
2. Before commencing any process that involves transferring money to or from the gaming machine i.e. cancel credit, collect cash, transfer of money to or from CCCE.

In the case of coin input, it will not be necessary to perform a self audit check after each coin is inserted, however the gaming machine must perform a self audit at intervals not longer than 15 seconds in the idle no-lockup mode.

The status condition is turned on when the gaming machine detects a self audit error and is turned off after a memory reset is performed.

This status condition is reported on the display and the communications ports.

Note: Gaming machine meters must be updated in such way to ensure that any output SDB reflects the actual state of the gaming machine as far as possible while at the same time, the gaming machine meters must satisfy the self audit check.

If the gaming machine meters to be reported in the SDB fail the self audit check, the machine must report the SDB Self Audit Error and the SDB Memory Error status flag both set to one in the same SDB.

A self audit error can only be cleared by performing a RAM reset operation.

This clause replaces section 3.3.17 & 18 of the National Standard. It also defines additional requirement(s) for section 3.3.19 of the National Standard.

Mechanical Meter Disconnect

5.1.20 This status condition is turned on at the time a disconnect condition of any mechanical meter is detected. It is turned off when the meter reconnected condition is detected.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for mechanical meter disconnect as it is stated in section 3.16.5 of the National Standard.

Display Error

5.1.21 This refers to an error condition detected on the display medium. Examples of display error are stepper “reel not spinning freely” and illegal reel movement.

The status condition is turned on when a display error condition is detected and is turned off when an authorised person executes a specified reset procedure.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for display related error as it is stated in section 3.16.5 of the National Standard.

Cash Input Error

5.1.22 This refers to an error or fault condition in the coin input system or the note acceptor system. That is, the cash input error status is a global status condition that can cover many different possible error conditions associated with coin or note input.

The details of the error type can be reported on the gaming machine or by the coin or note input system and the manufacturer may further wish to make use of the MDB for electronically reporting such details.

The status condition is turned on when an error or fault condition is detected and is typically turned off when an authorised person executes a specified reset procedure.

This status condition is reported on the display and the communications ports.

Cash Output Error

- 5.1.23 This refers to the detection of a fault condition in the coin output (hopper) system. Examples of cash output error are hopper empty, hopper jam, extra coin paid, hopper run away and hopper failure.

This status condition is turned on when a cash output error is detected and is turned off when the specified reset procedure is executed.

This status condition is reported on the display and the communications ports.

Manufacturer Specific Error

- 5.1.24 This refers to any error condition that is not mentioned in this document and causes the gaming machine to assume a "locked" condition.

The status condition is turned on when the error is detected and is turned off when an authorised person executes the specified reset procedure or the error condition is no longer detected.

This status condition is reported on the display and the communications ports.

Gaming Machine Power Save

- 5.1.25 In this condition, power is available to the logic board(s), subsidiary equipment interfaces and all security sensing devices while the display and other "lighting" may have power removed.

The power save status condition is turned on when the power save condition is detected and turned off when the power save mode is exited.

This status condition is reported on the communications ports.

Subsidiary Equipment Play Suspended

- 5.1.26 This refers to the gaming machine detecting a "failed" subsidiary equipment function signal (SEF) on any one of the ports P1 to P6. The SEF signal is generated by the interfaces connected to the ports or "looped back" using a "cap" when the port is not used.

The gaming machine must examine ("read") the signal on all ports "continuously" e.g. at least two times per second.

If SEF signal failure is detected on P1 to P4 and a game is in progress, play must not be suspended until the game is completed (machine returns to the idle state). It must not be possible for subsidiary equipment connected to Ports 1 to 4 to interrupt a game in progress.

For ports P1 to P4, the subsidiary equipment play suspended status bit (SDB Byte 12 bit #3) will be set (to '1') and the gaming machine will be in "idle, subsidiary equipment play suspended" state as a result of the SEF signal failing. The player must be able to redeem credits when the machine is in this state.

For gaming machines that support ticket printer function, neither the subsidiary equipment play suspended status bit will be set to one nor the subsidiary equipment play suspended status condition will be entered when any failed SEF signal is detected on port P3. The gaming machine must not issue any ticket under this situation (including cases where the host system is off-line) except a ticket that has started printing prior to the condition occurring, which is allowed to be completed.

If SEF signal failure is detected on P5 or P6, the gaming machine shall suspend game play immediately i.e. the game is to be interrupted. The purpose of "immediate suspension" is to avoid situations where the gaming machine can continue to operate when the progressive controller has partially or completely failed and the gaming machine operation is authorised for use only in conjunction with a progressive controller or continued operation can result in disputes about very large progressive prizes that may have accumulated and can potentially be claimed during malfunction situations.

Because of the "immediate suspension" requirement, it is possible for ports P5 and P6 to cause the machine to enter a "game play, P5/P6 subsidiary equipment play suspended" state i.e. game not completed and player cannot redeem credit. Note that a gaming machine only enters a "P5/P6 subsidiary equipment play suspended" state when it detects a failed SEF signal on port P5 and/or P6.

"Immediate suspension" in this context will mean halting the gaming machine where technically feasible, given the operation of the program. In a game play, for example, halting of the game may be postponed until after "critical states or operations" have been completed e.g.: "reel spin". However, it should not be possible, for example, to continue onto further parts of a game such as double-ups or free spins.

It must be possible to return the gaming machine from a "game play, P5/P6 subsidiary equipment play suspended" state to an "idle, no lockup" or "idle, P5/P6 subsidiary equipment play suspended" state by the following methods:

- a) After a standard linked gaming machine detects a valid manual activation of the Jackpot Reset key-switch procedure or successfully accepts and processes a CCCE progressive win payment transfer utilising the CCCE commands A1/A2 while being in a "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" state, it must reset the standard linked progressive win status bit (SDB Byte 10 bit #3) to zero, and enable the game play to complete unless any new and/or other lockup status condition(s) is/are detected. Once the game play is completed, the machine must enter an "idle, no lockup" state if it does not detect any lockup status condition. Note that the link controller must not initiate any CCCE progressive win payment transfer to the winning gaming machine if the jackpot win exceeds \$3,000.
- b) In the cases where a standard linked gaming machine is in a "game play, P5/P6 subsidiary equipment play suspended" state and there is a probability that one or more "standard linked jackpot" winning combinations may occur when the remaining of the game play is enabled to continue, the machine must remain in the "game play, P5/P6 subsidiary equipment play suspended" state (without enabling any game play) until no more failed SEF signal on ports P5 and P6 is detected. Once no more failed SEF signal on ports P5 and P6 is detected, the machine must enable the game play

to complete unless any new and/or other lockup status condition(s) is/are detected. At the end of the game play, the machine must enter an "idle, no lockup" state if it does not detect any lockup status condition.

- c) For all other cases than those listed in points 'a' and 'b' above, after a gaming machine detects an intervention of a venue attendant (ie., an appropriate manual key-switch procedure is executed) while being in a "game play, P5/P6 subsidiary equipment play suspended" state, it must enable the game play to complete unless any new and/or other lockup condition(s) is/are detected. Once the game play is completed, the machine must enter an "idle, P5/P6 subsidiary equipment play suspended" state if it still detects any failed SEF signal on port P5 and/or P6.

When the gaming machine is in an "idle, no lockup" or "idle, P5/P6 subsidiary equipment play suspended" state, the player can redeem accumulated credits from the machine. Note that other credit redemption conditions are defined in sections 3.7.1 and 8.2.1 of the National Standard.

During suspension of play, critical activities such as cycling of the random number generator must continue.

The subsidiary equipment play suspended status condition is automatically turned off when a failed SEF signal is no longer detected.

This status condition is reported on the display and the communications ports.

Note: Any change in the subsidiary equipment play suspended status condition and the port status must be updated in the same SDB.

Byte 84 needs to reflect the current status of the SEF ports at all times.

The requirement for "External Peripheral Controller Fault / Disconnect", as it is stated in section 3.16.5 of the National Standard, is replaced with that defined in this clause and clause 2.4.77 of the NSW Appendix to the National Standard.

Cancel Credit Error

5.1.27 This refers to the detection of a fault condition in the cancel credit process. As an example in the case of ticket printers integrated into a gaming machine, it may not be possible to complete the cancel credit process because the machine has detected a ticket printer error such as a paper jam (or paper out) in the ticket printer.

This status condition is turned on when the machine detects a failure in the cancel credit process, and is turned off when the machine no longer detects a failure.

This status condition is reported on the display and the communications ports.

Note: This clause defines additional requirement(s) for cancel credit error as it is stated in section 3.16.5 of the National Standard.

6 Communication Ports

6.1 Communication ports to be provided

The following ports are defined:

- P1 – Port for equipment using CCCE Command Class(es)
- P2 - Centralised Monitoring System Port (CMS Port)
- P3 - Data Gathering Port
- P4 - Promotional Equipment Port
- P5 - Progressive Port 1
- P6 - Progressive Port 2

6.2 Port Overview

Port P1

- 6.2.1 This port is capable of two-way communication. When being configured not to support any electronic money transfer (CCCE money transfer), apart from the Subsidiary Equipment Function (SEF) input signal, it must be able to accept and process the 'Time and Date' command class 4, the 'Machine Lockup' command set 50H-52H and the 'Message Display' command set 53H-54H. This port must transmit the Standard Data Block, and if applicable, the Manufacturer Data Block.

When being activated to process electronic money transfer (CCCE money transfer), this port in addition to the above-indicated functions, must also support the two CCCE transaction sets (the CCCE 'Increment in Credits' command class A and 'Decrement in Credits' command class B) that allow the credit to be incremented and decremented, and if applicable, other approved command classes.

The only permitted method for activating this port to support CCCE command class(es) is by the use of the program/software configuration set-up in the gaming machine or separate gaming program/software. It must involve opening the sealed logic cage containing the game controller board and program PSD for at least part of the procedure.

The manufacturer must bear in mind the need for authorised persons to be able to simply and effectively determine whether or not the CCCE money transfer functionality of the gaming machine (including port P1) is enabled or disabled. This is to be achieved through the audit screen display using the "CCCE Transfer Limit" field.

In the case where the CCCE money transfer functionality is disabled, the field is to indicate (at least) "OFF \$0.00". In the case where this functionality is enabled, the field is

to read (at least) "ON \$####.##" where \$####.## represents the maximum CCCE money transfer limit.

Note: The existing approved command classes are: 'Time and Date' command class 4, command class 5 ('Machine Lockup', 'Message Display', 'Program Signature Request', 'Non-Cash Prize Ticket Print'), 'Progressive Current Pool Value' command classes 6 & E, 'Ticket In Information' command class 7, 'Progressive Win Notification' command classes 8 & 9, command class A ('Increment In Credits', 'Increment In Cents'), command class B ('Decrement In Credits', 'Decrement In Cents'), and 'Progressive Win Payment' command classes C & D.

The command classes 4, 6 and 7 are defined in the TITO technical standard.

The command classes 5, 8, 9, A (A3h-A4h), B (B3h-B4h), C, D and E are defined in sections 6.6 to 6.10.4 of this standard.

The CCCE 'Increment In Credits' command class A (A1h-A2h) and 'Decrement In Credits' command class B (B1h-B2h) are also known as the CCCE transaction sets 'T1 and T2' respectively. These transaction sets are defined in sections 6.3.2 to 6.3.4 of this standard.

Ports P2, P3, P4, P5, P6

6.2.2 With the exception of the subsidiary equipment function signal, these ports are only capable of one way communication. In general terms, the gaming machine (Ports P2, P3, P4, P5 and P6) will output the Standard Data Block, Function Data Block, Progressive Data Block 1 and Progressive Data Block 2 whenever certain events occur. These ports may also output the Manufacturer Data Block.

General

6.2.3 All Ports P1 to P6 must be provided with a facility to "read" a "subsidiary equipment function" signal from an interface board. All unused ports must be sealed with a "cap".

All ports must be clearly identified to facilitate correct connection of subsidiary equipment.

To protect against any interference (malfunction) that could arise due to the connection of subsidiary equipment on another port, the ports must be electrically "isolated" from each other. In practice the minimum requirement would be separate "line drivers" for each port.

The "line drivers" must either be socketed to facilitate replacement of the drivers if required or directly mounted by means of surface mount technology onto the circuit board containing the ports. If the latter is used, the manufacturer is to bear in mind that whenever a line driver becomes faulty, the circuit board containing the faulty line driver will need to be replaced (instead of the integrated chip containing the faulty line driver).

6.3 Port Details. Port P1

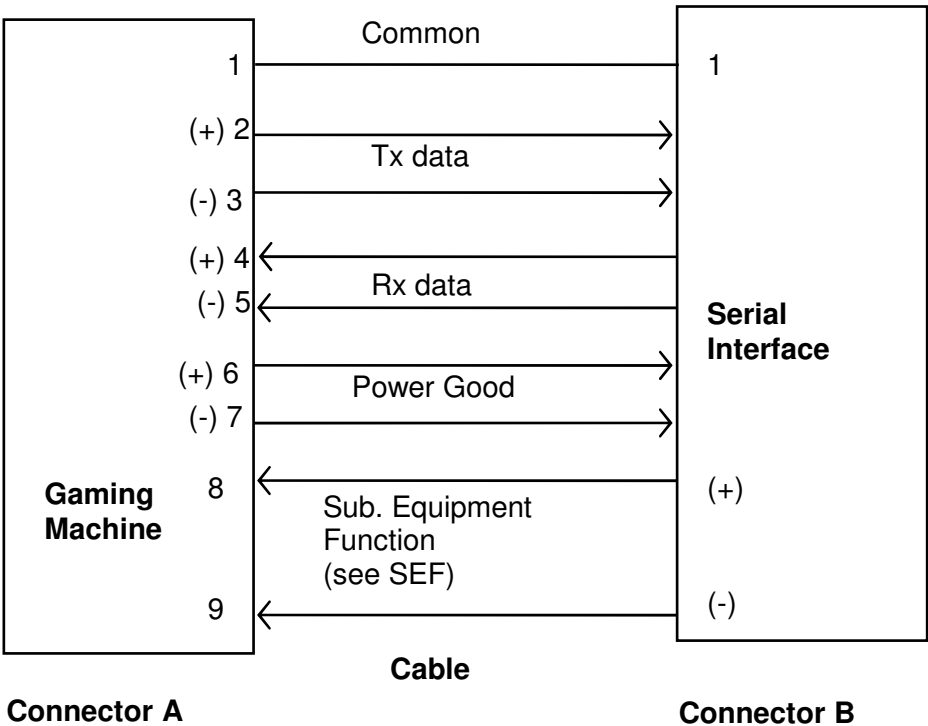
Hardware

6.3.1 EIA RS422 Electrical specification is to be followed by the interface ports. This specification covers the areas of signal level, slew rate and so on.

The speed of transmission is to be 9600 bps.

Each character or digit is to be represented by 11 bits (1 start bit + 8 data bits (packed BCD) + 1 parity bit (even) + 1 stop bit). The data byte is to be transmitted with the LSB first, parity is transmitted after the MSB.

Connection Diagram



Connector A is to be either a 10 pin straight or right angle header (standard 0.05" pitch). Each connector must have latch/eject levers and a centre polarisation slot. An example of this type of connector is the Amphenol 816 series.

Note: Even though a 10 pin connector is a non-standard connector for RS422, it will satisfy this requirement.

Standard ribbon cable, 0.05", to a maximum length of 1.5 metres is to be used. The interface will always be in communication mode when active. The Power Good and Subsidiary Equipment Function signals are as defined in the Sections titled "Power Good Signal" and "Subsidiary Equipment Function (SEF) Signal" below.

CCCE Protocol (Increment/Decrement In Credits)

Overview

6.3.2 Port P1 will output the Standard Data Block at the same time as ports P2,P3,P4,P5,P6 and in the "idle, no lockup" mode at intervals of 1.5 seconds (+/- 0.15 seconds).

Port P1 will **not output** any Function Data Block, Progressive Data Block 1 and Progressive Data Block 2. If it outputs a Manufacturer Data Block, it must do so in a manner that does not interfere with any CCCE transaction sequence.

Receipt of the SDB will be interpreted by the CCCE or the link controller as a signal to commence a money transfer, provided that the "idle, no lockup", "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" or "idle, P2-P6 subsidiary equipment play suspended" status conditions apply. In the "idle, no lockup" or "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" conditions, transaction set T1 is permitted. In the "idle, no lockup" or "idle, P2-P6 subsidiary equipment play suspended" conditions, transaction set T2 is permitted.

CCCE transactions are subject to time constraints which are described below in "Transaction Time Limits, Time Out".

After a timeout, P1 will not process any information received and a transaction may only recommence when the gaming machine transmits the next SDB, thereby opening another "window" for the transfer of money.

