

Instruction Name	Length	Example / Info	History
{00} Nop	01	00++ <pre>typedef struct { UCHAR Opcode; // 0x00 } Nop;</pre> This bytecode is used for alignment of 1-byte opcodes and ending Elself blocks.	08-02-2024 Newly Added
{01} Evt_end	02	01 00++ <pre>typedef struct { UCHAR Opcode; // 0x01 UCHAR zAlign; // Always Zero } Evt_end;</pre> This bytecode ends the current Main/Sub script.	08-02-2024 Newly Added
{02} Evt_next	01	02++ <pre>typedef struct { UCHAR Opcode; // 0x02 } Evt_next;</pre> This bytecode moves to the next event in the sequence.	08-02-2024 Newly Added
{03} Evt_chain	02	03 ID++ <pre>typedef struct { UCHAR Opcode; // 0x03 UCHAR NextEventId; // Event ID to chain to } Evt_chain;</pre> This bytecode chains the current event to the specified next event ID, allowing the script to continue execution from the linked event.	08-02-2024 Newly Added
{04} Evt_exec	04	04 ?? ?? ID++ <pre>typedef struct { UCHAR Opcode; // 0x04 UCHAR data1; // Typically FF UCHAR GoSub; // Opcode for GoSub 0x18 UCHAR ScdId; // Sub Script ID to Jump to } Evt_exec;</pre> This bytecode executes the specified event with given parameters.	08-02-2024 Newly Added
{05} Evt_kill	02	05 ID++ <pre>typedef struct { UCHAR Opcode; // 0x05 UCHAR EventId; // Event ID to terminate } Evt_kill;</pre> This bytecode terminates the specified event.	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{06} Ifel_ck	04	<p>06 00 SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x06 UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check } Ifel_ck; This bytecode checks a condition and branches accordingly.</pre>	08-02-2024 Newly Added
{07} Else_ck	04	<p>07 00 SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x07 UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check } Else_ck; This bytecode specifies the size of the block to check if the corresponding Ifel_ck condition is met.</pre>	08-02-2024 Newly Added
{08} Endif	02	<p>08 00++</p> <pre>typedef struct { UCHAR Opcode; // 0x08 UCHAR zAlign; // Always Zero (Alignment byte) } Endif; This bytecode marks the end of an If/Elseif/Else block.</pre>	08-02-2024 Newly Added
{09} Sleep	04	<p>09 ?? CO NT++</p> <pre>typedef struct { UCHAR Opcode; // 0x09 UCHAR Sleeping; // Opcode for Sleeping 0x0A USHORT Count; // Timer / Sleep Duration } Sleep; This bytecode pauses script execution for the specified duration.</pre>	08-02-2024 Newly Added
{0A} Sleeping	03	<p>0A CO NT++</p> <pre>typedef struct { UCHAR Opcode; // 0x0A USHORT Count; // Timer / Sleep Duration } Sleeping; This bytecode pauses script execution for the specified duration.</pre>	08-02-2024 Newly Added
{0B} Wsleep	01	<p>0B++</p> <pre>typedef struct { UCHAR Opcode; // 0x0B } Wsleep; This bytecode used before 0C will wait until the current XA sound has finished playing before proceeding.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{0C} Wsleeping	01	<p>0C++</p> <pre>typedef struct { UCHAR Opcode; // 0x0C } Wsleeping;</pre> <p>This bytecode used after 0B will wait until the current XA sound has finished playing before proceeding.</p>	08-02-2024 Newly Added
{0D} For	06	<p>0D 00 SI ZE CO NT++</p> <pre>typedef struct { UCHAR Opcode; // 0x0D UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check USHORT Count; // Amount of times block is looped } For;</pre> <p>This bytecode begins a for-loop with the specified start and end values.</p>	08-02-2024 Newly Added
{0E} Next	02	<p>0E 00++</p> <pre>typedef struct { UCHAR Opcode; // 0x0E UCHAR zAlign; // Always Zero (Alignment byte) } Next;</pre> <p>This bytecode marks the end of a for-loop.</p>	08-02-2024 Newly Added
{0F} While	04	<p>0F 00 SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x0F UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check } While;</pre> <p>This bytecode begins a while-loop that continues as long as the specified condition is true.</p>	08-02-2024 Newly Added
{10} Ewhile	02	<p>10 ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x10 UCHAR LoopId; // ID of the while-loop to end } Ewhile;</pre> <p>This bytecode ends the specified while-loop.</p>	08-02-2024 Newly Added
{11} Do	04	<p>11 00 SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x11 UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check } Do;</pre> <p>This bytecode begins a do-while loop that executes the loop body once before checking the condition.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{12} Edwhile	02	<p>12 ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x12 UCHAR LoopId; // ID of the do-while loop to end } Edwhile;</pre> <p>This bytecode ends the specified do-while loop.</p>	08-02-2024 Newly Added
{13} Switch	04	<p>13 ID SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x13 UCHAR SwitchId; // ID of the switch variable USHORT Size; // Size of the block to check } Switch;</pre> <p>This bytecode begins a switch-case block with the specified switch variable and default size.</p>	08-02-2024 Newly Added
{14} Case	06	<p>14 00 SI ZE VA LU++</p> <pre>typedef struct { UCHAR Opcode; // 0x14 UCHAR zAlign; // Always Zero (Alignment byte) USHORT Size; // Size of the block to check USHORT CaseValue; // Value to compare with the switch variable } Case;</pre> <p>This bytecode defines a case within a switch-case block.</p>	08-02-2024 Newly Added