Definition of the "idle, no lockup", "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" and "idle, P2-P6 subsidiary equipment play suspended" status conditions is given below:

Status byte 1	#0 = w (Idle) #1 = x (Game Cycle) #2 = 0 or 1 (Power up) #3 = 0 or 1 (Reset) #4 = 0 or 1 (CCCE Transfer Completed) #5 = 0 (spare) #6 = 0 (spare) #7 = 0
Status byte 2	#0 = 0 (Large Win or Large Accumulated Credit) #1 = 0 (Collect Cash) #2 = 0 (Cancel Credit) #3 = y (Progressive Win) #4 = 0 (Manufacturer Specific Win) #5 = 0 (Manufacturer Specific Win) #6 = 0 (Manufacturer Specific Win) #7 = 0
Status byte 3	= 00
Status byte 4	#0 = 0 (Audit Mode) #1 = 0 (Test Mode) #2 = 0 (Power Save) #3 = z (Subsidiary Equipment Play Suspended)

#4 = 0 (Mechanical Meter Disconnect)
#5 = 0 (Manufacturer Specific Error)
#6 = 0 (Manufacturer Specific Error)
#7 = 0

Note that:

- 1) The "idle, no lockup" status condition is defined as $w = 1$, $x = 0$, $y = 0$ and $z = 0$.
- 2) The "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" status condition is defined as $w = 0$, $x = 1$, $y = 1$, $z = 1$ and SDB Byte 84 bit #4 = 0 (or SDB Byte 84 bit #5 = 0).
- 3) The "idle, P2-P6 subsidiary equipment play suspended" status condition is defined as $w = 1$, $x = 0$, $y = 0$, $z = 1$ and SDB Byte 84 bit #0 = 1.
- 4) With respect to the CCCE transaction sets T1 and T2, link controllers are permitted to support the transaction set T1 only.
- 5) Only standard link controllers are permitted to initiate a CCCE progressive win payment transfer using the transaction set T1 to the winning gaming machine when the machine is in a "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" state.

The status bytes are defined in the "Standard Data Block" below.

Transaction Time Limits, Timeout

(Please refer to Information Flow below before reading this section).

A money transfer may only commence after the gaming machine transmits the last byte of a SDB and an appropriate status condition applies.

The gaming machine will assume a timeout has taken place if a complete "phase 1 transfer request" or a "phase 2 transfer request" is not received within 40ms from the time that the last byte of the SDB or the "phase 1 echo" is transmitted.

The gaming machine will respond to a legitimate "phase 1 transfer request" by completing the transmission of the "phase 1 echo" within 40ms of receiving the last byte of the "phase 1 transfer request".

The gaming machine will respond to a legitimate "phase 2 transfer request" by completing the transmission of the SDB within 200 mS of receiving the last byte of the "phase 2 transfer request".

Restarting a Transaction (T1 and T2)

Once the gaming machine terminates a transaction, a new transaction may only recommence when the gaming machine transmits the next SDB, thereby opening another "window" for the transfer of money.

If the status of the gaming machine changes during the transaction, the gaming machine will cease the normal sequence and output a SDB that reflects the changed status.

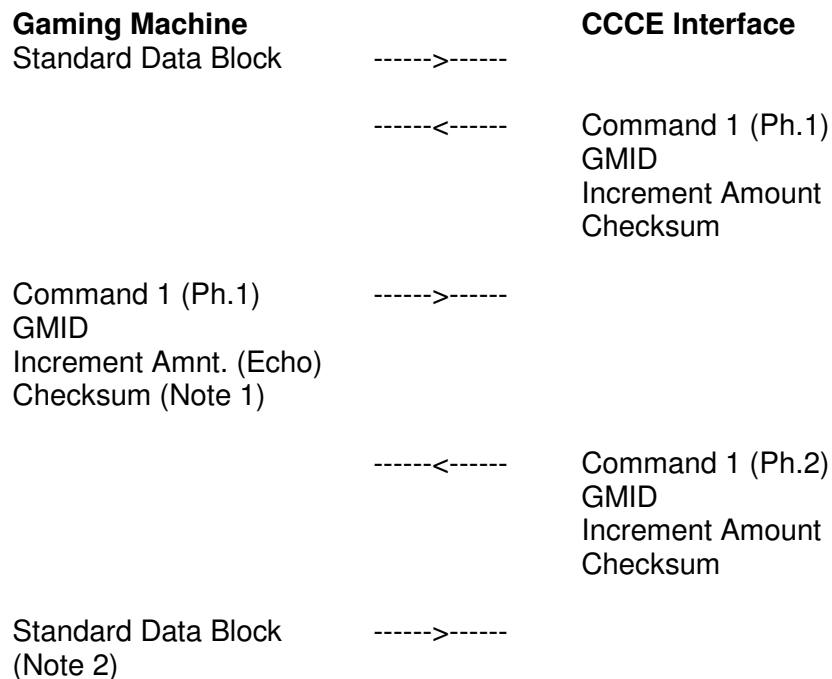
If the change in status occurs during a transmission, the transmission must be completed before the SDB is output. Note that only that portion of the transaction currently being transmitted needs to be completed not the entire transaction.

CCCE Transaction Set 1 (T1) (Increment In Credits)

- 6.3.3 This transaction effectively transfers money from the CCCE or the link controller into the gaming machine. It can only be executed if the "idle, no lockup" or "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" status condition applies. Note that only standard link controllers are permitted to initiate a CCCE progressive win payment transfer using transaction set T1 to the gaming machine when the machine is in a "game play, standard linked progressive win, P5/P6 subsidiary equipment play suspended" state. The money is transferred in units of the gaming machine base credit value. The gaming machine base credit value is available to the subsidiary equipment in the SDB.

T1 - Information flow:

(Also refer to "Transaction Time Limits, Timeout" above.)



Note 1: If an error in transmission occurs or if the CCCE transfer limit is exceeded, then the gaming machine will not respond and the transaction will be terminated.

Note 2: A successful transaction is determined by transmission of a SDB with the Credit and the Money In meters reflecting the appropriate increments as well as CCCE transfer completed status bit of the SDB set reflecting a successful transfer.

If an error occurs or the phase 1 and phase 2 amounts differ, then the gaming machine will not respond and the transaction will be terminated.

If a Timeout (see above) occurs, the gaming machine will terminate the transaction.

Byte transfer details CCCE to Gaming Machine Phase 1

Byte 1 (FF)- Start of block (unique)

Byte 2 - Command 1, Phase 1 (A1H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Increment Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Increment Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Increment Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

Byte 10 - Checksum upper nibble (X0H)

Byte transfer details CCCE to Gaming Machine Phase 2

Byte 1 (FF) - Start of block (unique)

Byte 2 - Command 1, Phase 2 (A2H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Increment Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Increment Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Increment Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

Byte 10 - Checksum upper nibble (X0H)

Byte transfer details Gaming Machine to CCCE

Standard data block is defined in Section 6.5.2 of this standard.

Byte transfer details Gaming Machine to CCCE (Echo)

Byte 1 (FF)- Start of block (unique)

Byte 2 - Command 1, Phase 1 (A1H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Increment Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Increment Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Increment Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

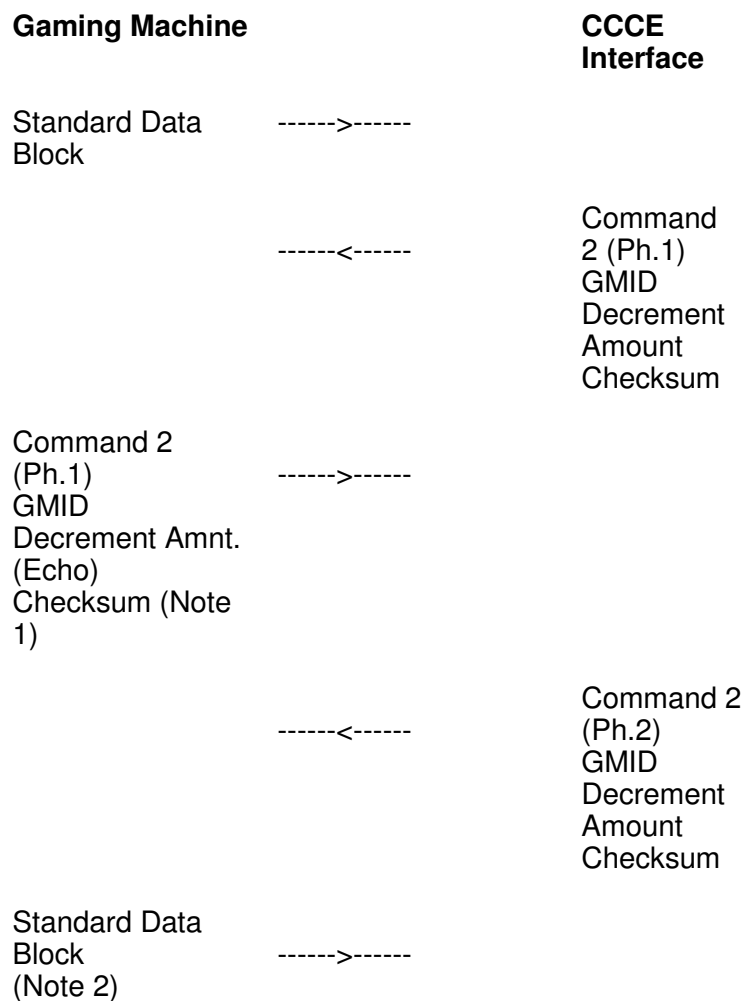
Byte 10 - Checksum upper nibble (X0H)

CCCE Transaction Set 2 (T2) (Decrement In Credits)

- 6.3.4 This transaction transfers money from a gaming machine to the CCCE. It can only be executed if the "idle, no lockup" or "idle, P2-P6 subsidiary equipment play suspended" status condition applies. The money is transferred in units of the gaming machine base credit value. The gaming machine base credit value is available to the subsidiary equipment in the SDB.

T2 - Information flow:

(Also refer to "Transaction Time Limits, Timeout" above.)



Note 1: If an error in transmission occurs or if the CCCE transfer limit is exceeded or the decrement amount exceeds the available credit, then the gaming machine will not respond and the transaction will terminate.

Note 2: A successful transaction is determined by transmission of a SDB with the Credit and the Money Out meters reflecting the appropriate decrement and increment as well as the CCCE transfer completed status bit of the SDB set reflecting a successful money transfer.

If an error occurs or the phase 1 and phase 2 amounts differ, then the gaming machine will not respond and the transaction will terminate.

If a Timeout (see above) occurs, the gaming machine will terminate the transaction.

Byte transfer details CCCE to Gaming Machine Phase 1

Byte 1 (FF) - Start of block (unique)

Byte 2 - Command 2, Phase 1 (B1H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Decrement Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Decrement Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Decrement Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

Byte 10 - Checksum upper nibble (X0H)

Byte transfer details CCCE to Gaming Machine Phase 2

Byte 1 (FF) - Start of block (unique)

Byte 2 - Command 2, Phase 2 (B2H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Decrement Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Decrement Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Decrement Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

Byte 10 - Checksum upper nibble (X0H)

Byte transfer details Gaming Machine to CCCE

Standard data block is defined in Section 6.5.2 of this standard.

Byte transfer details Gaming Machine to CCCE (Echo)

Byte 1 (FF) - Start of block (unique)

Byte 2 - Command 2, Phase 1 (B1H)

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Decrement Amount LSD and LSD+1. LSD in lower nibble.

Byte 7 - Decrement Amount LSD+2 and LSD+3. LSD+2 in lower nibble.

Byte 8 - Decrement Amount LSD+4 and MSD. LSD+4 in lower nibble.

Byte 9 - Checksum lower nibble (0XH)

Byte 10 - Checksum upper nibble (X0H)

6.4 Port Details. Ports P2, P3, P4, P5, P6

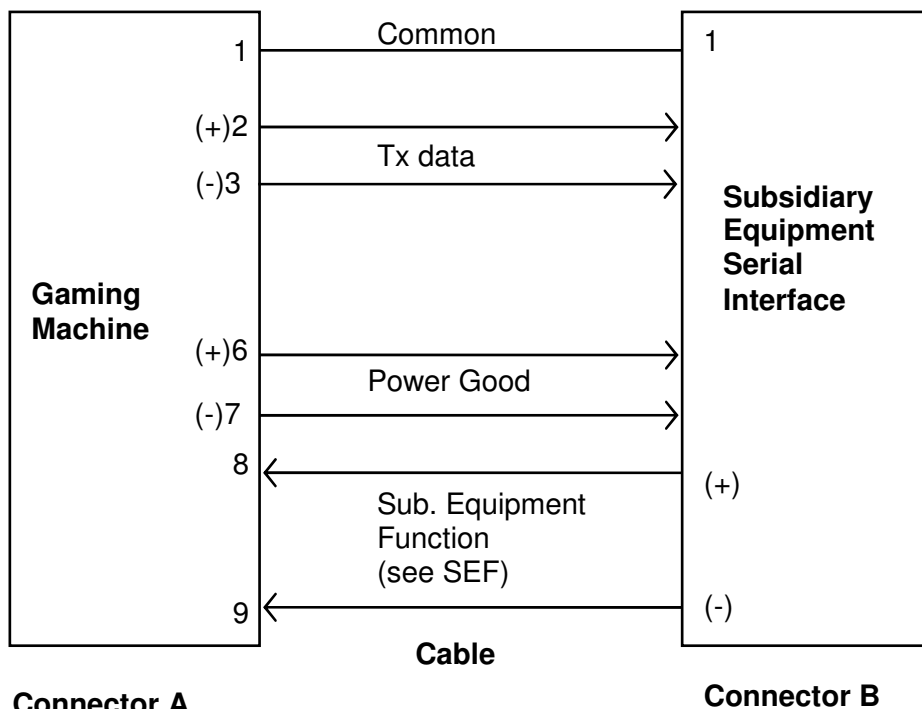
Hardware

- 6.4.1 The EIA RS422 Electrical specification will be followed. This specification covers the areas of signal level, slew rate and so on.

Speed of transmission is to be 9600 bps.

Each character or digit is to be represented by 11 bits (1 start bit + 8 data bits (generally, packed BCD format) + 1 parity bit (even parity) + 1 stop bit). The data byte is transmitted with LSB first, parity being transmitted after the MSB.

Connection Diagram



Connector A is to be either a 10 pin straight or right angle header (standard 0.05" pitch). Each connector must have latch/eject levers and a centre polarisation slot. An example of this connector type is the Amphenol 816 series.

Note: Even though a 10 pin connector is non-standard for RS422, it will satisfy the requirements of this interface specification.

Standard ribbon cable, 0.05", to a maximum length of 1.5 metres is to be used.

The interface will always be in the communication mode.

Power Good Signal

- 6.4.2 The power good indicator (pins 6 & 7) will operate such that the differential voltage at the output of the line driver will be negative ($V_{id} = -V$) whenever the main logic board and any other "sensitive" subsidiary boards have "normal" power applied. "Normal" power will mean that the board(s) can continue performing all functions with the exception of any functions that would not normally be available in the power save state.

Under all other conditions, the differential voltage at the output of the line driver should be positive ($V_d = +V$), however the manufacturer of the subsidiary equipment interface must ensure that other power fail conditions which may cause the output of the line driver to assume zero voltage or high impedance are correctly detected as power fail conditions. The gaming machine will act as the signal source.

Subsidiary Equipment Function (SEF) Signal

- 6.4.3 The SEF signal is an indication to the gaming machine that subsidiary equipment connected is functioning correctly however, it can also be used as a general purpose signal by subsidiary equipment to request play suspension.

Under "normal" conditions, the SEF differential signal voltage at the output of the interface line driver is negative ($V_d = -V$). A positive signal ($V_d = +V$) is to be interpreted by the gaming machine as a request to suspend play however, other "fail" conditions which may cause the output of the interface line driver to assume zero voltage or high impedance must also be interpreted as a request to suspend play by the gaming machine.

When subsidiary equipment is not connected, the port must be "capped". In the "cap" for the connector, pins 6 & 7 (Power Good) and pins 8 & 9 (SEF) must be linked so that the gaming machine can be provided with the SEF input signal.

When subsidiary equipment is to be connected, this "cap" must be removed and the signal must be controlled by the subsidiary equipment.

6.5 Data Transmission

Overview

- 6.5.1 Five types of data blocks (128 bytes each) can be output by ports P2, P3, P4, P5 and P6 of the gaming machine. These data blocks are:

1. Standard Data Block (SDB); it is a mandatory data block to be output by the machine.
2. Function Data Block (FDB); it is output if the machine supports secondary functions (i.e. standard linked progressive functions).
3. Progressive Data Block 1 (PDB1); it is output if the machine supports integrated standalone progressive functions (i.e. standalone gaming machine with integrated standalone progressive jackpot system).
4. Progressive Data Block 2 (PDB2); it is only output if the machine supports PDB1 (PDB1 is followed directly by PDB2).
5. Manufacturer Data Block (MDB); it is intended for manufacturer specific use.

Note: On power up, reset or RAM reset, the SDB must be output first, if applicable, followed directly by FDB, PDB1, PDB2 and MDB.

Requirements for the SDB, FDB, PDB1-PDB2 pair and MDB are correspondingly defined in sections 6.5.2 to 6.5.5 of this standard.

6.5.1a Apart from SDB and MDB, eight classes of CCCE commands can be output by port P1 of the gaming machine. They are:

1. CCCE 'Increment In Credits / Cents' command class A – Phase 1 Echo;
2. CCCE 'Decrement In Credits / Cents' command class B – Phase 1 Echo;
3. 'Progressive Win Notification with CCCE Pay' command class 8 – Echo;
4. 'Progressive Win Notification with Hand Pay' command class 9 – Echo;
5. 'Standard Progressive Win Payment' command class C – Phase 1 Echo;
6. 'Mystery Progressive Win Payment' command class D – Phase 1 Echo;
7. 'Ticket In Information' command class 7 – Ticket In Information packet and Phase 1 Echo; and
8. Command class 5 – Machine Lockup Command Echo,
Player Acknowledgement Receipt,
Message Display Command Echo,
Program Signature Request Command Echo,
Program Signature Result,
Non-Cash Prize Ticket Print Command Echo

The gaming machine port P1 must not output any FDB, PDB1 and PDB2.

Note: Each of the above-mentioned CCCE command classes consists of more than one CCCE command. For details of the CCCE commands in each class, refer to the appropriate section(s) of the standard.

The CCCE 'Increment In Credits' command class A and 'Decrement In Credits' command class B are also known as the CCCE transaction sets 'T1' and 'T2' respectively. Requirements for these transaction sets are listed in sections 6.3.2 to 6.3.4 of this standard.

Requirements for the CCCE command classes 5, 8, 9, A (A3h-A4h) and B (B3h-B4h), C and D are listed in sections 6.6 to 6.10.4 of this standard.

Requirements for the CCCE command class 7 are defined in the TITO technical standard.

Requirements for SDB and MDB are respectively defined in sections 6.5.2 and 6.5.5 of this standard.

Standard Data Block (SDB)

6.5.2 The SDB is transmitted by ports P1, P2, P3, P4, P5 and P6 of the gaming machine whenever:

- (a) A status change from idle to game cycle occurs i.e. at the beginning of each new game play.

- (b) A status change occurs in any of status bytes 2, 3, 4 or 5 i.e. any lockup condition is entered or exited.
- (c) Any successful money transfer between the gaming machine and CCCE interface.
- (d) Any power up or reset.
- (e) A period of 1.5 seconds (+/- 0.15 seconds) has elapsed since the last transmission when the machine status is in a "idle, non-lockup" state. And it must not be transmitted more frequently than 1.5 seconds (+/- 0.15 seconds).
- (f) A period of 1.5 seconds (+/- 0.15 seconds) has elapsed since the last transmission when the machine status is in a lockup state. And it must not be transmitted more frequently than 1.5 seconds (+/- 0.15 seconds).
- (g) A period of 15 seconds (+/- 1.5 seconds) has elapsed since the last transmission when the machine status is in a "game cycle, non-lockup" state i.e. a normal game lasting longer than 15 seconds. The block may be transmitted at more frequent intervals but not more frequently than 1.5 seconds (+/- 0.15 seconds).

If a change in state occurs during a transmission, the transmission of the block is not to be affected, i.e. the block is to be transmitted as if the change in state did not occur until after the transmission is complete.

If any one or more status bits change "at the same time", then the SDB is output once only provided that it accurately reflects the status of the gaming machine. The gaming machine must output at least one SDB with the SDB Byte 9 bit #0, "Idle" status bit being set to one, after the completion of each game play (i.e. game cycle) and before any new bet may be accepted by the gaming machine.

As for an example, the game play status bit is set and the reset and power up status bits are cleared at the same time, then the block is transmitted only once.

There must be a minimum of 20ms delay in transmission between any two consecutive data blocks, with the exception of CCCE transfer commands.

Standard Data Block Structure

Information is to be encoded in packed BCD format except in cases where it is impractical to apply this technique e.g. Start of block, status bytes 1, 2, 3 and 4 and firmware identification.

Byte 1 (FF) - Start of block (this is a unique code and must not appear anywhere else in the block)

Byte 2 (00) - SDB Identifier

Byte 3 (02) - SDB Data block version number LSD and LSD+1. LSD in lower nibble.

Byte 4 (01) - SDB Data block version number LSD+2 and MSD. LSD+2 in lower nibble.

(Note: this is interpreted as version 1.02 which supersedes version 1.01)

Byte 5 (XX) - Block Sequence Counter

Byte 6 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Status byte 1 (non-lockup)

- #0 - Idle
- #1 - Game Cycle
- #2 - Power Up
- #3 - Reset
- #4 - CCCE transfer completed
- #5 - 0 (spare)
- #6 - 0 (spare)
- #7 - 0

Byte 10 - Status Byte 2 (lockup)

- #0 - Large Win or Large Accumulated Credit (0 if not used)
- #1 - Collect Cash
- #2 - Cancel Credit (Book Pay)
- #3 - Progressive win (0 if not used)
- #4 - Manufacturer specific win (0 if not used)
- #5 - Manufacturer specific win (0 if not used)
- #6 - Manufacturer specific win (0 if not used)
- #7 - 0

Byte 11 - Status Byte 3 (lockup)

- #0 - Main Door(s) Open
- #1 - Security Cage(s) Open
- #2 - Display error (0 if not used)
- #3 - Self Audit Error
- #4 - Memory Error
- #5 - Cash Input Error
- #6 - Cash Output Error
- #7 - 0

Byte 12 - Status Byte 4 (lockup)

- #0 - Audit mode
- #1 - Test mode
- #2 - Power Save
- #3 - Subsidiary equipment play suspended
- #4 - Mechanical Meter Disconnect
- #5 - Manufacturer Specific Error (0 if not used)
- #6 - Manufacturer Specific Error (0 if not used)
- #7 - 0

Byte 13 - Status Byte 5 (lockup)

- #0 - Cancel Credit Error
- #1 - 0 (spare)
- #2 - 0 (spare)
- #3 - 0 (spare)
- #4 - 0 (spare)
- #5 - 0 (spare)

#6 - 0 (spare)
#7 - 0

Byte 14 - 00 (Spare)

Byte 15 - Game number of the multi-game. The Game Number byte is set to "00" if no multi-game option available.

Byte 16 - Multi-game combination number. The Multi-game combination number is set to "00" if no multi-game option available.

As for an example, to indicate that game 9 of the (game) set is being played using the multi-game combination 2, SDB Byte 15 = 09 and SDB Byte 16 = 02.

Byte 17 - Turnover meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 21 - Turnover meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 22 - Total Wins meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 26 - Total Wins meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 27 - Cashbox meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 31 - Cashbox meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 32 - Cancelled Credit meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 36 - Cancelled Credits meter (LSD+8 and MSD). LSD+8 in lower nibble

Byte 37 - Games Played Meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 40 - Games Played Meter (LSD+6 and MSD). LSD+6 in lower nibble.

Byte 41 - Spare (00)

Byte 42 - Money In meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 46 - Money In meter (LSD+8 and MSD). LSD+8 in lower nibble.

Note: If this meter is not used, the bytes must contain 00.

Byte 47 - Money Out meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 51 - Money Out meter (LSD+8 and MSD). LSD+8 in lower nibble.

Note: If this meter is not used, the bytes must contain 00.

Byte 52 - Cash In meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 56 - Cash In meter (LSD+8 and MSD). LSD+8 in lower nibble.

Note: If this meter is not used, the bytes must contain 00.

Byte 57 - Cash Out meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 61 - Cash Out meter (LSD+8 and MSD). LSD+8 in lower nibble.

Note: If this meter is not used, the bytes must contain 00.

Byte 62 - Credit meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 66 - Credit meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 67 - Miscellaneous accrual meter (LSD and LSD+1). LSD in lower nibble.

.
Byte 71 - Miscellaneous accrual meter (LSD+8 and MSD). LSD+8 in lower nibble.

Note: If this meter is not used, the bytes must contain 00.

Byte 72 - Power up (LSD and LSD+1). LSD in lower nibble.

.
Byte 75 - Power up (LSD+6 and MSD). LSD+6 in lower nibble.

Byte 76 - Games played since last Power up (LSD and LSD+1). LSD in lower nibble.

.
Byte 79 - Games played since last Power up (LSD+6 and MSD). LSD+6 in lower nibble.

Byte 80 - Games played since last (Main) Door open (LSD and LSD+1). LSD in lower nibble.

.
Byte 83 - Games played since last (Main) Door open (LSD+6 and MSD). LSD+6 in lower nibble.

Byte 84 - Port status byte

#0 Port P1, SEF signal status

#1 Port P2, SEF signal status

#2 Port P3, SEF signal status

#3 Port P4, SEF signal status

#4 Port P5, SEF signal status

#5 Port P6, SEF signal status

#6 = 0

#7 = 0

(**Note:** 1 = Subsidiary Equipment Function (SEF) signal present i.e. no request to suspend play)

Byte 85 - Base Credit Value (Expressed in units of cents, BCD. LSD and LSD+1. LSD in lower nibble)

Byte 86 - Base Credit Value (LSD+2 and MSD. LSD+2 in lower nibble)

Byte 87 - 00 (Spare)

Byte 88 - Program identification 1. 8 Ascii characters. Most significant character in byte 95, unused bytes to contain spaces (20H).

.
Byte 95

Byte 96 - Program identification 2. Format as for program ID 1.

.
Byte 103

Note: Spaces (20H) are to be stored in these bytes if not required.

Byte 104 - Program identification 3. Format as for program ID 1.

Byte 111

Note: Spaces (20H) are to be stored in these bytes if not required.

Byte 112 - Program Identification 4. Format as for program ID 1.

Byte 119

Note: Spaces (20H) are to be stored in these bytes if not required.

Byte 120 - Theoretical Percentage Return To Player (PRTP) (Expressed as a percentage, BCD. Right to decimal LSD and LSD+1. LSD in lower nibble.)

Byte 121 - Theoretical PRTP (Left to decimal LSD and MSD. LSD in lower nibble.)

e.g. PRTP 87.25%

Byte XX 25
Byte XX + 1 87

Byte 122 - Gaming Machine Secondary functions supported

#0 - Standard linked progressive function supported

#1 - (spare)

#6 - (spare)

#7 - 0

Byte 123 - 00 (Spare)

Byte 126 - 00 (Spare)

Byte 127 - Checksum lower nibble (0XH)

Byte 128 - Checksum upper nibble (X0H)

In the SDB, the data is to be interpreted by the receiver depending on its position in the data stream. A time period of approximately 140 ms is required to complete the transmission of 128 bytes (9600 bps).

The gaming machine must ensure that the transmission of each SDB is completed in a period not exceeding 200 mS. The timing will commence with the transmission of the first bit.

Function Data Block (FDB)

6.5.3 The FDB will follow the SDB only if any of the bits in the "secondary function supported" byte is set i.e. gaming machine supports secondary functions. At this time only the standard linked progressive function is defined as a secondary function.

The FDB is transmitted by ports P2, P3, P4, P5 and P6 of the gaming machine whenever:

- (a) A power up or reset procedure occurs i.e. during a low level to high level transition of the status bits corresponding to power up and reset and also when a power up or reset is performed without the status bits being cleared from the previous occasion.
- (b) A logic cage access occurs i.e. during a low to high level transition of the SDB byte 11 bit #1.
- (c) A RAM reset has occurred.
- (d) A period of 1.5 seconds (+/- 0.15 second) has elapsed since the last transmission when the machine is in a Jackpot Win "lockup" condition. This will ensure that progressive equipment receives a "win indication" at intervals of approximately 1.5 seconds.
- (e) At regular intervals every 30 seconds otherwise.

If a change in state occurs during a transmission, the current block being output is to be completed before the next block is sent.

There must be a minimum of 20ms delay in transmission between any two consecutive data blocks, with the exception of CCCE transfer commands.

Function Data Block Structure

Information is to be encoded in packed BCD format except in cases where it is impractical to apply this technique e.g. Start of block.

Byte 1 (FF) - Start of block (this is a unique code and must not appear anywhere else in the block)

Byte 2 (01) - FDB Identifier

Byte 3 (00) - FDB Data block version number LSD and LSD+1. LSD in lower nibble.

Byte 4 (01) - FDB Data block version number LSD+2 and MSD. LSD+2 in lower nibble.

Byte 5 (XX) - Block Sequence Counter

Byte 6 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Progressive Levels Supported (none = 00, 1 = 01, 2 = 02, 3 = 03, 4 = 04.)

Byte 10 - "Spare" currently this byte will contain 00H

Byte 11 - Progressive win level (no win=00, level 1 win= 01, level 2 =02, level 3=03, level 4=04).

Byte 12 - "Spare" currently this byte will contain 00H

Byte 13 - Progressive Jackpot, Level 1 Occurrence meter (LSD and LSD+1). LSD in lower nibble.

Byte 15 - Progressive Jackpot, Level 1 Occurrence meter (LSD+4 and MSD). LSD + 4 in lower nibble.

Byte 16 - Progressive Jackpot, Level 2 Occurrence meter (LSD and LSD+1). LSD in lower nibble.

Byte 18 - Progressive Jackpot, Level 2 Occurrence meter (LSD+4 and MSD). LSD + 4 in lower nibble.

Byte 19 - Progressive Jackpot, Level 3 Occurrence meter (LSD and LSD+1). LSD in lower nibble.

Byte 21 - Progressive Jackpot, Level 3 Occurrence meter (LSD+4 and MSD). LSD + 4 in lower nibble.

Byte 22 - Progressive Jackpot, Level 4 Occurrence meter (LSD and LSD+1). LSD in lower nibble.

Byte 24 - Progressive Jackpot, Level 4 Occurrence meter (LSD+4 and MSD). LSD + 4 in lower nibble.

Byte 25 - "Spare" currently this byte will contain 00H

Byte 26 - "Spare" currently this byte will contain 00H

Byte 27 - Probability of Progressive jackpot winning combination Level 1 (right to decimal LSD and LSD+1). LSD in lower nibble.

Byte 33 - Probability of Progressive jackpot winning combination Level 1 (right to decimal LSD+12 and MSD). LSD+12 in lower nibble.

e.g. Probability of 0.00000000000125

Byte 27 25
Byte 28 01
Byte 29 00
Byte 30 00
Byte 31 00
Byte 32 00
Byte 33 00

Byte 34 - Probability of Progressive jackpot winning combination Level 2 (right to decimal LSD and LSD+1). LSD in lower nibble.

Byte 40 - Probability of Progressive jackpot winning combination Level 2 (right to decimal LSD+12 and MSD). LSD+12 in lower nibble.

Byte 41 - Probability of Progressive jackpot winning combination Level 3 (right to decimal LSD and LSD+1). LSD in lower nibble.

Byte 47 - Probability of Progressive jackpot winning combination Level 3 (right to decimal LSD+12 and MSD). LSD+12 in lower nibble.

Byte 48 - Probability of Progressive jackpot winning combination Level 4 (right to decimal LSD and LSD+1). LSD in lower nibble.

Byte 54 - Probability of Progressive jackpot winning combination Level 4 (right to decimal LSD+12 and MSD). LSD+12 in lower nibble.

Byte 55 - 00 (spare)

Byte 126 - 00 (spare)

Byte 127 - Checksum lower nibble (0XH)

Byte 128 - Checksum upper nibble (X0H)

In the FDB, the data is to be interpreted by the receiver depending on its position in the data stream. A time period of approximately 140 ms is required to complete the transmission of 128 bytes (9600 bps).

The gaming machine must ensure that the transmission of each FDB is completed in a period not exceeding 200 mS. The timing will commence with the transmission of the first bit.

Progressive Data Block 1 (PDB1) and Progressive Data Block 2 (PDB2) Pair

6.5.4 The PDB1 and PDB2 are to be output as a pair by the integrated standalone progressive gaming machine (iSAP-GM which is a standalone gaming machine with integrated standalone progressive jackpot system). PDB1 must be always followed by PDB2. The PDB1-PDB2 pair will report the full status and metering information of the iSAP-GM. At any time, the PDB1-PDB2 pair is output, it must hold the same data as the equivalent electronic meters and status variables in the machine.