{15} Default	02	<p>15 00++</p> <pre>typedef struct { UCHAR Opcode; // 0x15 UCHAR zAlign; // Always Zero (Alignment byte) } Default;</pre> <p>This bytecode marks the default case in a switch-case block.</p>	08-02-2024 Newly Added
{16} Eswitch	02	<p>16 00++</p> <pre>typedef struct { UCHAR Opcode; // 0x16 UCHAR zAlign; // Always Zero (Alignment byte) } Eswitch;</pre> <p>This bytecode ends the switch-case block.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{17} Goto	06	<p>17 ?? ?? ?? ?? ?? ??++ typedef struct { UCHAR Opcode; // 0x17 UCHAR Ifel_ctr; // Always 0xFF (or 0x01 on specific cases) UCHAR Loop_ctr; // Always 0xFF (or 0x00 on specific cases) UCHAR zAlign; // Always 0x00 SHORT Offset; // Relative Pointer, always references the same script } Goto; This bytecode jumps to the specified offset within the script.</p>	08-02-2024 Newly Added
{18} Gosub	02	<p>18 ID++ typedef struct { UCHAR Opcode; // 0x18 UCHAR SubroutineId; // ID of the subroutine to call } Gosub; This bytecode calls the specified subroutine.</p>	08-02-2024 Newly Added
{19} Return	02	<p>19 00++ typedef struct { UCHAR Opcode; // 0x19 UCHAR zAlign; // Always Zero (Alignment byte) } Return; This bytecode returns from the specified subroutine.</p>	08-02-2024 Newly Added
{1A} Break	02	<p>1A 00++ typedef struct { UCHAR Opcode; // 0x1A UCHAR zAlign; // Always Zero (Alignment byte) } Break; This bytecode breaks out of the specified loop.</p>	08-02-2024 Newly Added
{1B} For2	06	<p>1B 00 XX XX 00 XX XX++ typedef struct { UCHAR Opcode; // 0x1B UCHAR zAlign; // Always Zero (Alignment byte) SHORT StartValue; // Start value of the loop counter UCHAR zAlign2; // Always Zero (Alignment byte) SHORT EndValue; // End value of the loop counter } For2; This bytecode begins a for-loop with the specified start and end values.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{1C} Break_point	01	<p>1C++</p> <pre>typedef struct { UCHAR Opcode; // 0x1C } Break_point;</pre> <p>This bytecode sets a breakpoint for debugging purposes.</p>	08-02-2024 Newly Added
{1D} Work_copy	04	<p>1D ?? ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x1D UCHAR Source; // Source index UCHAR DestinationId; // Destination index UCHAR Typecast; // Typecast operation } Work_copy;</pre> <p>This bytecode copies a value from the source index to the destination index with an optional typecast.</p>	08-02-2024 Newly Added
{1E} Nop	01	<p>1E++</p> <pre>typedef struct { UCHAR Opcode; // 0x1E } Nop;</pre> <p>This bytecode is used for alignment of 1-byte opcodes and ending Elself blocks.</p>	08-02-2024 Newly Added
{1F} Nop	01	<p>1F++</p> <pre>typedef struct { UCHAR Opcode; // 0x1F } Nop;</pre> <p>This bytecode is used for alignment of 1-byte opcodes and ending Elself blocks.</p>	08-02-2024 Newly Added
{20} Nop	01	<p>20++</p> <pre>typedef struct { UCHAR Opcode; // 0x20 } Nop;</pre> <p>This bytecode is used for alignment of 1-byte opcodes and ending Elself blocks.</p>	08-02-2024 Newly Added
{21} Ck	04	<p>21 ?? ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x21 UCHAR Flag; // System flag to check UCHAR Id; // Bit ID to check UCHAR OnOff; // On/Off state to check } Ck;</pre> <p>This bytecode checks the specified system flag and bit ID for the given On/Off state.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{22} Set	04	<p>22 ?? ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x22 UCHAR Flag; // System flag to set UCHAR Id; // Bit ID to set UCHAR OnOff; // On/Off state to set } Set;</pre> <p>This bytecode sets the specified system flag and bit ID to the given On/Off state.</p>	08-02-2024 Newly Added
{23} Cmp	06	<p>23 00 ID ?? VA LU++</p> <pre>typedef struct { UCHAR Opcode; // 0x23 UCHAR zAlign; // Always 0x00 UCHAR MemberId; // Id of the member to compare UCHAR Operator; // Type of comparison SHORT Value; // Value to compare the member to } Cmp;</pre> <p>This bytecode compares the specified system flag with the given value using the provided comparison operator.</p>	08-02-2024 Newly Added
{24} Save	04	<p>24 ID XX XX++</p> <pre>typedef struct { UCHAR Opcode; // 0x24 UCHAR Destination; // Destination index SHORT Source; // Source value } Save;</pre> <p>This bytecode saves the specified source value to the destination index.</p>	08-02-2024 Newly Added
{25} Copy	03	<p>25 ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x25 UCHAR Destination; // Destination index UCHAR Source; // Source index } Copy;</pre> <p>This bytecode copies the value from the source index to the destination index.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{26} Calc	06	<p>26 00 ?? ?? VA LU++</p> <pre>typedef struct { UCHAR Opcode; // 0x26 UCHAR zAlign; // Always Zero (Alignment byte) UCHAR Operator; // Arithmetic operation to perform UCHAR Flag; // Memory Location to apply math to SHORT Value; // Amount used in operation } Calc; </pre> <p>This bytecode performs the specified arithmetic operation on the operands and stores the result.</p>	08-02-2024 Newly Added