The PDB1-PDB2 pair is transmitted by ports P2, P3, P4, P5 and P6 of the iSAP-GM whenever:

- (a) A status change occurs. Status changes that cause an output are defined by PDB1 Status Byte 1, bits #0 and #1, and PDB1 Status Byte 3, bits #4 and #5. Note: For iSAP-GM, PDB1-PDB2 is to be output when change occurs in the "Normal", "Jackpot Hit" and "Manufacturer Specific Error" Status. The PDB1-PDB2 pair is output when the status condition is turned on and off (i.e. low level to high level [0 – 1] and high level to low level [1 – 0] transition of the status bits.
- (b) Any power up or reset procedure. This condition is necessary because a reset or power up may be performed a number of times without the status bits being cleared from the previous occasion.
- (c) A continuous period of 15 seconds (+/- 1.5 seconds) has elapsed since the last transmission and 1.5 seconds (+/- 0.15 seconds) under Jackpot Hit status condition.

If a change in state occurs during a transmission, the current block being output is to be completed before the next block is sent.

If any one or more status bits change "at the same time", then the progressive data block is output once only provided that it accurately reflects the current status and metering information of the iSAP-GM.

For example, the normal status bit is set and the reset and power up status bits are cleared at the same time, then the block needs only be transmitted once.

There must be a minimum of 20ms delay in transmission between any two consecutive data blocks, with the exception of CCCE transfer commands.

Note: The PDB1-PDB2 pair will have the same block sequence counter, i.e. any change of information in PDB1 data block, causing the PDB1 block sequence counter to increment will automatically cause the PDB2 block sequence counter to increment to the same value. In this way, it will be ensured that PDB1 and PDB2 are always synchronised and provide full “progressive related” status and metering information of the iSAP-GM at the time when the PDB1-PDB2 pair is output.

Progressive Data Block 1 Structure

Information is to be encoded in packed BCD format except in cases where it is impractical to apply this technique e.g. Start of block and status bytes 1, 2 and 3.

Byte 1 (FF) - Start of block (this is a unique code and must not appear anywhere else in the block)

Byte 2 (10) - PDB1 Identifier

Byte 3 (02) - PDB 1 version number LSD and LSD+1. LSD in lower nibble.

Byte 4 (01) - PDB 1 version number LSD+2 and MSD. LSD+2 in lower nibble.

Byte 5 (XX) - Block Sequence Counter

Byte 6 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - "Spare". Currently this byte will contain 00H.

Byte 10 - "Spare". Currently this byte will contain 00H.

Byte 11 - Status byte 1 (non lockup)

#0 - Normal
#1 - Jackpot Hit * (see note below)
#2 - Reset
#3 - Power Up
#4 - Audit Mode
#5 - 0 (spare)
#6 - 0 (spare)
#7 - 0

Note: “*” indicates that more frequent (every 1.5 second) PDB1 output is required when any of these status conditions are turned on.

Byte 12 - Status byte 2 (lookup)

- #0 - Gaming Machine Power Down
- #1 - Meter Disconnect
- #2 - Communication Failure
- #3 - Configuration Change
- #4 - 0 (spare)
- #5 - 0 (spare)
- #6 - 0 (spare)
- #7 - 0

Note: Reference Section 5.1.

Byte 13 - Status byte 3 (lookup)

- #0 - Security Cage Open
- #1 - Memory Error
- #2 - Power Save
- #3 - Test Mode
- #4 - Manufacturer Specific Error (0 if not used)
- #5 - Manufacturer Specific Error (0 if not used)
- #6 - 0 (spare)
- #7 - 0

Byte 14 - 00 (Spare)

Byte 15 - Progressive Levels Supported
(1 = 01, 2 = 02, 3 = 03, 4 = 04.)

Byte 16 - Set Increment Percentage Level 1 (Expressed in percentage, BCD. Right to decimal LSD and LSD + 1. LSD in lower nibble)

Byte 17 - Set Increment Percentage Level 1 (Expressed in percentage, BCD. Right to decimal LSD+2 and LSD+3. LSD+2 in lower nibble)

Byte 18 - Set Increment Percentage Level 1 (Expressed in percentage, BCD. Right to decimal LSD+4 and LSD+5. LSD+4 in lower nibble)

Byte 19 - Set Increment Percentage Level 1 (Left to decimal. LSD and MSD. LSD in lower nibble)

e.g. Increment Percentage = 1.987654%

Byte XX	54
Byte XX+1	76
Byte XX+2	98
Byte XX+3	01

Byte 20 - Set Increment Percentage Level 2 (Expressed in percentage, BCD. right to decimal LSD and LSD + 1. LSD in lower nibble)

Byte 23 - Set Increment Percentage Level 2 (left to decimal. LSD and MSD. LSD in lower nibble)

Byte 24 - Set Increment Percentage Level 3 (Expressed in percentage, BCD. right to decimal LSD and LSD + 1. LSD in lower nibble)

Byte 27 - Set Increment Percentage Level 3 (left to decimal. LSD and MSD. LSD in

lower nibble)

 Byte 28 - Set Increment Percentage Level 4 (Expressed in percentage, BCD. right to decimal LSD and LSD + 1. LSD in lower nibble)

.
 Byte 31 - Set Increment Percentage Level 4 (left to decimal. LSD and MSD. LSD in lower nibble)

 Byte 32 - Total Theoretical Standalone Progressive Percentage Return (Expressed in percentage, BCD. right to decimal LSD and LSD+1. LSD in lower nibble.)

.
 Byte 35 - Total Theoretical Standalone Progressive Percentage Return (left to decimal LSD and MSD. LSD in lower nibble)

(This is the total theoretical/estimated progressive PRTP for the standalone progressive jackpots supported by the iSAP-GM.)

 Byte 36 - Number of machines communicating (Expressed in BCD. LSD and LSD + 1. LSD in lower nibble)

Byte 37 - Number of machines communicating (Expressed in BCD. LSD and MSD. LSD in lower nibble)

“0001” (i.e. byte 36 = 01, byte 37 = 00) is the static value to be reported in PDB1 bytes 36 – 37 by the iSAP-GM.

 Byte 38 - 00 (Spare)

 Byte 39 - Number of machines not communicating (Expressed in BCD. LSD and LSD + 1. LSD in lower nibble)

Byte 40 - Number of machines not communicating (Expressed in BCD. LSD and MSD. LSD in lower nibble)

(“0000” (i.e. byte 39 = byte 40 = 00) is the static value to be reported in PDB1 bytes 39 – 40 by the iSAP-GM.)

 Byte 41 - 00 (Spare)

 Byte 42 - Current Jackpot Value Level 1. Amount expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.

Byte 43 - Dollar LSD and LSD+1. LSD in lower nibble.

.
 Byte 46 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

Example: Current Jackpot = \$5432.98

Byte XX = 98
 Byte XX+1 = 32
 Byte XX+2 = 54
 Byte XX+3 = 00
 Byte XX+4 = 00

 Byte 47 - Current Jackpot Value Level 2 expressed in dollars and cents (format as in Jackpot Level 1)

Byte 51

Byte 52 - Current Jackpot Value Level 3 expressed in dollars and cents (format as in Jackpot Level 1).

.

Byte 56

Byte 57 - Current Jackpot Value Level 4 expressed in dollars and cents (format as in Jackpot Level 1).

.

Byte 61

Byte 62 - Current Jackpot Win Progressive Level (1 = 01, 2 = 02, 3 = 03, 4 = 04, 00 = no win) i.e. first waiting to be reset.

.

Byte 63 - Gaming Machine GMID of Current Jackpot Winning Machine (i.e. first waiting to be reset). Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 64 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 65 - GMID of Current Jackpot Winning Machine (first waiting to be reset) LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

(The GMID reported in these bytes will be the same as that (the GMID) reported in each SDB output by the iSAP-GM.)

Byte 66 - Current Jackpot Win (first waiting to be reset). Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.

Byte 67 - (Dollar LSD and LSD+1). LSD in lower nibble.

.

Byte 70 - (Dollar LSD+6 and MSD). LSD+6 in lower nibble.

Byte 71 - Number of jackpots waiting to be reset/cleared (Expressed in BCD. LSD and MSD. LSD in lower nibble)

Byte 72 - Program identification 1. 8 ASCII characters. Most significant character in byte 79, unused bytes to contain spaces (20H).

.

Byte 79

Byte 80 - Program identification 2. Format as for program ID 1. Spaces (20H) if not required.

.

Byte 87

Byte 88 - Program identification 3. Format as for program ID 1. Spaces (20H) if not required.

.

Byte 95

Byte 96 - Program Identification 4. Format as for program ID 1. Spaces (20H) if not required.

.

Byte 103

Byte 104 - 00 (spare)

.

Byte 126 - 00 (spare)

Byte 127 - Checksum lower nibble (0XH)

Byte 128 - Checksum upper nibble (X0H)

In the PDB1, the data is to be interpreted by the receiver depending on its position in the data stream. A time period of approximately 140 ms is required to complete the transmission of 128 bytes (9600 bps).

The gaming machine must ensure that the transmission of PDB1 is completed in a period not exceeding 200 mS. The timing will commence with the transmission of the first bit.

Progressive Data Block 2 Structure

Information is to be encoded in packed BCD format except in cases where it is impractical to apply this technique e.g. Start of block.

Byte 1 (FF) - Start of block (this is a unique code and must not appear anywhere else in the block)

Byte 2 (11) - PDB2 Identifier

Byte 3 (02) - PDB 2 Data block version number LSD and LSD+1. LSD in lower nibble.

Byte 4 (01) - PDB 2 Data block version number LSD+2 and MSD. LSD+2 in lower nibble.

Byte 5 (XX) - Block Sequence Counter

This counter is incremented dependent of the PDB1. Each PDB1-PDB2 pair must have the same block sequence counter. Any change of information in PDB1 data block causing the PDB1 block sequence counter to increment will automatically cause the PDB2 block sequence counter to increment to the same value.

Byte 6 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - "Spare" currently this byte will contain 00H

Byte 10 - "Spare" currently this byte will contain 00H

Byte 11 - Reset Value Level 1. Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.

Byte 12 - Dollar LSD and LSD+1. LSD in lower nibble.

Byte 15 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

Example: Reset value = \$5432.00

Byte XX = 00
 Byte XX+1 = 32
 Byte XX+2 = 54
 Byte XX+3 = 00
 Byte XX+4 = 00

 Byte 16 - Reset Value Level 2 (format as for level 1)

.
 Byte 20

 Byte 21 - Reset Value Level 3 (format as for level 1)

.
 Byte 25

 Byte 26 - Reset Value Level 4 (format as for level 1)

.
 Byte 30 -

 Byte 31 - Jackpot Limit Level 1. Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.

Byte 32 - Dollar LSD and LSD+1. LSD in lower nibble.

.
 Byte 35 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

 Byte 36 - Jackpot Limit Level 2 (format as for level 1)

.
 Byte 40

 Byte 41 - Jackpot Limit Level 3 (format as for level 1)

.
 Byte 45

 Byte 46 - Jackpot Limit Level 4 (format as for level 1)

.
 Byte 50

 Byte 51 - Accrued Hidden Jackpot Value Level 1. Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.

Byte 52 - Dollar LSD and LSD+1. LSD in lower nibble.

.
 Byte 55 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

 Byte 56 - Accrued Hidden Jackpot Value Level 2 (format as for level 1)

.
 Byte 60

 Byte 61 - Accrued Hidden Jackpot Value Level 3 (format as for level 1)

.
 Byte 65

 Byte 66 - Accrued Hidden Jackpot Value Level 4 (format as for level 1)

.
 Byte 70

Byte 71 - Number of jackpot resets Level 1 (LSD and LSD+1). LSD in lower nibble.
 .
 Byte 75 - Number of jackpot resets Level 1 (LSD+8 and MSD). LSD+8 in lower nibble.

 Byte 76 - Number of jackpot resets Level 2 (LSD and LSD+1). LSD in lower nibble.
 .
 Byte 80 - Number of jackpot resets Level 2 (LSD+8 and MSD). LSD+8 in lower nibble.

 Byte 81 - Number of jackpot resets Level 3 (LSD and LSD+1). LSD in lower nibble.
 .
 Byte 85 - Number of jackpot resets Level 3 (LSD+8 and MSD). LSD+8 in lower nibble.

 Byte 86 - Number of jackpot resets Level 4 (LSD and LSD+1). LSD in lower nibble.
 .
 Byte 90 - Number of jackpot resets Level 4 (LSD+8 and MSD). LSD+8 in lower nibble.

 Byte 91 - Total value of all jackpots won Level 1. Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.
 .
 Byte 92 - Dollar LSD and LSD+1. LSD in lower nibble.
 .
 Byte 95 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

 Byte 96 - Total value of all jackpots won Level 2 (format as for level 1)
 .
 Byte 100

 Byte 101 - Total value of all jackpots won Level 3 (format as for level 1).
 .
 Byte 105

 Byte 106 - Total value of all jackpots won Level 4 (format as for level 1)
 .
 Byte 110

 Byte 111 - Total turnover accumulated by iSAP-GM since Start Up. Amount Expressed in Dollars and Cents (Cents LSD and LSD+1). LSD in lower nibble.
 .
 Byte 112 - Dollar LSD and LSD+1. LSD in lower nibble.
 .
 Byte 115 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

 Byte 116 - Total turnover accumulated by iSAP-GM since the last configuration change. Amount expressed in dollars and cents (cents LSD and LSD+1). LSD in lower nibble.
 .
 Byte 117 - Dollar LSD and LSD+1. LSD in lower nibble.
 .
 Byte 120 - Dollar LSD+6 and MSD. LSD+6 in lower nibble.

 Byte 121 - 126 - 00 (spare)

 Byte 127 - Checksum lower nibble (0XH)

Byte 128 - Checksum upper nibble (X0H)

In the PDB2, the data is to be interpreted by the receiver depending on its position in the data stream. A time period of approximately 140 ms is required to complete the transmission of 128 bytes (9600 bps).

The gaming machine must ensure that the transmission of each PDB2 is completed in a period not exceeding 200 mS. The timing will commence with the transmission of the first bit.

Manufacturer Data Block (MDB)

- 6.5.5 The MDB can be transmitted by ports P1, P2, P3, P4, P5 and P6 of the gaming machine according to the requirements of the manufacturer. However, the transmission of the MDB must not interfere with the transmission of the data blocks like SDB, FDB and PDB1-PDB2 pair as outlined by the transmission requirements in this standard.

The transmission of the MDB or its contents, must not compromise or have the potential to compromise the security of the gaming machine in any way.

The Authority must be supplied with complete details of the block for evaluation, and approval must be given prior to use.

There must be a minimum of 20ms delay in transmission between any two consecutive data blocks, with the exception of CCCE transfer commands.

Manufacturer Data Block Structure

Byte 1 (FF) - Start of block (unique byte)

Byte 2 (22) - MDB Identifier

Byte 3 (XX) - Manufacturer ID. (unique number, as advised by the Authority)

Byte 4 (00) - Spare

Byte 5 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 6 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 7 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 8 - Manufacturer Specific

. (FF is not to be used in any of these bytes)

Byte 126 - Manufacturer Specific

Byte 127 - Checksum lower nibble (0XH)

Byte 128 - Checksum upper nibble (X0H)

In the MDB, the data is to be interpreted by the receiver depending on its position in the data stream. A time period of approximately 140 ms is required to complete the transmission of 128 bytes (9600 bps).

The gaming machine must ensure that the transmission of each MDB is completed in a period not exceeding 200 mS. The timing will commence with the transmission of the first bit.

6.5.6 Deleted.

6.6 Extended CCCE Functions

With the exception of the 'Time and Date', 'Machine Lockup' and 'Message Display' command classes, the implementation of the extended CCCE functions is optional. When the extended CCCE commands are utilised, not all the extended CCCE functions need to be implemented, and each of the implemented CCCE commands must satisfy the defined requirements in the standard.

For example, if the game only supports a mystery linked progressive component, then it is not necessary to also implement the standard linked progressive extended CCCE command structure. However, using the above example, the command classes to enable a CCCE money transfer from the mystery link controller to the gaming machine will need to support the CCCE mystery progressive win payment in cents as well as the CCCE Increment in credits.

Note: The gaming machine must be able to accept and process the 'Time and Date' command class 4, 'Machine Lockup' command set 50H-52H and 'Message Display' command set 53H-54H regardless of whether it is enabled to support any other CCCE command classes or not.

When being activated to support the CCCE money transfer function, the gaming machine must also be able to accept and process the CCCE 'Increment In Credits' command class A and 'Decrement In Credits' command class B. For details of the method that can be used to activate CCCE command classes, refer to section 6.2.1 of this standard.

Apart from the CCCE command classes 5, 8, 9, A, B, C, D and E that are listed in this standard, the requirements for other CCCE command classes (classes 4, 6 and 7) are defined in the TITO technical standard.