{27} Calc2	04	<p>27 ?? ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x27 UCHAR Operator; // Type of operation UCHAR Flag; // F_atari, etc UCHAR Value; // Value to use in the operation } Calc2; </pre> <p>This bytecode performs the specified arithmetic operation on the two operands and stores the result.</p>	08-02-2024 Newly Added
{28} Sce_rnd	01	<p>28++</p> <pre>typedef struct { UCHAR Opcode; // 0x28 } Sce_rnd; </pre> <p>This bytecode generates a random value.</p>	08-02-2024 Newly Added
{29} Cut_chg	02	<p>29 ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x29 UCHAR CutId; // ID of the cutscene } Cut_chg; </pre> <p>This bytecode changes the current camera to the specified camera ID.</p>	08-02-2024 Newly Added
{2A} Cut_old	01	<p>2A++</p> <pre>typedef struct { UCHAR Opcode; // 0x2A } Cut_old; </pre> <p>This bytecode reverts to the previous camera.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{2B} Message_on	06	<p>2B 00 ?? ?? SI ZE++</p> <pre>typedef struct { UCHAR Opcode; // 0x2B UCHAR zAlign; // Always Zero (Alignment byte) UCHAR Type; // Message type UCHAR MessageId; // ID of the message to display USHORT DisplayTime; // Time to display the message } Message_on; This bytecode displays the specified message.</pre>	08-02-2024 Newly Added
{2C} Aot_set	20	<p>2C ID ?? ?? ?? ?? XX XX ZZ ZZ SI ZE SI ZE ?? ?? ?? ?? ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x2C CHAR Aot; // Aot UCHAR SCE; // Id UCHAR SAT; // Type UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X; // Position SHORT Z; // Position USHORT W; // Size USHORT D; // Size USHORT Data0; // Sce_Message / Flag Type / Always 0x00FF USHORT Data1; // Sce_Message / Flag Id / Script Id Init & Complete USHORT Data2; // Always 0xFFFF } Aot_set; This bytecode sets the properties of the specified AOT.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{2D} Obj_model_set	38	<p>2D ?? ID ?? ?? ?? ?? ?? ?? TI PE BE FL XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ XX XX YY YY ZZ ZZ XX XX YY YY ZZ ZZ++</p> <pre> typedef struct { UCHAR Opcode; // 0x2D UCHAR MD1; // MD1 File Id UCHAR Id; // } Obj_model_set; Global->Obj_model[0].Id UCHAR Ccol_old; // CC_WORK structure UCHAR Ccol_no; // CC_WORK structure UCHAR Ctex_old; // CC_WORK structure UCHAR nFloor; // UCHAR Super; // USHORT Type; // Global->Obj_model[0].Type USHORT BeFlag; // Global->Obj_model[0].Be_flg SHORT Attribute; // Global->Obj_model[0].Attribute SHORT X; // SHORT Y; // SHORT Z; // SHORT DirX; // SHORT DirY; // SHORT DirZ; // SHORT AtariOffsetX; // SHORT AtariOffsetY; // SHORT AtariOffsetZ; // SHORT AtariSizeX; // SHORT AtariSizeY; // SHORT AtariSizeZ; // } Obj_model_set; This bytecode sets the properties of the specified object model. </pre>	08-02-2024 Newly Added
{2E} Work_set	03	<p>2E ?? ID++</p> <pre> typedef struct { UCHAR Opcode; // 0x2E UCHAR Type; // Type of Work } Work_set; Set to Select UCHAR EntityId; // ID of Entity to select } Work_set; This bytecode sets the properties of the specified work (task). </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{2F} Speed_set	04	2F ID ?? 00++ <pre>typedef struct { UCHAR Opcode; // 0x2F UCHAR SpeedId; // ID of the speed setting USHORT SpeedValue; // Value of the speed setting } Speed_set; </pre> This bytecode sets the specified speed setting.	08-02-2024 Newly Added
{30} Add_speed	01	30++ <pre>typedef struct { UCHAR Opcode; // 0x30 } Add_speed; </pre> This bytecode increments the speed setting.	08-02-2024 Newly Added
{31} Add_aspeed	01	31++ <pre>typedef struct { UCHAR Opcode; // 0x31 } Add_aspeed; </pre> This bytecode increments the angular speed setting.	08-02-2024 Newly Added
{32} Pos_set	08	32 00 XX XX YY YY ZZ ZZ++ <pre>typedef struct { UCHAR Opcode; // 0x32 UCHAR zAlign; // Always Zero (Alignment byte) SHORT PosX; // X position SHORT PosY; // Y position SHORT PosZ; // Z position } Pos_set; </pre> This bytecode sets the position in 3D space.	08-02-2024 Newly Added
{33} Dir_set	08	33 00 DX DX DY DY DZ DZ++ <pre>typedef struct { UCHAR Opcode; // 0x33 UCHAR zAlign; // Always Zero (Alignment byte) SHORT DirX; // X direction SHORT DirY; // Y direction SHORT DirZ; // Z direction } Dir_set; </pre> This bytecode sets the direction in 3D space.	08-02-2024 Newly Added
{34} Member_set	04	34 ?? ?? ??++ <pre>typedef struct { UCHAR Opcode; // 0x34 UCHAR Destination; // Destination SHORT Source; // Source } Member_set; </pre> This bytecode sets the properties of the specified member.	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{35} Member_set2	03	<p>35 ?? ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x35 UCHAR Destination; // Destination UCHAR Source; // Source } Member_set2;</pre> <p>This bytecode sets a single property of the specified member.</p>	08-02-2024 Newly Added
{36} Se_on	12	<p>36 ?? ?? ?? ?? XX XX YY YY ZZ ZZ++</p> <pre>typedef struct { UCHAR Opcode; // 0x36 UCHAR VAB; // VAB Bank Id SHORT EDT; // EDT Sample Id SHORT data0; // Sound Reverberation, Work Aot/Obj No SHORT X; // Position SHORT Y; SHORT Z; } Se_on;</pre> <p>This bytecode plays the specified sound effect with the given parameters.</p>	08-02-2024 Newly Added