CCCE Command Classes Overview

6.6.1 Apart from the extended CCCE command classes defined in the TITO technical standard, the gaming machine is permitted to recognise and process the following CCCE commands:

(a) Progressive Win Notification

Class 8	1000	With CCCE Pay	80H - 8FH
Class 9	1001	With Handpay	90H – 9FH

(b) CCCE Increment/Decrement

Class A	1010	CCCE Increment	A1H – A2H	Credits
		CCCE Increment	A3H – A4H	Cents
Class B	1011	CCCE Decrement	B1H – B2H	Credits
		CCCE Decrement	B3H – B4H	Cents

Note: See section 6.3.2 in the NSW Communication Protocol for existing CCCE transfer requirements in Credits.

(c) Progressive Win Payment

Class C 1100 Standard Progressive Win Payment C0H – CFH

Class D 1101 Mystery Progressive Win Payment D0H – DFH

Note: The intent here is that a gaming machine can be connected to, and accept CCCE transfers from 3 points: a Mystery progressive jackpot controller, a Standard progressive jackpot controller and a cashless system.

(d) Progressive Game Current Pool Value

Class E 1110 Progressive Game Pool Value E0H – EFH

(e) Machine Lockup / Message Display / Program Signature Request / Non-Cash Prize Ticket Print

Class 5 0101

Machine Lockup (or Soft Lockup) 50H – 52H

Requirements of the Machine Lockup command set 50H – 52H are defined in sections 6.7 to 6.7.5 of this standard.

Message Display 53H – 54H

Requirements of the Message Display command set 53H – 54H are defined in sections 6.8 to 6.8.4 of this standard.

Program Signature Request 56H – 57H

Requirements of the Program Signature Request command set 56H – 57H are defined in sections 6.9 to 6.9.5 of this standard.

Non-Cash Prize Ticket Print 59H – 5AH

Requirements of the Non-Cash Prize Ticket Print command set 59H – 5AH are defined in sections 6.10 to 6.10.4 to of this standard.

CCCE commands class A (A1 & A2 / A3 & A4) and class B (B1 & B2 / B3 & B4) are generated by a CCCE system. They can only be executed during gaming machine “Idle, no lockup” status condition – “Type 1”.

CCCE command class B (B1 & B2 / B3 & B4) can also be executed during “Idle, P2 – P6 SEF lockup” – “Type 2” gaming machine status condition.

CCCE command classes 8, 9, C, D and E are generated by a progressive game link controller.

CCCE command class E is a global broadcast command to all gaming machines attached to the progressive game link controller. It must be able to be processed by a gaming machine at any time.

CCCE command class C, Standard Progressive Win Payment, can only be accepted by the gaming machine in the “Game cycle, standard progressive win - Type 3” status condition for progressive win awarded by a standard progressive game controller.

CCCE command class D, Mystery Progressive Win Payment, can only be accepted by a gaming machine in "Idle, mystery progressive win - Type 4" status condition for progressive win awarded by a mystery progressive game controller.

Gaming Machine Status Requirements for CCCE Cents Transfer and Progressive Win CCCE Transfer

6.6.2 The Gaming Machine Status requirements for CCCE Cents Transfer and Progressive Win CCCE Transfer are:

- (a) Gaming machine may accept Cents input from a CCCE system via CCCE Increment command class A3 – A4, only if in "Idle, No lockup" status condition, "Type 1", as defined below.

Type 1 - status condition for CCCE Increment and CCCE Decrement commands

Status byte 1

- #0 = 1 (Idle)
- #1 = 0 (Game Cycle)
- #2 = X (Power up)
- #3 = X (Reset)
- #4 = X (CCCE transfer completed) *Master Flag*
- #5 = 0 (Spare)
- #6 = 0 (Spare)
- #7 = 0 (Spare)

Status byte 2 = 00

Status byte 3 = 00

Status byte 4 = 00

Status byte 5 = 00

- (b) Gaming machine may transfer Cents to a CCCE system via CCCE Decrement command class B3 – B4 when in "Idle, No lockup" status condition, "Type 1". Cents decrement to a CCCE system can also be performed during "Idle, Subsidiary Equipment Suspended" status condition, "Type 2", as defined below.

Type 2 - status condition for CCCE Decrement command

Status byte 1

- #0 = 1 (Idle)
- #1 = 0 (Game Cycle)
- #2 = X (Power up)
- #3 = X (Program Reset)
- #4 = X (CCCE transfer completed) *Master Flag*
- #5 = 0 (Spare)
- #6 = 0 (Spare)
- #7 = 0

Status byte 2 = 00

Status byte 3 = 00

Status byte 4

- #0 = 0 (Audit Mode)
- #1 = 0 (Test Mode)
- #2 = 0 (Power Save condition)
- #3 = 1 (Subsidiary Equipment play suspend)
- #4 = 0 (Mechanical Meter disconnect)
- #5 = 0 (Manufacturer Specific Error)
- #6 = 0 (Manufacturer Specific Error)
- #7 = 0

Status byte 5 = 00

- (c) Gaming machine may accept progressive win increment from a standard progressive game controller, via CCCE command class C0H – CFH only if in “Game Cycle, Standard Progressive Win” status condition, “Type 3”, as defined below.

Type 3 – status condition for Progressive Win CCCE Payment

Status byte 1

- #0 = 0 (Idle)
- #1 = 1 (Game Cycle)
- #2 = X (Power up)
- #3 = X (Reset)
- #4 = X (CCCE transfer completed) *Master Flag*
- #5 = 0 (Spare)
- #6 = 0 (Spare)
- #7 = 0

Status byte 2

- #0 = X (Large Win or Large Accumulated Credit)
- #1 = 0 (Collect Cash)
- #2 = 0 (Cancel Credit – Book Pay)
- #3 = 1 (Progressive Win) 0 if unused
- #4 = X (Manufacturer Specific Win) 0 if unused
- #5 = X (Manufacturer Specific Win) 0 if unused
- #6 = X (Manufacturer Specific Win) 0 if unused
- #7 = 0

Status byte 3 = 00

Status byte 4 = 00

Status byte 5 = 00

While the gaming machine is in “Game Cycle, Standard Progressive Win” status condition, the Standard Progressive Win CCCE Payment is the only CCCE Cents increment command that gaming machine may accept.

The gaming machine will remain in “Game Cycle, Standard Progressive Win” status condition until the progressive win CCCE transfer is completed or an appropriate “key off” condition is detected.

Note: The controller is to leave a win notification message up for a reasonable amount of time. Any delay by the controller in transferring a progressive win payment (under \$3,000) to the winning gaming machine beyond 25 seconds is considered unreasonable. The gaming machine is to provide an option to clear a jackpot win after 30 seconds if it did not receive a win payment.

- (d) The gaming machine may accept progressive win increment from a mystery progressive game controller, via CCCE command class D0H – DFH only if in “Idle, Mystery Progressive Win” status condition – “Type 4”, as defined below.

Type 4 – status condition for Mystery Progressive Win CCCE Payment

Status byte 1

#0 = 1 (Idle)
#1 = 0 (Game Cycle)
#2 = X (Power up)
#3 = X (Reset)
#4 = X (CCCE transfer completed) *Master Flag*
#5 = 0 (Spare)
#6 = 0 (Spare)
#7 = 0

Status byte 2

#0 = X (Large Win or Large Accumulated Credit)
#1 = 0 (Collect Cash)
#2 = 0 (Cancel Credit – Book Pay)
#3 = 0 (Standard Progressive Win)
#4 = X (Manufacturer Specific Win) 0 if unused
#5 = X (Manufacturer Specific Win) 0 if unused
#6 = X (Manufacturer Specific Win) 0 if unused
#7 = 0

Status byte 3 = 00

Status byte 4 = 00

Status byte 5 = 00

While the gaming machine is in “Idle, Mystery Progressive Win” status condition, the Mystery Progressive Win CCCE Payment is the only CCCE Cents increment command the gaming machine may accept.

The gaming machine will remain in “Idle, Mystery Progressive Win” status condition until the Mystery Progressive win CCCE transfer is completed or an appropriate “key off” condition is detected.

Note: The controller is to leave a win notification message up for a reasonable amount of time. Any delay by the controller in transferring a progressive win payment (under \$3,000) to the winning gaming machine beyond 25 seconds is considered unreasonable. The gaming machine is to provide an option to clear a jackpot win after 30 seconds if it did not receive a win payment.

Other requirements

6.6.3

(a) Transaction Time Limits, Timeout

A money transfer may only commence after the gaming machine transmits the last byte of a SDB and an appropriate status condition applies.

The gaming machine will assume a timeout has taken place if a complete "phase 1 transfer request" or a "phase 2 transfer request" is not received within 40ms from the time that the last byte of the SDB or the "phase 1 echo" is transmitted.

The gaming machine will respond to a legitimate "phase 1 transfer request" by completing the transmission of the "phase 1 echo" within 40ms of receiving the last byte of the "phase 1 transfer request".

The gaming machine will respond to a legitimate "phase 2 transfer request" by completing the transmission of the SDB within 200 ms of receiving the last byte of the "phase 2 transfer request".

When a successful CCCE transfer occurs the gaming machine must output (if applicable) PERIPHERAL MDB by port P1 with an appropriate payment completed flag set to 1 in byte 96 within 460 ms of a legitimate last byte of the "phase 2 transfer" request.

Note: The host CCCE device will automatically timeout after 4 attempts (or 10 minutes) whichever occurs first.

(b) SDB CCCE transfer Completed Flag (Master Flag)

The SDB status byte 1, 'CCCE transfer completed' flag (byte 9 bit #4), is considered to be a master CCCE transfer completed flag. Whenever any CCCE increment / decrement in credits or cents, Standard Progressive win payment or Mystery win payment transfer is successful the master flag 'CCCE transfer completed' in SDB must be set to 1 together (if applicable) with an appropriate payment completed flag in Peripheral MDB byte 96.

Whenever the master flag 'CCCE transfer completed' in SDB is cleared all three payments completed flags in peripheral MDB byte 96 must be cleared.

(c) SDB Money In meter (Master Money In meter)

The Money In meter in SDB (byte 42 to 46) is considered to be a master Money In meter. The Master Money In meter in SDB will be incremented for any type of successful money transfer in. However if successful CCCE transfer in cents or successful Standard Progressive jackpot win transfer in cents or successful Mystery jackpot win transfer in cents in addition to the Master Money In meter in SDB is incremented an appropriate Payment in or Transfer In meter in Peripheral MDB between byte 100 to 114 is to be incremented.

If the CCCE transfer in is done using the credit (type A1, A2) method the Total amount for CCCE Transfer In with cents meter in Peripheral must not be incremented.

(d) CCCEID - Centralised Cash Control Equipment Identification

CCCEID is a six-digit number, which will be used to identify the Centralised Cash control equipment for the purpose of CCCE transaction from the cashless systems.

(e) PCID - Progressive Controller Identification Number

PCID is a six-digit number, which will be used to identify the progressive controller for the purpose of CCCE transaction from jackpot controller.

PCID number will be same as the allocated GMID number of the controller.

(f) Restarting a Transaction Command

Once the gaming machine terminates a transaction, a new transaction may only recommence when the gaming machine transmits the next SDB, thereby opening another "window" for the transfer of money.

If the status of the gaming machine changes during the transaction, the gaming machine will cease the normal sequence and output a SDB that reflects the changed status.

If the change in status occurs during a transmission, the transmission must be completed before the SDB is output. Note that only that portion of the transaction currently being transmitted needs to be completed not the entire transaction.

(g) Sequence Number Definition

The Sequence Number (SN) will be a two byte BCD number ranging from 1 to 9999. The first sequence number will be 1 and it will roll over from 9999 to 1. The SN will be a number incremented by the CCCE host on each new transaction (including Cancel Credits).

(h) Sequence Number Validation for accepting CCCE command

As soon as any CCCE payment is accepted by a gaming machine, the gaming machine will store the Host Identification number, Sequence Number, Amount and Time of receipt in the memory.

The gaming machine will have the capability to store at least the last CCCE transfer information (Host Identification number, Sequence Number, Amount and Time) received from each of the host devices (Mystery link controller, Standard link controller, and Cashless system).

When the gaming machine receives any further CCCE transfers, it will validate it with the last valid CCCE transfer previously received from the relevant host. If at least one of Host Identification, Sequence Number or Amount is different from the previous transfer information it will be treated as a valid CCCE transfer and the gaming machine will process it as normal.

However, if all three items match, the gaming machine will compare the time associated with the previous CCCE transfer and will accept the current transfer as long as the time difference between the previous transfer and the current transfer is more than 20 minutes. This will enable the gaming machine to operate correctly if a valid CCCE payment with the same transaction information was received in situations such as a Rapid Hit Jackpot System where the sequence number can be repeated within a short period of time.

CCCE Command Classes

6.6.4

(a) CCCE Command Class 8 and 9: Progressive Win Notification

The Progressive Win Notification command classes 8 and 9 are utilised to effectively notify the gaming machine of a progressive game win from the progressive game controller. Each Progressive Win Notification command class defines a standard function and must be interpreted by the gaming machine at any time (i.e. during idle, game cycle and lockup states) – except in memory error state and self audit error.

Note for Mystery Progressives:

When a gaming machine receives a progressive win notification, it must echo the win notification message and then output PERIPHERAL MDB with mystery progressive win flag set to 1 in byte 96 bit #3.

After win notification the machine must complete the current game cycle and enter idle mode and output at least one SDB. And then wait for win payment. See CCCE Command class D for Mystery Progressive Win Payment.

The status condition is cleared when appropriate payment received or "key off" procedure is executed and the gaming machine must output PERIPHERAL MDB with mystery progressive win flag set to 0 in byte 96 bit #3.

Note for controllers:

If a Progressive Win Notification command has been attempted a maximum of three times, and failed, the progressive game controller shall generate a new command class for manual progressive win payment.

An attempt is defined as a CCCE transaction sequence being transmitted after the last byte of an SDB has been received. The attempt timeout will occur after 3 SDBs (approx. 4.5 seconds).

- ◆ The command information flow is presented below.

Gaming Machine	Progressive Win Notification	Progressive Game Controller
	-----<-----	Command GMID PCID SN Amount (cents) Checksum
Command (Echo) GMID PCID SN Amount (Echo) Checksum	----->-----	

A complete Progressive Win Notification transmission packet must be completed within 30ms by the Progressive Game Controller and the gaming machine.

The gaming machine must respond from the last byte of a legitimate Progressive Win Notification command, with a complete echo within 200ms.

A legitimate Progressive Win Notification request is defined as a communication packet that meets the correct structure (defined below) and has the correct GMID for the gaming machine to respond. If any section of the request is incorrect, the gaming machine must not respond with an echo and must not display any progressive win information.

- ◆ The following table outlines the determination of the Command Class value.

Note: This protocol supports 4 standard and 4 mystery levels (on 2 separate link controllers), or 8 mystery levels (on 2 link controllers).

Progressive Win Notification Command Class Bit Specification							
7,6,5	Function	4	Function	3	Function	2,1,0	Function
1,0,0	Progressive Win Notification Mask	0	CCCE Transfer	0	Standard Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,0,0	Progressive Win Notification Mask	0	CCCE Transfer	1	Mystery Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,0,0	Progressive Win Notification Mask	1	Handpay	0	Standard Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,0,0	Progressive Win Notification Mask	1	Handpay	1	Mystery Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8

- ◆ The structure of the Progressive Win Notification Command is defined as follows:

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (80H - 9FH)
#2,1,0 - X,X,X - Progressive Game Win Level
#3 - X - Type of Progressive Game
#4 - X - CCCE or Handpay
#5 - 0 - Progressive Win Notification mask
#6 - 0 - Progressive Win Notification mask
#7 - 1 - Progressive Win Notification mask

Byte 3 - GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - PCID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - PCID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - PCID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

- ◆ Progressive Game Win Notification (Echo) structure is specified below.

Byte 1 -FF - Start of block (unique)

Byte 2	-Command Class (80H - 9FH)
#2,1,0	-X, X, X - Progressive Game Win Level
#3	-X - Type of Progressive Game
#4	-X - CCCE or Handpay
#5	-0 - Progressive Win Notification mask
#6	-0 - Progressive Win Notification mask
#7	-1 - Progressive Win Notification mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 -PCID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 -PCID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 -PCID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

(b) CCCE Increment and Decrement in Cents (A3, A4, B3, B4 Commands)

The CCCE Increment and Decrement commands are utilised to effectively transfer money from the CCCE system into the gaming machine and from the gaming machine to the CCCE system. Each command class defines a standard function and can only be executed if the required status type condition applies. The money is transferred in cents.

- ◆ The command information flow is presented below.

Gaming Machine	CCCE Increment CCCE Decrement	CCCE Interface
Standard Data Block	----->-----	
	-----<-----	Command (Ph.1) GMID CCCEID SN Amount (in cents) Checksum
(Echo) Command (Ph.1) GMID CCCEID SN Amount Checksum (Note 1)	----->-----	
	-----<-----	Command (Ph.2) GMID CCCEID SN Amount (in cents) Checksum
Standard Data Block (Note 2)	----->-----	

Note 1: If an error in transmission occurs or if the CCCE transfer limit is exceeded then the gaming machine will not respond and the transaction will be terminated.