{37} Sca_id_set	04	<p>37 ?? ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x37 UCHAR iEntry; // SCA Collision Entry USHORT Id; // New Collision ID } Sca_id_set;</pre> <p>This bytecode sets the specified scale ID to the given value.</p>	08-02-2024 Newly Added
{38} Flr_set	03	<p>38 ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x38 UCHAR Id; UCHAR Flag; } Flr_set;</pre> <p>This bytecode sets the specified floor ID to the given value.</p>	08-02-2024 Newly Added
{39} Dir_ck	08	<p>39 00 XX XX ZZ ZZ ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x39 UCHAR zAlign; // 0x00 SHORT X; SHORT Z; SHORT Add; } Dir_ck;</pre> <p>This bytecode checks the specified direction coordinates.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{3A} Sce_espr_on	16	<p>3A 00 ?? ?? ?? ?? ?? ?? ?? XX XX YY YY ZZ ZZ DY DY++</p> <pre> typedef struct { UCHAR Opcode; // 0x3A UCHAR zAlign; // 0x00 USHORT data0; USHORT data1; USHORT data2; SHORT X; SHORT Y; SHORT Z; SHORT DirY; } Sce_espr_on; </pre> <p>This bytecode turns on the specified espr with the given parameters.</p>	08-02-2024 Newly Added
{3B} Door_aot_set	32	<p>3B ID ?? ?? ?? ?? ?? ?? XX XX ZZ ZZ SI ZE SI ZE XX XX YY YY ZZ ZZ 00 ?? ?? ?? ?? ++</p> <pre> typedef struct { UCHAR Opcode; // 0x3B UCHAR Aot; // Aot UCHAR SCE; // Id /* tagSCE_AOT UCHAR SAT; // Type UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X; // Position SHORT Z; // Position USHORT W; // Size USHORT D; // Size */ tagSCE_AOT SHORT Next_pos_x; // */ tagIN_DOOR_WORK SHORT Next_pos_y; // SHORT Next_pos_z; // SHORT Next_cdir_y; // UCHAR Next_stage; // UCHAR Next_room; // UCHAR Next_cut; // UCHAR Next_nfloor; // UCHAR Dtex_type; // D02 File Id UCHAR Door_type; // D02 Animation Set UCHAR Knock_type; // Sound UCHAR Key_id; // UCHAR Key_type; // UCHAR Free; // */ tagIN_DOOR_WORK } Door_aot_set; </pre> <p>This bytecode sets the specified door AOT parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{3C} Cut_auto	02	<p>3C ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x3C UCHAR OnOff; // 00 OFF, 01 } Cut_auto; </pre> <p>This bytecode sets the specified auto cutscene parameters.</p>	08-02-2024 Newly Added
{3D} Member_copy	03	<p>3D ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x3D UCHAR Destination; // Destination UCHAR Source; // Source } Member_copy; </pre> <p>This bytecode copies the specified member parameters from the source to the destination.</p>	08-02-2024 Newly Added
{3E} Member_cmp	06	<p>3E 00 ?? ?? VA LU++</p> <pre>typedef struct { UCHAR Opcode; // 0x3E UCHAR zAlign; // Always Zero UCHAR Flag; // } Member_cmp; </pre> <p>Load_member_addr_branch() argv[1] UCHAR Operator; // Comparison operator SHORT Value; // Value to compare</p> <p>This bytecode compares the specified member parameters with the given value.</p>	08-02-2024 Newly Added
{3F} Plc_motion	04	<p>3F ID ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x3F UCHAR MotionId; // EDD/EMR Id UCHAR Mode; // Mode to set } Plc_motion; </pre> <p>the motion to UCHAR Param; // Parameter for the motion</p> <p>This bytecode sets the specified motion parameters.</p>	08-02-2024 Newly Added
{40} Plc_dest	08	<p>40 00 ?? ?? XX XX ZZ ZZ++</p> <pre>typedef struct { UCHAR Opcode; // 0x40 UCHAR zAlign; // 0x00 UCHAR Animation; // EDD/EMR Id UCHAR Bit; // Room_flg SHORT X; // Destination SHORT Z; // Destination } Plc_dest; </pre> <p>This bytecode sets the specified destination parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{41} Plc_neck	10	<pre>41 ?? XX XX YY YY ZZ ZZ ?? ?? ++ typedef struct { UCHAR Opcode; // 0x41 UCHAR Op; // SHORT X; // SHORT Y; // SHORT Z; // UCHAR SpeedX; // UCHAR SpeedZ; // } Plc_neck; This bytecode sets the specified neck parameters.</pre>	08-02-2024 Newly Added
{42} Plc_ret	01	<pre>42++ typedef struct { UCHAR Opcode; // 0x42 } Plc_ret; This bytecode returns from the specified plc function.</pre>	08-02-2024 Newly Added
{43} Plc_flg	04	<pre>43 00 ?? ??++ typedef struct { UCHAR Opcode; // 0x43 UCHAR zAlign; // 0x00 UCHAR data0; // UCHAR data1; // } Plc_flg; This bytecode sets the specified flag parameters.</pre>	08-02-2024 Newly Added
{44} Sce_em_set	22	<pre>44 00 ?? ID ?? ?? ?? ?? ?? XX XX YY YY ZZ ZZ DY DY ?? ?? ++ typedef struct { UCHAR Opcode; // 0x44 UCHAR Nop; // 0x00 CHAR Em_no; // Em_no UCHAR Id; // EMD File Id for Em_bin_load() function USHORT Type; // Type UCHAR nFloor; // nFloor UCHAR Sound_flg; // Sound_flg UCHAR Model_type; // Model_type UCHAR Em_set_flg; // Em_set_flg SHORT Pos_x; // Position X SHORT Pos_y; // Position Y SHORT Pos_z; // Position Z SHORT Cdir_y; // Rotation Y SHORT Motion; // Motion SHORT Ctr_flg; // Ctr_flg } Sce_em_set; This bytecode sets the specified enemy parameters.</pre>	08-02-2024 Newly Added
{45} Col_chg_set	05	<pre>45 ?? ?? ?? ??++ typedef struct { UCHAR Opcode; // 0x45 UCHAR Data0; // Data0 UCHAR Data1; // Data1 UCHAR Data2; // Data2 UCHAR Data3; // Data3 } Col_chg_set; This bytecode sets the specified color parameters.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{46} Aot_reset	10	<p>46 ID ?? ?? ?? XX XX XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x46 CHAR Aot; // Aot UCHAR SCE; // SCE UCHAR SAT; // SAT SHORT Data0; // Data0 SHORT Data1; // Data1 SHORT Data2; // Data2 } Aot_reset; </pre> <p>This bytecode resets the specified AOT parameters.</p>	08-02-2024 Newly Added
{47} Aot_on	02	<p>47 ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x47 CHAR Aot; // Aot } Aot_on; </pre> <p>This bytecode turns on the specified AOT.</p>	08-02-2024 Newly Added
{48} Super_set	16	<p>48 00 ?? ID XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ ++</p> <pre>typedef struct { UCHAR Opcode; // 0x48 UCHAR zAlign; // Alignment byte (always 0x00) UCHAR Work; // Work Type UCHAR Id; // Work Aot/Obj No SHORT pX; // Parent X SHORT pY; // Parent Y SHORT pZ; // Parent Z SHORT dX; // Destination X SHORT dY; // Destination Y SHORT dZ; // Destination Z } Super_set; </pre> <p>This bytecode sets the specified super parameters.</p>	08-02-2024 Newly Added
{49} Super_reset	08	<p>49 00 DX DX DY DY DZ DZ ++</p> <pre>typedef struct { UCHAR Opcode; // 0x49 UCHAR zAlign; // Alignment byte (always 0x00) SHORT dX; // Destination X SHORT dY; // Destination Y SHORT dZ; // Destination Z } Super_reset; </pre> <p>This bytecode resets the specified super parameters.</p>	08-02-2024 Newly Added
{4A} Plc_gun	02	<p>4A ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x4A UCHAR GunId; // ID of the gun } Plc_gun; </pre> <p>This bytecode sets the specified gun parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{4B} Cut_replace	03	<pre> 4B ID ??++ typedef struct { UCHAR Opcode; // 0x4B UCHAR Id; // Id UCHAR Value; // Value } Cut_replace; This bytecode replaces the specified cutscene parameters. </pre>	08-02-2024 Newly Added
{4C} Sce_espr_kill	05	<pre> 4C ID ?? ?? ??++ typedef struct { UCHAR Opcode; // 0x4C UCHAR id; // Esp_kill() argv[0] UCHAR tp; // Esp_kill() argv[1] CHAR WorkKind; // Get_matrix() argv[0] CHAR WorkNo; // Get_matrix() argv[1] } Sce_espr_kill; This bytecode kills the specified espr with the given parameters. </pre>	08-02-2024 Newly Added
{4D} Door_model_set	22	<pre> 4D ?? ID ?? ?? ?? ?? XX XX YY YY ZZ ZZ DY DY ?? ?? ?? ?? ?? ?? ++ typedef struct { UCHAR Opcode; // 0x4D UCHAR data0; // data0 UCHAR Id; // DOOR_WORK.Id UCHAR ofsY; // DOOR_WORK.Atd[3] UCHAR BeFlg; // DOOR_WORK.Be_flg UCHAR data5; // data5 USHORT data6; // data6 SHORT X; // DOOR_WORK.Atd[3] SHORT Y; // DOOR_WORK.Atd[3] SHORT Z; // DOOR_WORK.Atd[3] SHORT DirY; // DOOR_WORK.Atd[3] USHORT data10; // DOOR_WORK.MATRIX USHORT data11; // DOOR_WORK.MATRIX USHORT data12; // DOOR_WORK.MATRIX } Door_model_set; This bytecode sets the specified door model parameters. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{4E} Item_aot_set	22	<p>4E ID ?? ?? ?? ?? ?? XX XX ZZ ZZ SI ZE SI ZE ?? ?? ?? ?? ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x4E UCHAR Aot; // Aot UCHAR SCE; // SCE UCHAR SAT; // SAT UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X; // X SHORT Z; // Z USHORT W; // W USHORT D; // D USHORT iItem; // iItem USHORT nItem; // nItem USHORT Flag; // Flag UCHAR MD1; // MD1 UCHAR Action; // Action } Item_aot_set;</pre> <p>This bytecode sets the specified item AOT parameters.</p>	08-02-2024 Newly Added
{4F} Sce_key_ck	04	<p>4F ?? VA LU ++</p> <pre>typedef struct { UCHAR Opcode; // 0x4F UCHAR Flag; // Flag USHORT Value; // Value } Sce_key_ck;</pre> <p>This bytecode checks the specified key parameters.</p>	08-02-2024 Newly Added
{50} Sce_trg_ck	04	<p>50 ?? VA LU ++</p> <pre>typedef struct { UCHAR Opcode; // 0x50 UCHAR Flag; // Flag USHORT Value; // Value } Sce_trg_ck;</pre> <p>This bytecode checks the specified trigger parameters.</p>	08-02-2024 Newly Added
{51} Sce_bgm_control	06	<p>51 ID ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x51 UCHAR Id; // Id UCHAR Op; // Op UCHAR Type; // Type UCHAR VolL; // VolL UCHAR VolR; // VolR } Sce_bgm_control;</pre> <p>This bytecode controls the specified BGM parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{52} Sce_espr_control	06	<p>52 ID ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x52 UCHAR Id; // Id UCHAR Type; // Type UCHAR Return; // Return CHAR WorkKind; // WorkKind CHAR WorkNo; // WorkNo } Sce_espr_control;</pre> <p>This bytecode controls the specified espr parameters.</p>	08-02-2024 Newly Added
{53} Sce_fade_set	06	<p>53 ?? ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x53 UCHAR data0; // data0 UCHAR data1; // data1 UCHAR data2; // data2 USHORT data3; // data3 } Sce_fade_set;</pre> <p>This bytecode sets the specified fade parameters.</p>	08-02-2024 Newly Added
{54} Sce_espr3d_on	22	<p>54 00 ?? ?? ?? ?? XX XX YY YY ZZ ZZ DX DX DY DY DZ DZ ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x54 UCHAR zAlign; // zAlign USHORT data0; // data0 USHORT data1; // data1 USHORT data2; // data2 SHORT X; // X SHORT Y; // Y SHORT Z; // Z SHORT DirX; // DirX SHORT DirY; // DirY SHORT DirZ; // DirZ SHORT data3; // data3 } Sce_espr3d_on;</pre> <p>This bytecode turns on the specified 3D espr with the given parameters.</p>	08-02-2024 Newly Added