Note 2: A successful transaction is determined by transmission of an SDB with the Credit and Money In meters reflecting the appropriate increments as well as CCCE transfer completed status bit of the SDB set reflecting a successful transfer.

If an error occurs or the phase 1 and phase 2 amounts differ then the gaming machine will not respond and the transaction will be terminated.

If a Timeout (see above) occurs the gaming machine will terminate the transaction.

- ◆ The following table outlines the determination of the Command Class value.

CCCE Increment / Decrement Command Class Bit Specification							
7,6,5	Function	4	Function	3	Function	2,1,0	Function
1,0,1	CCCE Increment Mask	0	CCCE Increment	0	CCCE Mask	0,1,1 1,0,0	CCCE Transaction Phase 1 CCCE Transaction Phase 2
1,0,1	CCCE Decrement Mask	1	CCCE Decrement	0	CCCE Mask	0,1,1 1,0,0	CCCE Transaction Phase 1 CCCE Transaction Phase 2

- ◆ The structure of the CCCE Increment/Decrement Command is defined as follows:

Byte 1 -FF - Start of block (unique)

 Byte 2 -Command Class (A3H – A4H, B3H – B4H)
 #0,1,2 -X, X, X - CCCE Phase 1 or Phase 2
 #3 -0 - CCCE Mask
 #4 -X - Type of transfer
 #5 -1 - CCCE Mask
 #6 -0 - CCCE Mask
 #7 -1 - CCCE Mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 6 -CCCEID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 -CCCEID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 -CCCEID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

 Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

 Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

- ◆ The structure of the CCCE Increment/Decrement (Echo) Command from gaming machine to CCCE system is defined as follows:

Byte 1 -FF - Start of block (unique)

 Byte 2 -Command Class (A3H, B3H)
 #0,1,2 -1, 1, 0 - CCCE Phase 1 Echo
 #3 -0 - CCCE Mask
 #4 -X - Type of transfer
 #5 -1 - CCCE Mask
 #6 -0 - CCCE Mask
 #7 -1 - CCCE Mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 6 -CCCEID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 -CCCEID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 -CCCEID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

 Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

 Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

(c) CCCE Command Class C and D: Progressive Win Payment

The Progressive Win Payment commands are utilised to effectively transfer money of the progressive win value generated by a progressive game controller into the gaming machine. Each command class defines a standard function and can only be executed if the required status type condition applies.

The Jackpot win payment should be round up by the controller to the nearest base credit value of the gaming machine.

- ◆ The command information flow is presented below.

Gaming Machine	Progressive Win Payment	Progressive Game Controller
Standard Data Block	----->-----	
	-----<-----	Command (Ph.1) GMID PCID SN Amount (Cents) Checksum
(Echo) Command (Ph.1) GMID PCID SN Amount (Cents) Checksum (Note 1)	----->-----	
	-----<-----	Command (Ph.2) GMID PCID SN Amount (Cents) Checksum
Standard Data Block (Note 2)	----->-----	

- ◆ The following table outlines the determination of the Command Class value.

Progressive Win Payment Command Class Bit Specification							
7,6,5	Function	4	Function	3	Function	2,1,0	Function
1,1,0	Progressive Win Payment Mask	0	Standard Progressive	0	Phase 1	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,1,0	Progressive Win Payment Mask	0	Standard Progressive	1	Phase 2	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8

1,1,0	Progressive Win Pay Mask	1	Mystery Progressive	0	Phase 1	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,1,0	Progressive Win Payment Mask	1	Mystery Progressive	1	Phase 2	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8

- ◆ The structure of Progressive Win Payment command is defined as follows:

Byte 1 -FF - Start of block (unique)

Byte 2 -Command Class (C0H - DFH)

#2,1,0 -X, X, X - Progressive Win Level

#3 -X - Phase 1 or Phase 2

#4 -X - Type of Progressive Game

#5 -0 - Progressive Win Payment Mask

#6 -1 - Progressive Win Payment Mask

#7 -1 - Progressive Win Payment Mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 -PCID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 -PCID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 -PCID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

- ◆ The structure of Progressive Win Payment (Echo) command from gaming machine to progressive game controller is defined below:

Byte 1 -FF - Start of block (unique)

Byte 2 -Command (C0H - C7H, D0H - D7H)

#2,1,0 -X, X, X - Progressive Win Level

#3 -0 - Phase 1 Echo

#4 -X - Type of Progressive Game

#5 -0 - Progressive Win Payment Mask

#6 -1 - Progressive Win Payment Mask

#7 -1 - Progressive Win Payment Mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 -PCID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 7 -PCID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 -PCID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble.

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7. LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD. LSD+8 in lower nibble

Byte 16 -Checksum lower nibble (0XH)

Byte 17 -Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

(d) CCCE Command Class E: Progressive Pool Value

Progressive game controller shall use the Progressive Pool Value command to transmit the current Progressive Game Pool Value to all attached gaming machines.

Progressive Pool Value command does not require an echo from the gaming machine.

- ◆ The Progressive Pool Value command information flow is presented below:

Gaming Machine	Progressive Game Pool Value	Progressive Game Controller
	-----<-----	Command Code GMID PCID SN Amount Checksum

If GMID is specified as zero, the gaming machine shall still accept the command as a global broadcast command.

If GMID is specified as non zero, only the gaming machine with the matching GMID should process the command.

If the Progressive pool value is not updated for more than 2 minutes from the controller the gaming machine should display an appropriated message (if applicable).

A complete Progressive Pool Value command must be completed within 30ms.

PCID is a Progressive Controller ID.

Note that the Sequence Number reported in the 'Progressive Pool Value' command class E is zero at all times.

- ◆ The following table outlines the determination of the Command Class value.

Progressive Pool Value Command Class Bit Specification					
7,6,5,4	Function	3	Function	2,1,0	Function

1,1,1,0	Progressive Pool Value Mask	0	Standard Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8
1,1,1,0	Progressive Pool Value Mask	1	Mystery Progressive	0,0,0 0,0,1 0,1,0 0,1,1 1,0,0 1,0,1 1,1,0 1,1,1	Jackpot Level 1 Jackpot Level 2 Jackpot Level 3 Jackpot Level 4 Jackpot Level 5 Jackpot Level 6 Jackpot Level 7 Jackpot Level 8

- ◆ The structure of Progressive Pool Value command is defined as follows:

Byte 1 -FF - Start of block (unique)

Byte 2 -Command Class (E0H - EFH)
 #0,1,2 -X,X,X - Progressive Pool Level
 #3 -X - Type of Progressive Game
 #4 -0 - Progressive Game Pool Value mask
 #5 -1 - Progressive Game Pool Value mask
 #6 -1 - Progressive Game Pool Value mask
 #7 -1 - Progressive Game Pool Value mask

Byte 3 -GMID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble

Byte 4 -GMID LSD+2 and LSD+3. LSD+2 in the lower nibble

Byte 5 -GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble

Byte 6 -PCID Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble

Byte 7 -PCID LSD+2 and LSD+3. LSD+2 in the lower nibble

Byte 8 -PCID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble

Byte 9 -SN Least Significant Digit (LSD) and LSD+1. The LSD is to be in the lower nibble

Byte 10 -SN LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble

Byte 11 -Amount LSD and LSD+1. LSD in lower nibble

Byte 12 -Amount LSD+2 and LSD+3. LSD+2 in lower nibble

Byte 13 -Amount LSD+4 and LSD+5. LSD+4 in lower nibble

Byte 14 -Amount LSD+6 and LSD+7, LSD+6 in lower nibble

Byte 15 -Amount LSD+8 and MSD, LSD+8 in lower nibble

Byte 16 -Checksum lower nibble (0XH)

Byte 17 - Checksum upper nibble (X0H)

The checksum is generated by applying modulo 2 addition to each of the bytes 2 to 15.

(e) **CCCE Command Class 5:**

- Machine Lockup command set 50H – 52H requirements are list in sections 6.7 to 6.7.5 of this standard.
- Message Display command set 53H – 54H requirements are listed in sections 6.8 to 6.8.4 of this standard.
- Program Signature Request command set 56H – 57H requirements are listed in sections 6.9 to 6.9.5 of this standard.
- Non-Cash Prize Ticket Print command set 59H – 5AH requirements are listed in sections 6.10 to 6.10.4 of this standard.

6.7 Machine Lockup Command Set 50H-52H

The Machine Lockup command is primarily used by the host to disable the game play function of a gaming machine for pre-commitment lockups.

It may also be used to clear the display of message due to the Message Display command 53H.

6.7.1 Machine Lockup command information flow is presented below:

Step	Gaming machine	Machine Lockup Command Information	Host
1	Gaming machine transmits SDB	----->-----	
2		-----<-----	Command (50H) GMID Host ID Lockup Reason Code Player ACK Status
3	Machine Lockup Command Acknowledgement	----->-----	Command (51H) GMID Host ID Player ACK Status
4	Machine will enable game play if not detecting any errors	----->-----	Command (52H) GMID Host ID Enable Game Play

6.7.2 Machine Lockup Command – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (50H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 – Game Enable Status

0x00 -> enable game play

0x01 -> remove display of message due to 53H command and enable game play

0x02 -> disable game play and allow collect till the player acknowledgement is received

0x03 -> disable game play and allow collect

0x04 -> disable game play and disallow collect

0x05 -> disable game play for a period of 10 seconds

0x06 -> disable game play for a period of 20 seconds

0x07 -> disable game play for a period of 30 seconds

0x08 -> disable game play for a period of 60 seconds

0x09 -> clear display of message due to 53H command

Byte 10 – Reason for Game Disable

Reason Code (Bit)	Default Message to be Displayed by Gaming Machine
0	Nil
1	Turnover limit reached, Promotions Disabled
2	Loss limit reached, Game Play Disabled
3	Time limit reached, Promotions Disabled
4	Non-Cash Prize Promotional Voucher Awarded
5	Temporarily Unavailable
6	Disabled by Host
7	Always set to Zero

Byte 11 – Player Acknowledgement Status from Gaming Machine

Reason Code (Bit)	Player Acknowledgement Status From Gaming Machine
0	Player Acknowledgement required for soft lockup with no message to display
1	Player Acknowledgement required for Turnover Limit Reached
2	Player Acknowledgement required for Loss Limit Reached
3	Player Acknowledgement required for Time Limit Reached
4	Player Acknowledgement not Required
5	Set to zero; Currently not used
6	Waiting for Player Acknowledgement
7	Always set to Zero

Byte 12 – to Display the Default or Alternative Message

0x00 -> Default Message as listed at Byte 10 above is to be displayed;

0x01 -> Alternative message that is input to the host by the manufacturer is to be displayed instead of the default message.

Byte 13 to Byte 72 (Maximum 60 printable ASCII characters)

Alternative Message Information in Simple ASCII format. Unused locations will be filled with Space (20 Hex) character. Refer to implementation notes on the method of sending this message to the gaming machine.

Byte 73 - Checksum lower nibble (0XH)

Byte 74 - Checksum upper nibble (X0H)

6.7.3 Machine Lockup Command Acknowledgment – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (51H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Player Acknowledgement Status from Gaming Machine

Reason Code (Bit)	Player Acknowledgement Status from Gaming Machine
0	Player Acknowledgement required for lockup with no message to display
1	Player Acknowledgement required for Turnover Limit Reached
2	Player Acknowledgement required for Loss Limit Reached
3	Player Acknowledgement required for Time Limit Reached
4	Player Acknowledgement not Required
5	Set to zero; Currently not used
6	Waiting for Player Acknowledgement
7	Always set to Zero

Byte 10 - Checksum lower nibble (0XH)

Byte 11 - Checksum upper nibble (X0H)

6.7.4 Player Acknowledgement Receipt Command – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (52H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 – Player Acknowledgement Status from Gaming Machine

Reason Code (Bit)	Player Acknowledgement Receipt Status from Gaming Machine
0	Player Acknowledgement received for lockup with no message to display
1	Player Acknowledgement received for Turnover Limit Reached
2	Player Acknowledgement received for Loss Limit Reached
3	Player Acknowledgement received for Time Limit Reached
4, 5, 6	Set to Zero
7	Always set to Zero

Byte 10 -Checksum lower nibble (0XH)

Byte 11 - Checksum upper nibble (X0H)

6.7.5 Machine Lockup Command Implementation Notes

The host must only send a machine lockup command 50H to the gaming machine when it has received a SDB with 'idle' bit set.

As soon as the lockup command 50H is received by the gaming machine, the machine must enter into a soft lockup state. In the soft lockup state, the gaming machine must meet the following:

- Disable any game play;
- Disable any cash and ticket inputs;
- Allow the player to collect their credits;
- Continue to process any CCCE payments (including any jackpot win notification/payments) and any failed SEF signal on communication ports P5-P6 (this is to reduce any player 'walk away' on jackpot wins); and
- Act on any detected errors; and
- Set bit 4 of byte 96 in MDB to indicate it is in this state.

If the gaming machine has commenced a game play by the time the lockup command 50H is received, the machine must enter into a soft lockup state immediately after the completion of the current game play.

Transmission time requirements for machine lockup command class 50H-52H are:

- The host must complete the transmission of a machine lockup command 50H to the gaming machine within 150ms from the time the last byte of a SDB is received or from the time it decides to remove the soft lockup from the gaming machine.
- The gaming machine must respond to a legitimate machine lockup command by completing a machine lockup command acknowledgement 51H within 280ms of receiving the last byte of the machine lockup command 50H.
- The gaming machine must complete the transmission of a player acknowledgement receipt command 52H to the host within 80ms.

If the host did not receive any lockup acknowledgement command from the gaming machine after 80ms from the time the last byte of the machine lockup command was transmitted, the host must attempt to transmit another machine lockup command to the gaming machine within 150ms from the time the next SDB with 'idle bit' set is received.

Below is an illustration of how the machine lockup command class may be used for the pre-commitment implementation with and without acknowledgement from player. Note that these are for illustration purpose only. Contact OLGR to find out which of these methods that may be submitted for approval consideration by the Authority.

Illustration 1: Pre-Commitment Lockup with Player Acknowledgement utilising Machine Lockup Command 50H

When the set loss limit is reached for the first time, assuming that the host wants the gaming machine to display the default message, the host must send a machine lockup command 50H (with Byte 9 set to 02h, bit 2 of Byte 10 set as well as bit 2 of Byte 11 set) to the affected gaming machine after receiving the next SDB with "idle" bit set. This command instructs the gaming machine to display the default loss limit reached message and enter into a soft lockup till the player acknowledgement is received. The gaming machine is now expected to enter into a soft lockup state and display the message '*Loss Limit Reached, Promotions Disabled*'. The gaming machine will also transmit 51H with bits 2 & 6 of Byte 9 in set indicating that it is waiting for player acknowledgement for the loss limit reached. The gaming machine is expected to pop up an option for the player to acknowledge that the set loss limit has been reached as shown in Fig 1. The gaming machine will set bit 4 of Byte 96 in MDB to indicate that the gaming machine has entered the soft lockup state. As soon as the player acknowledgement is detected, the gaming machine will automatically remove the soft lockup and revert to game play status (if there are no other lockup errors detected) and transmit 52H with bit 2 of Byte 9 set indicating that an acknowledgment from the player for

the loss limit reached has been received. The gaming machine will also reset bit 4 of Byte 96 in MDB to indicate that the machine has exited from the soft lockup state.

When the player inserts a card that has already reached the set loss limit and the limit reached has been previously acknowledged by the player as given above, assuming that the host wants the gaming machine to display the default message, the host must send a machine lockup command 50H (with Byte 9 set to 05h, bit 2 of Byte 10 set as well as bit 4 of Byte 11 set) to the affected gaming machine after receiving the next SDB with “idle” bit set. This command informs the gaming machine to display the default loss limit reached message for a period of 10 seconds, enter into a soft lockup while the message is being displayed and remove the loss limit reached message and the machine soft lockup after 10 seconds. While the gaming machine is in the soft lockup state, bit 4 of Byte 96 in MDB will be set to indicate that the gaming machine is in a soft lockup state.

If the host had sent a command to gaming machine to enter into a soft lockup state and is waiting for a player acknowledgement and the host detects the card was removed, then the host must transmit a Game Play Enabling command (a machine lockup command 50H with Byte 9 set to 00h) to the gaming machine. In this case the gaming machine is to ignore the contents of the remaining bytes in this 50H command. The Game Play Enabling command will be used by the gaming machine as an instruction to exit from the soft lockup state even though it has not yet received the player acknowledgement and remove the soft lockup. The host in this case must remember that the player has reached the limit set and has not acknowledged this. The host must also ensure that the gaming machine has exited from the soft lockup by verifying that bit 4 of Byte 96 in MDB is reset. If this MDB bit continues to be set, the host must retransmit this command again.