{55} Member_calc	06	<p>55 ?? ?? ?? ?? VA LU ++</p> <pre>typedef struct { UCHAR Opcode; // 0x55 UCHAR Operator; // Operator USHORT Flag; // Flag SHORT Value; // Value } Member_calc;</pre> <p>This bytecode performs the specified arithmetic operation on the member with the given value.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{56} Member_calc2	04	<p>56 ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x56 UCHAR Operator; // Operator UCHAR Flag; // Flag UCHAR Value; // Value } Member_calc2;</pre> <p>This bytecode performs the specified arithmetic operation on the member with the given value.</p>	08-02-2024 Newly Added
{57} Sce_bgmtbl_set	08	<p>57 00 ?? ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x57 UCHAR zAlign; // zAlign UCHAR Stage; // Stage UCHAR Room; // Room USHORT data1; // data1 USHORT data2; // data2 } Sce_bgmtbl_set;</pre> <p>This bytecode sets the specified BGM table parameters.</p>	08-02-2024 Newly Added
{58} Plc_rot	04	<p>58 ID ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x58 UCHAR Id; // Id USHORT Sce_free0; // Sce_free0 } Plc_rot;</pre> <p>This bytecode sets the specified rotation parameters.</p>	08-02-2024 Newly Added
{59} Xa_on	04	<p>59 ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x59 UCHAR Mode; // Mode USHORT Number; // Number } Xa_on;</pre> <p>This bytecode turns on the specified XA with the given parameters.</p>	08-02-2024 Newly Added
{5A} Weapon_chg	02	<p>5A ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x5A UCHAR WeaponId; // ID of the // weapon to change } Weapon_chg;</pre> <p>This bytecode changes the specified weapon parameters.</p>	08-02-2024 Newly Added
{5B} Plc_cnt	02	<p>5B ID ++</p> <pre>typedef struct { UCHAR Opcode; // 0x5B UCHAR Id; // Id } Plc_cnt;</pre> <p>This bytecode performs the specified arithmetic operation on the counter with the given value.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{5C} Sce_shake_on	03	<p>5C ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x5C CHAR Slide_ofs; // Slide_ofs CHAR Copy_ofs; // Copy_ofs } Sce_shake_on;</pre> <p>This bytecode turns on the specified shake effect with the given parameters.</p>	08-02-2024 Newly Added
{5D} Mizu_div_set	02	<p>5D ID ++</p> <pre>typedef struct { UCHAR Opcode; // 0x5D UCHAR Id; // Id } Mizu_div_set;</pre> <p>This bytecode sets the specified water division parameters.</p>	08-02-2024 Newly Added
{5E} Keep_Item_ck	02	<p>5E ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x5E UCHAR ItemId; // ID of the // item to check } Keep_Item_ck;</pre> <p>This bytecode checks the specified item parameters.</p>	08-02-2024 Newly Added
{5F} Xa_vol	02	<p>5F ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x5F UCHAR Volume; // XA volume // level } Xa_vol;</pre> <p>This bytecode sets the specified XA volume and pan parameters.</p>	08-02-2024 Newly Added
{60} Kage_set	14	<p>60 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x60 UCHAR Work; // Work CHAR Id; // Id UCHAR Data0; // Data0 UCHAR Data1; // Data1 UCHAR Data2; // Data2 USHORT Data3; // Data3 USHORT Data4; // Data4 USHORT Data5; // Data5 USHORT Data6; // Data6 } Kage_set;</pre> <p>This bytecode sets the specified shadow parameters.</p>	08-02-2024 Newly Added
{61} Cut_be_set	04	<p>61 ID ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x61 UCHAR Id; // Id UCHAR Value; // Value UCHAR OnOff; // OnOff } Cut_be_set;</pre> <p>This bytecode sets the specified cutscene parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{62} Sce_Item_lost	02	<p>62 ID++</p> <pre>typedef struct { UCHAR Opcode; // 0x62 UCHAR ItemId; // ID of the lost item } Sce_Item_lost;</pre> <p>This bytecode removes the specified item from the inventory.</p>	08-02-2024 Newly Added
{63} Plc_gun_eff	01	<p>63++</p> <pre>typedef struct { UCHAR Opcode; // 0x63 } Plc_gun_eff;</pre> <p>This bytecode sets the specified gun effect parameters.</p>	08-02-2024 Newly Added
{64} Sce_espr_on2	16	<p>64 ?? ?? ?? ?? ?? ?? XX XX YY YY ZZ ZZ ?? ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x64 UCHAR dir_y_id2; // dir_y_id2 USHORT data1; // data1 UCHAR WorkKind; // WorkKind UCHAR WorkNo; // WorkNo USHORT data3; // data3 SHORT X; // X SHORT Y; // Y SHORT Z; // Z USHORT DirY; // DirY } Sce_espr_on2;</pre> <p>This bytecode turns on the specified espr with the given parameters.</p>	08-02-2024 Newly Added
{65} Sce_espr_kill2	02	<p>65 ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x65 UCHAR EsprId; // ID of the effect sprite to kill } Sce_espr_kill2;</pre> <p>This bytecode kills the specified espr with the given parameters.</p>	08-02-2024 Newly Added
{66} Plc_stop	01	<p>66++</p> <pre>typedef struct { UCHAR Opcode; // 0x66 } Plc_stop;</pre> <p>This bytecode stops the specified plc function.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{67} Aot_set_4p	28	<pre> 67 ID ?? ?? ?? ?? ?? XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ ?? ?? ?? ?? ++ typedef struct { UCHAR Opcode; // 0x67 UCHAR Aot; // Aot UCHAR SCE; // Id UCHAR SAT; // Type UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X0; // X0 SHORT Z0; // Z0 SHORT X1; // X1 SHORT Z1; // Z1 SHORT X2; // X2 SHORT Z2; // Z2 SHORT X3; // X3 SHORT Z3; // Z3 USHORT Data0; // Data0 USHORT Data1; // Data1 USHORT Data2; // Data2 } Aot_set_4p; This bytecode sets the specified 4-point AOT parameters.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{68} Door_aot_set_4p	40	<pre> 68 ID ?? ?? ?? ?? ?? XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ XX XX YY YY ZZ ZZ ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ++ typedef struct { UCHAR Opcode; // 0x68 UCHAR Aot; // Aot UCHAR SCE; // Id UCHAR SAT; // Type UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X0; // X0 SHORT Z0; // Z0 SHORT X1; // X1 SHORT Z1; // Z1 SHORT X2; // X2 SHORT Z2; // Z2 SHORT X3; // X3 SHORT Z3; // Z3 SHORT Next_pos_x; // Next_pos_x SHORT Next_pos_y; // Next_pos_y SHORT Next_pos_z; // Next_pos_z SHORT Next_cdir_y; // Next_cdir_y UCHAR Next_stage; // Next_stage UCHAR Next_room; // Next_room UCHAR Next_cut; // Next_cut UCHAR Next_nfloor; // Next_nfloor UCHAR Dtex_type; // Dtex_type UCHAR Door_type; // Door_type UCHAR Knock_type; // Knock_type UCHAR Key_id; // Key_id UCHAR Key_type; // Key_type UCHAR Free; // Free } Door_aot_set_4p; </pre> <p>This bytecode sets the specified 4-point door AOT parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{69} Item_aot_set_4p	30	<pre> 69 ID ?? ?? ?? ?? ?? XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ XX XX ZZ ZZ ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ++ typedef struct { UCHAR Opcode; // 0x69 UCHAR Aot; // Aot UCHAR SCE; // Id UCHAR SAT; // Type UCHAR nFloor; // nFloor UCHAR Super; // Super SHORT X0; // X0 SHORT Z0; // Z0 SHORT X1; // X1 SHORT Z1; // Z1 SHORT X2; // X2 SHORT Z2; // Z2 SHORT X3; // X3 SHORT Z3; // Z3 USHORT iItem; // iItem USHORT nItem; // nItem USHORT Flag; // Flag UCHAR MD1; // MD1 UCHAR Action; // Action } Item_aot_set_4p; </pre> <p>This bytecode sets the specified 4-point item AOT parameters.</p>	08-02-2024 Newly Added
{6A} Light_pos_set	06	<pre> 6A 00 ?? ?? XX XX ++ typedef struct { UCHAR Opcode; // 0x6A UCHAR zAlign; // zAlign UCHAR Index; // Index UCHAR XYZ; // XYZ SHORT Position; // Position } Light_pos_set; </pre> <p>This bytecode sets the specified light position parameters.</p>	08-02-2024 Newly Added
{6B} Light_kido_set	04	<pre> 6B ?? ?? XX XX ++ typedef struct { UCHAR Opcode; // 0x6B UCHAR Index; // Index SHORT Luminosity; // Luminosity } Light_kido_set; </pre> <p>This bytecode sets the specified light intensity and color parameters.</p>	08-02-2024 Newly Added
{6C} Rbj_reset	01	<pre> 6C++ typedef struct { UCHAR Opcode; // 0x6C } Rbj_reset; </pre> <p>This bytecode resets the specified rbj parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{6D} Sce_scr_move	04	<pre>6D 00 XX XX ++ typedef struct { UCHAR Opcode; // 0x6D UCHAR zAlign; // zAlign SHORT Scrl_y; // Scrl_y } Sce_scr_move; This bytecode moves the specified screen with the given parameters.</pre>	08-02-2024 Newly Added
{6E} Parts_set	06	<pre>6E 00 ?? ?? XX XX ++ typedef struct { UCHAR Opcode; // 0x6E UCHAR zAlign; // zAlign CHAR Id; // Id CHAR Type; // Type SHORT Value; // Value } Parts_set; This bytecode sets the specified parts parameters.</pre>	08-02-2024 Newly Added
{6F} Movie_on	02	<pre>6F ??++ typedef struct { UCHAR Opcode; // 0x6F UCHAR MovieId; // ID of the movie } Movie_on; This bytecode plays the specified movie.</pre>	08-02-2024 Newly Added
{70} Splc_ret	01	<pre>70++ typedef struct { UCHAR Opcode; // 0x70 } Splc_ret; This bytecode returns from the specified splc function.</pre>	08-02-2024 Newly Added
{71} Splc_sce	01	<pre>71++ typedef struct { UCHAR Opcode; // 0x71 } Splc_sce; This bytecode sets the specified splc parameters.</pre>	08-02-2024 Newly Added
{72} Super_on	16	<pre>72 00 ?? ?? ?? ?? XX XX XX XX XX XX XX XX ++ typedef struct { UCHAR Opcode; // 0x72 UCHAR zAlign; // zAlign UCHAR data0; // data0 UCHAR data1; // data1 SHORT data2; // data2 SHORT data3; // data3 SHORT data4; // data4 SHORT data5; // data5 SHORT data6; // data6 SHORT data7; // data7 } Super_on; This bytecode turns on the specified super with the given parameters.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{73} Mirror_set	08	<p>73 ?? ?? XX XX XX XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x73 UCHAR Flag; // Flag USHORT Position; // Position USHORT Min; // Min USHORT Max; // Max } Mirror_set; </pre> <p>This bytecode sets the specified mirror position parameters.</p>	08-02-2024 Newly Added
{74} Sce_fade_adjust	04	<p>74 ?? ?? XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x74 UCHAR data0; // data0 SHORT data1; // data1 } Sce_fade_adjust; </pre> <p>This bytecode adjusts the specified fade parameters.</p>	08-02-2024 Newly Added
{75} Sce_espr3d_on2	22	<p>75 ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x75 UCHAR dir_y_id2; // dir_y_id2 USHORT bit; // bit USHORT data4; // data4 USHORT data6; // data6 USHORT data8; // data8 USHORT dataA; // dataA USHORT dataC; // dataC USHORT dataE; // dataE USHORT data10; // data10 USHORT data12; // data12 USHORT data14; // data14 } Sce_espr3d_on2; </pre> <p>This bytecode turns on the specified 3D espr with the given parameters.</p>	08-02-2024 Newly Added
{76} Sce_Item_get	03	<p>76 ID ?? ++</p> <pre>typedef struct { UCHAR Opcode; // 0x76 UCHAR Id; // Id UCHAR Num; // Num } Sce_Item_get; </pre> <p>This bytecode gets the specified item with the given quantity.</p>	08-02-2024 Newly Added