When the host detects the insertion of a card for which the player has not acknowledged the limit reached message, it must send a machine lockup command 50H (with Byte 9 with a value of 02h, bit 2 of Byte 10 set and bit 2 of Byte 11 set) to the gaming machine after receiving the next SDB with "idle" bit set. This command instructs the gaming machine to display the default loss limit reached message and enter into a soft lockup till the player acknowledgement is received.

If the host wants the gaming machine to display a message '*You Have Reached Your Set Time Limit*' instead of the default message of '*Time limit reached, Promotions Disabled*', it will transmit the following in Byte 13 to 72.

*'20 74 69 6D 69 4C
20 65 6D 69 54 20 74 65 53 20 72 75 6F 59 20 64 65 68 63 61 65 52 20 65 76 61 48 20 75
6F 59'.*

Shown in Fig 1 below is a typical display when the loss limit is reached for the first time and acknowledgement from player is required.



Fig 1

Illustration 2: Pre-Commitment Lockup without Player Acknowledgement utilising Machine Lockup Command 50H

When the set loss limit is reached for the first time, assuming that the host wants the gaming machine to display the message “Loss Limit Reached, Game Play Disabled”, the host must send a machine lockup command 50H (with Byte 9 set to 03h, bit 0 of Byte 10 set, bit 4 of Byte 11 set, Byte 12 set to 01h, and the alternative message) to the gaming machine after receiving the next SDB with “idle” bit set. This command instructs the gaming machine to enter into a soft lockup and display the alternative message. The gaming machine is now expected to enter into a soft lockup state and transmit 51H with bit 4 of Byte 9 in set and display the message eg, ‘*Loss Limit Reached, Game Play Disabled*’ as shown in Fig 2. The gaming machine will set bit 4 of Byte 96 in MDB to indicate that the gaming machine has entered the soft lockup state.

As soon as the player card is removed, the host must transmit a Game Play Enabling command (a machine lockup command 50H with Byte 9 set to 00h) to the gaming machine. In this case the gaming machine is to ignore the contents of the remaining bytes in this 50H command. The Game Play Enabling command will be used by the gaming machine as an instruction to remove the display message and exit the soft lockup state. In this case, the host must also ensure that the gaming machine has exited from the soft lockup by verifying that bit 4 of Byte 96 in MDB is reset. If this MDB bit continues to be set, the host must retransmit this command again.

When the player inserts a card that has already reached the set loss limit during the applicable limit period, the host must send a machine lockup command 50H (with Byte 9 set to 03h, bit 0 of Byte 10 set, bit 4 of Byte 11 set, Byte 12 set to 01h, and the alternative message) to the gaming machine after receiving the next SDB with “idle” bit set. This command instructs the gaming machine to enter into a soft lockup and display the alternative message. The gaming machine is now expected to enter into a soft lockup state and transmit 51H with bit 4 of Byte 9 in set and display the alternative message (eg, “Loss Limit Reached, Game Play Disabled”). The gaming machine will set bit 4 of Byte 96 in MDB to indicate that the gaming machine has entered the soft lockup state.

If the host detects that the gaming machine is in the soft lockup state (bit 4 of Byte 96 in MDB set) and the host decides to remove the display message and the soft lockup from the gaming machine (says due to the applicable limit period was expired), the host must transmit a Game Play Enabling command (a machine lockup command 50H with Byte 9 set to 00h) to the gaming machine. In this case the gaming machine is to ignore the contents of the remaining bytes in this 50H command. The Game Play Enabling command will be used by the gaming machine as an instruction to remove the display message and exit the soft lockup state. In this case, the host must also ensure that the gaming machine has exited from the soft lockup by verifying that bit 4 of Byte 96 in MDB is reset. If this MDB bit continues to be set, the host must retransmit this command again.

Shown in Fig 2 below is a typical display when the loss limit is reached (no acknowledgement from player is required):



Fig 2

6.8 Message Display Command Set 53H-54H

The Message Display command will be used by the host to send various player related information to the gaming machine for displaying them on the main machine display.

An example of using this message display command will be to remind a player of a responsible gambling message like “TAKE A BREAK” displayed on the machine screen. Another example will be to inform a player that he/she has won a specific promotion.

6.8.1 Message Display command information flow is presented below:

Step	Gaming machine	Message Display Command Information	Host
1	Gaming machine transmits SDB	----->-----	
2		-----<-----	Command (53H) GMID Host ID Display Info
3	Acknowledgement with the Packet Sequence Counter	----->-----	Command (54H) GMID Host ID Display Info

6.8.2 Message Display Command – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (53H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.
Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.
Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

Byte 10 - Message Type
0x01 - To be displayed on the middle of the machine screen.
0x02 - To be scrolled through in the bottom of the machine screen.

Byte 11 - Group Identifier Number
The Group Identifier Number will be used to identify the group of messages to be displayed. The host will increment this every time it transmits a new group set to the gaming machine.
If Byte 11 is zero, it indicates that there is only one message and the gaming machine shall ignore the value in Bytes 12 and 13 of the packet.

If Byte 11 is non-zero, it indicates that this packet is carrying a multi-packet message where the message information to be displayed on the main machine screen is grouped by the host into two or more parts. Each part will separately be sent by the host using a message display command.

Byte 12 - Total Number of Message Display Command Packets used for a multi-packet message. This total number is in BCD format.

Byte 13 - Packet Reference Number of a multi-packet message.
The Packet Reference Number is in BCD format and indicates which part of a message is being sent in the current message display command packet.

Byte 14 - Length of Message to be displayed
dd – Message length in BCD format (Maximum length is 99). This field indicates only the length of the message and not the length of the packet. Hence it will be the length of data from Byte 18 to the byte preceding the first byte of the checksum.

Byte 15 - Time to Live
dd - Time to be displayed in seconds in BCD format (Maximum time is 99 seconds). A value of 0 indicates permanent display of the message till it is erased with a blank message.

Byte 16 - Enable Sound
0x00 - No sound
0x01 - Enable sound

Byte 17 - Duration of Sound
dd - Duration of Time the Sound must be played. (Maximum time is 99 seconds)

Byte 18 to Byte 117 (Max)
Message Information in Simple ASCII format; (Supports only printable characters, horizontal tab [09h], new line feed [0Ah] and carriage return [0Dh])

As an example, to transmit the message "TAKE A BREAK" using the message display command,

Byte 18 to Byte 29 of the message display command must correspondingly have "4B 41 45 52 42 20 41 20 45 4B 41 54".

But if Byte 18 to Byte 29 of the message display command correspondingly have "54 41 4B 45 20 41 20 42 52 45 41 4B", it would mean that this message "KAERB A EKAT" will be transmitted.

Byte 118 - Checksum lower nibble (0XH)*
Byte 119 - Checksum upper nibble (X0H)*

** Assuming the message information was 99 bytes long.*

6.8.3 Message Display Command Acknowledgment – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (54H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter (in BCD format).

Byte 10 - Checksum lower nibble (0XH)

Byte 11 - Checksum upper nibble (X0H)

6.8.4 Message Display Command Implementation Notes

The host must only send the Message Display command 53H when it has received a SDB with 'idle' bit set.

If the gaming machine has commenced a game play by the time the display command 53H is received, the machine must still acknowledge the receipt of this packet and display the message.

As soon as the 53H command is received by the gaming machine, it will display the message on the main machine screen. If the host wants to lock the machine and display the message, it will first send the machine lockup command 50H with Byte 9 set to 03h, bit 0 of Byte 10 set and Byte 12 set 00h. This will make the machine to go into a soft lockup state with no message displayed. Once the host decides to remove the lockup, it must send another 50H command with Byte 9 set to 00h to instruct the gaming machine to go to a game play enable status.

The host can either display this message continuously or for a predefined period on the gaming machine by selecting the parameter in Byte 15.

If the enable sound parameter is set to zero (Byte 16 - No Sound), the gaming machine must disregard the value in Byte 17 (Duration of sound). The sound to be played is currently not defined.

The host will set the packet sequence counter (Byte 9) to a value of zero when the gaming machine is installed into the system. It will rollover to a value of 1 once it has reached a value of 99. The packet sequence counter will be incremented every time a new packet is being sent by the host. If the same packet is being repeated, the counter will remain unchanged. The packet sequence counter will be used by the host as a retry mechanism if the message display command acknowledgment was not received from the gaming machine.

The host must complete the transmission of a message display command to the gaming machine within 180ms from the time the last byte of a SDB is received.

The gaming machine must respond to a legitimate message display command by completing a message display command acknowledgement within 280ms of receiving the last byte of the message display command.

While the gaming machine is processing the message display command 53H from the host or is displaying the message, the machine must meet the following:

- Allow the player to collect their credits;
- Continue to process any CCCE payments (including any jackpot win notification/payments) and any failed SEF signal on communication ports P5-P6 (this is to reduce any player 'walk away' on jackpot wins); and
- Act on any detected errors.

An example for the typical transmission of a multi-packet message is given below:

- Let's say that the message information to be displayed on the main machine screen is 120 characters long, and the host divides this message information into 2 parts for transmission. The part 1 contains the first 99 characters of the message information; and the part 2 contains the remaining 21 characters of the message information.
- The host will then send the machine lockup command 50H with Byte 9 set to 03h, bit 0 of Byte 10 set to 1 and Byte 12 set to 00h. This will make the machine to go into a soft lockup state with no message displayed. Host must confirm that the gaming machine has entered the soft lockup state by verifying that bit 4 of Byte 96 in the MDB is set before transmitting the display command.
- The first message display command 53H with Byte 11 set to the current packet sequence counter value, Byte 11 set to 1 (assuming this is the first group set the host is transmitting to the relevant gaming machine), Byte 12 set to 2, Byte 13 set to 1 and the part 1 of the message information will be sent by the host.
- The second message display command 53H with Byte 11 & Byte 12 having the same value as that sent in the first message display command packet, Byte 13 set to 2 and the part 2 of the message information will be sent by the host. Note: Subsequent message display command packets for a multi-packet message may be sent by the host immediately after receiving the previous message display acknowledgement from the gaming machine.
- As soon as technically feasible all the parts representing the multi-packet message information are received by the gaming machine, the machine must display the message on the machine screen. The machine must support a display of a minimum of five messages concurrently.
- Once the host decides to remove display of message and the soft lockup, it must send another 50H command with Byte 9 set to 01h to instruct the gaming machine to remove the display of message and go to a game play enable status.
- Once the host decides to remove the message/s being currently displayed and retain the soft lockup, it must send another 50H command with Byte 9 set to 09h.

The host may use the Clear Displayed Message command (the machine lockup command 50H with Byte 9 set to 09h, bit 0 of Byte 10 set, bit 4 of Byte 11 set and Byte 12 set to 00h) to instruct the gaming machine to remove any message displayed on the main machine screen due to the command 53H.

A typical message to be displayed in the middle of the main machine screen is given below:



Fig 3

A typical message to be displayed in the bottom of the main machine screen is given below:



Fig 4

6.9 Program Signature Request Command Set 56H-57H

The Program Signature Request command will be used by the host to perform signature checking of the gaming machine software program.

6.9.1 Program Signature Request command information flow is presented below:

Step	Gaming machine	Program Hash Request Command Information	Host
1	Gaming machine transmits SDB	----->-----	
2		-----<-----	Command (56H) GMID Host ID Program Hash Seed
3	Program Signature Request Echo	----->-----	Command (56H) GMID Host ID Program Hash Seed
4	Program Signature Result	----->-----	Command (57H) GMID Host ID Program Hash Result

6.9.2 Program Signature Request Command – Packet Structure

Byte 1 - FF - Start of block (unique)

 Byte 2 - Command Class (56H)

 Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

 Byte 10 - 34

Seed Value in Decimal Format Equivalent of 40 Hex value

 Byte 35 - Checksum lower nibble (0XH)

Byte 36 - Checksum upper nibble (X0H)

6.9.3 Program Signature Request Command Echo – Packet Structure

Byte 1 - FF - Start of block (unique)

 Byte 2 - Command Class (56H)

 Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

 Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

Byte 10 - 34

Seed Value in Decimal Format Equivalent of 40 Hex value

Byte 35 - Checksum lower nibble (0XH)

Byte 36 - Checksum upper nibble (X0H)

6.9.4 Program Signature Result Command – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (57H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

Byte 10 - 34

Result Value in Decimal Format Equivalent of 40 byte Hex value

Byte 35 - Checksum lower nibble (0XH)

Byte 36 - Checksum upper nibble (X0H)

6.9.5 Program Signature Request / Result Command Implementation Notes

The host must only send the Program Signature Request command 56H to the gaming machine when it has received a SDB with 'idle' bit set.

If the gaming machine has commenced a game play by the time the Program Signature Request command 56H is received, the machine must still acknowledge the receipt of this packet and commence processing it.

The host must complete the transmission of a Program Signature Request command to the gaming machine within 80ms from the time the last byte of a SDB is received.

The gaming machine must respond to a legitimate Program Signature Request command by completing a Program Signature Request command echo within 280ms of receiving the last byte of the Program Signature Request command.

The gaming machine must complete the transmission of the Program Signature Result command to the host within 10 minutes of receiving the last byte of the Program Signature Request command.

The gaming machine must compute the signature as specified in section 3.10.7 of the National Standard using the signature key given in bytes 10 to 34 of the command class 56H and return the Master result as specified in National Standard to the host.

The host will set the packet sequence counter to a value of zero when the gaming machine is installed into the system. It will rollover to a value of 1 once it has reached a value of 99. The packet sequence counter will be incremented every time a new program signature request is being sent by the host. If the same packet is being repeated, the counter will remain unchanged. The packet sequence counter will be used by the host as a retry mechanism if the program signature request echo was not received from the gaming machine.

If the gaming machine receives a new program signature request with a different sequence number while performing the calculation of the result for a previous program signature request, the machine must abort the current calculation and commence calculation of the result with the new information received in the new command. Typically it will be a different seed value.

In the case where the gaming machine has received the Program Signature Request command 56H from the host or is calculating the program signature requested by the command 56H, the machine must meet the following:

- Allow the player to collect their credits;
- Continue to process any CCCE payments (including any jackpot win notification/payments) and any failed SEF signal on communication ports P5-P6 (this is to reduce any player 'walk away' on jackpot wins);
- Indicate that it is performing this activity by a suitable message at the bottom of the machine's main display; and
- Act on any detected errors.

The gaming machine may also continue other functions like game play activities, etc.

Once the host has detected a failure in the program signature result from the gaming machine due to the received program signature result being different to the expected value, the host shall disable the affected gaming machine.

As an example if the host wants to send the following seed to the gaming machine, it will transmit the information given below in Bytes 10 to 34 in command class 56H:

Signature key (Hex): 64c5 f08e 45f1 5ad7 8031 0ccd 306a e94c c262 64e4

Signature key (Decimal): 5753 1327 9248 1841 2815 7312 6627 9635 6392 1087 9866 5956

Byte 10 to 34 of 56H: 56 59 66 98 87 10 92 63 35 96 27 66 12 73 15 28 41 18 48 92 27 13 53 57

As an example, when computation is complete, the gaming machine will send the Program Signature Result 57H packet with the following contents in Bytes 10 to 34 as given below:

Master Result (Hex): 5aa5 c54f 8622 d7ae a78e c394 249a 3fe9 2535 465a

Master Result (Decimal): 5175 0598 0512 8361 1760 6054 9024 3198 6964 7908 7796 3866

Byte 10 to 34 of 57H: 66 38 96 77 08 79 64 69 98 31 24 90 54 60 60 17 61 83 12 05 98 05 75 51

6.10 Non-Cash Prize Ticket Print Command Set 59H-5AH

The Non-Cash Prize Ticket Print command is only permitted for use by the host (to instruct the gaming machine) to print any type of non-cash prize tickets that cannot be redeemed at the gaming machine.

Printing of any type of cashable tickets is not currently supported by the Non-Cash Prize Ticket Print command.

6.10.1 Non-Cash Prize Ticket Print command information flow is presented below:

Step	Gaming machine	Non-Cash Prize Ticket Printing Information	Host
1	Gaming machine transmits SDB	----->-----	
2		-----<-----	Command (59H) GMID Host ID Printing of non-cash prize ticket
3	Acknowledgement with the Packet Sequence Counter	----->-----	Command (5AH) GMID Host ID Printing of non-cash prize ticket

6.10.2 Non-Cash Prize Ticket Print Command – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (59H)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

Byte 10 - Unique Identifier (Total of 20 Digits) (LSD and LSD+1). LSD in lower nibble

LSD in the lowest nibble. Last 2 bytes will be zero filled.

Byte 19 - (LSD+18 and MSD). LSD+18 in lower nibble.

Byte 20 - 39 - First Message

Byte 40 - 99 - Second Message

Byte 100 - Checksum lower nibble (0XH)

Byte 101 - Checksum upper nibble (X0H)

6.10.3 Non-Cash Prize Ticket Print Command Echo – Packet Structure

Byte 1 - FF - Start of block (unique)

Byte 2 - Command Class (5AH)

Byte 3 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 4 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 5 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 6 - Host ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 7 - LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 8 - LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 9 - Packet Sequence Counter is represented by 2 BCD digits.

Byte 10 - Unique Identifier (Total of 20 Digits) (LSD and LSD+1). LSD in lower nibble
LSD in the lowest nibble. Last 2 bytes will be zero filled.
Byte 19 - (LSD+18 and MSD). LSD+18 in lower nibble.

Byte 20 - Checksum lower nibble (0XH)
Byte 21 - Checksum upper nibble (X0H)

6.10.4 Non-Cash Prize Ticket Print Command Implementation Notes

The host must only send the Non Cash Prize Ticket Print command 59H when it has received a SDB with 'idle' bit set.

The host must complete the transmission of a non cash ticket print command to the gaming machine within 120ms from the time the last byte of a SDB is received.

If the gaming machine has commenced a game play by the time the Non Cash Prize Ticket Print command 59H is received, the machine must still acknowledge the receipt of this packet and commence processing it. It is up to the EGM manufacturers to print this during the middle of a game play or at the end of the current game.

The gaming machine will respond to a legitimate non cash ticket print command by completing a non cash ticket print command echo within 280ms of receiving the last byte of the non cash ticket print command.

The host will set the packet sequence counter to a value of zero when the gaming machine is installed into the system. It will reset the packet sequence counter to a value of 1 once it has reached a value of 99. The packet sequence counter will be incremented every time a new non cash ticket printing packet is being sent by the host. If the same packet is being repeated, the counter will remain unchanged. The packet sequence counter will be used by the host as a retry mechanism if the non cash ticket printing acknowledgment was not received from the gaming machine.

Bit 6 of Byte 15 in MDB will be used to indicate the status of the promotional ticket printing. This bit will be normally set to zero. When a successful promotional ticket printing is completed, the gaming machine will transmit two MDBs with this bit set to one before resetting it back to zero. The host can use this bit to confirm the successful printing of the promotional ticket.

The gaming machine will store the history of a minimum of the last five promotional tickets printed.

If the gaming machine detects any printer error while printing a non cash ticket, it will set the respective bit in Byte 14 in MDB (Bit 1, 3 or 4). However, it will not set any cancel credit error bit in SDB.

A sample promotional ticket is given below:



The printing of promotional ticket is similar to the printing of a hand pay cancel credit ticket.

The field 'Promotional Ticket' will always be printed. A font size of 2 will be used for printing this value (For font sizes, refer to the NSW TITO Technical Standard).

The information in bytes 20 to 39 will be printed below the 'Promotional Ticket' field. In the above example this field contained '\$10 Meal Voucher'. The host will fill the unused locations in this field with a blank symbol (20h). A font size of 4 will be used for printing this value.

The '\$10 MEAL VOUCHER' will be transmitted in Bytes 20 to 39 of command class 59H in the following manner:

20 20 20 20 52 45 48 43 55 4F 56 20 4C 41 45 4D 20 30 31 24

The gaming machine must centralise the message and print it.

The information in bytes 40 to 99 will be printed below the previous message. In the above example this field contained 'Buy 1 meal and receive a second meal for \$10'. The host will fill the unused locations in this field with a blank symbol (20h). A font size of 1 will be used for printing this value.

The host will use the same algorithm to generate the barcode as used by the gaming machine for normal ticket printing. However for the promotional ticket printing, the host will use its own GMID for this calculation. (For details on generating the validation number, refer to section 2.2.6 of NSW TITO Technical Standard.)

While the gaming machine is processing the Non Cash Prize Ticket Print command 59H or is printing a non cash ticket, the machine must meet the following:

- Allow the player to collect their credits;
- Continue to process any CCCE payments (including any jackpot win notification/payments) and any failed SEF signal on communication ports P5-P6 (this is to reduce any player 'walk away' on jackpot wins);
- Indicate that it is performing this activity by a suitable message at the bottom of the main machine display; and
- Act on any detected errors.

The gaming machine may also continue other functions like game play activities, etc.

The gaming machine must have the capability to display full information on the last 5 non-cash prize tickets printed.

6.11 Peripheral Manufacturer Data Block Version 1.06

The Peripheral Manufacturer Data Block is incremented from 1.05 to 1.06 and is updated to support the functionalities offered by the extended CCCE command class 5. However, the gaming machine should continue to have a configuration option to support transmission of MDB according to an earlier version (version 1.01 to 1.05) to support interfacing with legacy data gathering systems. If the gaming machine is configured with an earlier MDB version than 1.05, it will not support any capabilities of the command class 5.

6.11.1 Peripheral Manufacturer Data Block Version 1.06 – Packet Structure:

Information is to be encoded in packed BCD format except in cases where it is impractical to apply this technique e.g. Start of block, all configuration, status & implementation bytes and the unique identifier field of the ticket.

Byte 1 (FF) - Start of block (unique byte)

Byte 2 (22) - Manufacturer Data Block Identifier

Byte 3 - Manufacturer ID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble. - The manufacturer ID will be a unique number allocated by the CLGCA.

Byte 4 (00) - Spare currently "00".

Byte 5 - GMID Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 6 - GMID LSD+2 and LSD+3. LSD+2 in the lower nibble.

Byte 7 - GMID LSD+4 and Most Significant Digit (MSD). LSD+4 in the lower nibble.

Byte 8 (A5) - Data block version number LSD and LSD+1. LSD in lower nibble.

Byte 9 (5A) - Data block version number LSD+2 and MSD. LSD+2 in lower nibble.

Byte 10 (06) - MDB Type Least Significant Digit LSD and LSD+1. The LSD is to be in the lower nibble.

Byte 11 (01) - MDB Type LSD+2 and Most Significant Digit (MSD). LSD+2 in the lower nibble.

_ 0100 Defines the original AGMMA V1.x MDB for the Bill Acceptor

_ 0101 Defines the AGMMA V2.x MDB for Peripheral devices.

_ 0103 Defines the AGMMA V3.x MDB for CCCE protocol extension

_ 0104 Defines the AGMMA V4.10 MDB for TITO Support

_ 0105 Defines the AGMMA V4.18 & GTA 4.20 MDB for TITO Support

_ 0106 Defines the AGMMA GTA 4.20 MDB for TITO Support & Extensions to support Machine Disabling functions

Byte 10 - 06 means version 6 of MDB and Byte 11 - 01 means Peripheral MDB

Byte 12 - Bill Acceptor Status byte 1

#0 - Bill Acceptor Door Open.

#1 - Bill Acceptor Comms Error.

#2 - Bill Acceptor Failure.

#3 - Bill Acceptor Full.

#4 - Bill Acceptor Stacker Removed.

#5 - Bill Acceptor Out of Service.

#6 - Reserved for Future Use.

#7 - Always Zero.

Byte 13 - Bill Acceptor Status byte 2

#0 - Reserved for Future Use.

#1 - Reserved for Future Use.

#2 - Reserved for Future Use.

#3 - Reserved for Future Use.

#4 - Reserved for Future Use.

- #5 - Reserved for Future Use.
- #6 - Reserved for Future Use.
- #7 - Always Zero

Byte 14 - Misc Status byte 1

#0 - Cash Box Drop Door.

#1 - Paper Low

If the printer can detect the paper low then this bit is to be set to '1' when the printer is running low on paper. This is not a lockup condition.

#2 - Printer Valid "Ticket Out" Data

This indicates that a valid "Ticket Out" Data is being transmitted through port P1.

#3 - Printer Fault

This is set to '1' during a printer fault that requires intervention by authorised personnel. This will put the machine into a lockup state and as such the "Cancel Credit Error" flag in the SDB is also required to be set to '1' when a printer fault occurs.

#4 - Printer Paper Out

When the printer detects the paper is out this bit is to be set to '1' and the GAMING MACHINE is to be placed in a lockup state. The Printer Fault bit above is also to be set to '1' as this is a printer fault.

#5 - Printer Valid "Ticket In" Data

'0' means the machine is waiting for response from host for a "Ticket In" request.

'1' means the machine is no longer waiting for any response from the host for a "Ticket In" request.

#6 - Always Zero.

#7 - Always Zero.

Byte 15 - Misc Status byte 2

#0 - "Ticket In" Communication Error.

#1 - "Ticket In" Rejected by host.

#2 - Ten Consecutive Rejects.

#3 - Miscellaneous/Manufacturer specific error in ticket redemption.

#4 - Ticket Rejected - Ticket value less than BCV.

#5 - Ticket Stacking Completed.

#6 - Promotional Ticket Printed.

#7 - Always Zero.

Note: Ticket In Communication Error, Ticket In Rejected by host, Miscellaneous/Manufacturer specific error in ticket redemption & Ticket Rejected - Ticket value less than BCV bits are reset when the next ticket in command (70H) is initiated.

Byte 16 - Number of Bills Inserted \$5 meter (LSD and LSD+1). LSD in lower nibble.

Byte 20 - Number of Bills Inserted \$5 meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 21 - Number of Bills Inserted \$10 meter (LSD and LSD+1). LSD in lower nibble.

Byte 25 - Number of Bills Inserted \$10 meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 26 - Number of Bills Inserted \$20 meter (LSD and LSD+1). LSD in lower nibble.

Byte 30 - Number of Bills Inserted \$20 meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 31 - Number of Bills Inserted \$50 meter (LSD and LSD+1). LSD in lower nibble.

Byte 35 - Number of Bills Inserted \$50 meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 36 - Number of Bills Inserted \$100 meter (LSD and LSD+1). LSD in lower nibble.

Byte 40 - Number of Bills Inserted \$100 meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 41 - Total number of Tickets accepted (LSD and LSD+1). LSD in lower nibble.

Byte 45 - Total number of Tickets accepted (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 46 - Total number of Tickets rejected (LSD and LSD+1). LSD in lower nibble.

Byte 50 - Total number of Tickets rejected (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 51 - Number of Bills Inserted Spare meter 3 (LSD and LSD+1). LSD in lower nibble.

Byte 55 - Number of Bills Inserted Spare meter 3 (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 56 - Total Bills Inserted Dollar Value meter (LSD and LSD+1). LSD in lower nibble.

The Total Bills Inserted Dollar Value meter is in cents.

Byte 60 - Total Bills Inserted Dollar Value meter (LSD+8 and MSD). LSD+8 in lower nibble.

This meter will increment only when bank notes are accepted.

Byte 61 - Total Number of Bills Inserted meter (LSD and LSD+1). LSD in lower nibble.

Byte 66 - Total Number Bills Inserted meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 67 - Date of Ticket Print (LSD and LSD+1). LSD in lower nibble.

Date as appearing on the ticket, in ddmmyyyy format, Eg: 25111997 (25/11/97). LSD in the lowest nibble (ie. LSD in low nibble of byte 67 and LSD+1 in the high nibble. MSD in the high nibble of byte 70)

Byte 70 - (LSD+6 and MSD). LSD+6 in lower nibble.

Byte 71 - Time of the Ticket Print (LSD and LSD+1). LSD in lower nibble.

Time in 24hr format. Eg: 230137 (11:01:37pm). LSD in the lowest nibble. (ie. LSD in low nibble of byte 71 and LSD+1 in the high nibble. MSD in the high nibble of byte 73).

Byte 73 - (LSD+4 and MSD). LSD+4 in lower nibble.

Byte 74 - Unique Identifier (LSD and LSD+1). LSD in lower nibble

Refer to the Printer Unique Identifier section later on the following pages. LSD in the lowest nibble. (ie. LSD in low nibble of byte 74 and LSD+1 in the high nibble. MSD in the high nibble of byte 81). Last 2 bytes will be zero filled.

Byte 83 - (LSD+18 and MSD). LSD+18 in lower nibble.

Byte 84 - Amount of the Ticket in cents (LSD and LSD+1). LSD in lower nibble.

Amount in cents (10 digits, 5 bytes). LSD in the lowest nibble. (ie. LSD in low nibble of byte 84 and LSD+1 in the high nibble. MSD in the high nibble of byte 88).

Byte 88 - (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 89 - Sequential Number of the Ticket (LSD and LSD+1). LSD in lower nibble.

The numerical characters from the sequential ticket number that appears in the ticket (10 digits, 5 bytes). LSD in the lowest nibble. (ie. LSD in low nibble of byte 89 and LSD+1 in the high nibble. MSD in the high nibble of byte 93).

Byte 93 - (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 94 - Machine Configuration byte 1

#0 - Hopper Configured.

If the gaming machine has a hopper installed then this bit is set to 1. This indicates that gaming machine can payout of the hopper, it does not indicate if the hopper is enabled or operational.

#1 - Bill Acceptor Configured.

If the gaming machine has a bill acceptor installed then this bit is set to 1. This indicates that the gaming machine can accept bills, it does not indicate if the bill acceptor is enabled and or operational.

#2 - Printer Configured.

If the gaming machine has a printer installed then this bit is set to 1. This indicates that the gaming machine can print various tickets, it does not indicate if the printer is enabled and or operational.

#3 - Reserved for Future Use.

#4 - Reserved for Future Use.

#5 - Reserved for Future Use.

#6 - Reserved for Future Use.

#7 - Always Zero.

Byte 95 - Machine Configuration byte 2

#0 - EGM Disabling Function Supported (50H & 51H)

#1 - EGM Display Command Supported (53H & 54H)

#2 - EGM Program Hash Calculation Supported (56H & 57H)

#3 - Promotional Ticket Printing Supported (59H & 5AH)

#4 - Combined Jackpot Broadcast Pool Supported (60H or 62H)

#5 - Individual Jackpot Pool Broadcast Supported (E0-EF)

#6 - Reserved for Future Use.

#7 - Always Zero.

Byte 96 - Status 1 byte for CCCE Class (*Note 1*)

#0 - 1 Standard Progressive win Payment Completed.

#1 - 1 Mystery Progressive win Payment Completed.

#2 - 1 CCCE with cents transfer Completed

#3 - 1 Mystery Progressive win
#4 - EGM in soft lockup state
#5 - 0 spare
#6 - 0 spare
#7 - Always Zero.

Byte 97 - Status 2 byte for CCCE Class

#0 - 0 spare
#1 - 0 spare
#2 - 0 spare
#3 - 0 spare
#4 - 0 spare
#5 - 0 spare
#6 - 0 spare
#7 - Always Zero.

Byte 98 - CCCE Class implementation byte 1

#0 - 1 Standard Progressive win notification implemented.
#1 - 1 Mystery Progressive win notification implemented
#2 - 1 CCCE increment / decrement in cents implemented
#3 - 1 Standard Progressive win payment implemented
#4 - 1 Mystery Progressive win payment implemented
#5 - 1 Standard Progressive game current pool value implemented
#6 - 1 Mystery Progressive game current pool value implemented
#7 - Always Zero.

Byte 99 - TITO Class implementation byte

#0 - "Ticket in" Using Port 1 Supported
#1 - "Time Broadcast" Implemented
#2 - 0 Spare
#3 - 0 Spare
#4 - 0 spare
#5 - 0 spare
#6 - 0 spare
#7 - Always Zero.

meter is in cents

Byte 100 - Total amount for Standard Progressive win payment in meter (LSD and LSD+1). LSD in lower nibble.

Byte 104 - Total amount for Standard Progressive win payment in meter (LSD+8 and MSD). LSD+8 in lower nibble.

meter is in cents

Byte 105 - Total amount for Mystery Progressive win Payment in meter (LSD and LSD+1). LSD in lower nibble.

Byte 109 - Total amount for Mystery Progressive win Payment in meter (LSD+8 and MSD). LSD+8 in lower nibble.

meter is in cents

Byte 110 - Total amount for CCCE transfer in with cents meter (LSD and LSD+1). LSD in lower nibble.

Byte 114 - Total amount for CCCE transfer in with cents meter (LSD+8 and MSD). LSD+8 in lower nibble.

meter is in cents

Byte 115 - Total amount for Ticket Printed (Ticket Out) in with cents meter (LSD and LSD+1). LSD in lower nibble.

Byte 119 - Total amount for Ticket Printed (Ticket Out) in with cents meter (LSD+8 and MSD). LSD+8 in lower nibble.

meter is in cents

Byte 120 - Total amount for Ticket Accepted (Ticket In) in with cents meter (LSD and LSD+1). LSD in lower nibble.

Byte 124 - Total amount for Ticket Accepted (Ticket In) in with cents meter (LSD+8 and MSD). LSD+8 in lower nibble.

Byte 125 & 126 (00) - Reserved for future use.

Byte 127 - Checksum lower nibble (0XH)

Byte 128 – 9

----- End -----