{77} Sce_line_start	04	<p>77 ID XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x77 UCHAR Id; // Id USHORT Value; // Value } Sce_line_start; </pre> <p>This bytecode starts the specified line with the given parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{78} Sce_line_main	06	<p>78 ID XX XX XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x78 UCHAR Id; // Id SHORT Data0; // Data0 SHORT Data1; // Data1 } Sce_line_main; </pre> <p>This bytecode sets the specified line parameters.</p>	08-02-2024 Newly Added
{79} Sce_line_end	01	<p>79 ++</p> <pre>typedef struct { UCHAR Opcode; // 0x79 } Sce_line_end; </pre> <p>This bytecode ends the specified line.</p>	08-02-2024 Newly Added
{7A} Sce_parts_bomb	16	<p>7A 00 ?? ?? ?? ?? XX XX XX XX XX XX XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x7A UCHAR zAlign; // zAlign UCHAR data2; // data2 UCHAR data3; // data3 UCHAR data4; // data4 UCHAR data5; // data5 SHORT data6; // data6 SHORT data8; // data8 SHORT dataA; // dataA SHORT dataC; // dataC SHORT dataE; // dataE } Sce_parts_bomb; </pre> <p>This bytecode bombs the specified parts with the given parameters.</p>	08-02-2024 Newly Added
{7B} Sce_parts_down	16	<p>7B ID XX XX XX XX XX XX XX XX XX XX ++</p> <pre>typedef struct { UCHAR Opcode; // 0x7B UCHAR Id; // Id SHORT X; // X SHORT Y; // Y SHORT Z; // Z SHORT cDirZ; // cDirZ SHORT DirX; // DirX SHORT DirY; // DirY SHORT DirZ; // DirZ } Sce_parts_down; </pre> <p>This bytecode moves down the specified parts with the given parameters.</p>	08-02-2024 Newly Added
{7C} Light_color_set	06	<p>7C ?? ?? ?? ?? ?? 00 ++</p> <pre>typedef struct { UCHAR Opcode; // 0x7C UCHAR Index; // Index UCHAR R; // R UCHAR G; // G UCHAR B; // B UCHAR zAlign; // zAlign } Light_color_set; </pre> <p>This bytecode sets the specified light color parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{7D} Light_pos_set2	06	<pre>7D ?? ?? ?? ?? XX XX ++ typedef struct { UCHAR Opcode; // 0x7D UCHAR nCut; // nCut UCHAR Index; // Index UCHAR XYZ; // XYZ SHORT Position; // Position } Light_pos_set2; This bytecode sets the specified light position parameters.</pre>	08-02-2024 Newly Added
{7E} Light_kido_set2	06	<pre>7E 00 ?? ?? XX XX ++ typedef struct { UCHAR Opcode; // 0x7E UCHAR zAlign; // zAlign UCHAR nCut; // nCut UCHAR Index; // Index USHORT Luminosity; // Luminosity } Light_kido_set2; This bytecode sets the specified light intensity and color parameters.</pre>	08-02-2024 Newly Added
{7F} Light_color_set2	06	<pre>7F ?? ?? ?? ?? ?? ?? ++ typedef struct { UCHAR Opcode; // 0x7F UCHAR nCut; // nCut UCHAR Index; // Index UCHAR R; // R UCHAR G; // G UCHAR B; // B } Light_color_set2; This bytecode sets the specified light color parameters.</pre>	08-02-2024 Newly Added
{80} Se_vol	02	<pre>80 ?? ??++ typedef struct { UCHAR Opcode; // 0x80 UCHAR Volume; // Volume level } Se_vol; This bytecode sets the specified sound effect volume and pan parameters.</pre>	08-02-2024 Newly Added
{81} Keep_Item_ck2	03	<pre>81 ID ??++ typedef struct { UCHAR Opcode; // 0x81 UCHAR ItemId; // ID of the item to check UCHAR Quantity; // Quantity of the item to check } Keep_Item_ck2; This bytecode checks the specified item parameters.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{82} Sce_espr_task	03	<p>82 ?? ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x82 CHAR WorkKind; // WorkKind CHAR WorkNo; // WorkNo } Sce_espr_task; </pre> <p>This bytecode sets the specified espr task parameters.</p>	08-02-2024 Newly Added
{83} Plc_heal	01	<p>83++</p> <pre>typedef struct { UCHAR Opcode; // 0x83 } Plc_heal; </pre> <p>This bytecode heals the specified plc function.</p>	08-02-2024 Newly Added
{84} St_map_hint	02	<p>84 ?? ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x84 UCHAR DrModeTag; // DrModeTag } St_map_hint; </pre> <p>This bytecode sets the specified map hint parameters.</p>	08-02-2024 Newly Added
{85} Sce_em_pos_ck	06	<p>85 ?? ?? ?? ?? CO NT++</p> <pre>typedef struct { UCHAR Opcode; // 0x85 UCHAR Id; // Id UCHAR data1; // data1 UCHAR Att; // Att USHORT Flg; // Flg } Sce_em_pos_ck; </pre> <p>This bytecode checks the specified enemy position parameters.</p>	08-02-2024 Newly Added
{86} Poison_ck	01	<p>86++</p> <pre>typedef struct { UCHAR Opcode; // 0x86 } Poison_ck; </pre> <p>This bytecode checks the specified poison parameters.</p>	08-02-2024 Newly Added
{87} Poison_clr	01	<p>87++</p> <pre>typedef struct { UCHAR Opcode; // 0x87 } Poison_clr; </pre> <p>This bytecode clears the specified poison parameters.</p>	08-02-2024 Newly Added
{88} Sce_Item_lost2	03	<p>88 ID ??++</p> <pre>typedef struct { UCHAR Opcode; // 0x88 UCHAR ItemId; // ID of the item UCHAR Quantity; // Quantity of the item } Sce_Item_lost2; </pre> <p>This bytecode removes the specified item from the inventory.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{89} Evt_next2	01	<p>89++ typedef struct { UCHAR Opcode; // 0x89 } Evt_next2; This bytecode moves to the next event in the sequence.</p>	08-02-2024 Newly Added
{8A} Vib_set0	06	<p>8A 00 ?? ?? ?? ??++ typedef struct { UCHAR Opcode; // 0x8A UCHAR zAlign; // zAlign USHORT data0; // data0 USHORT data1; // data1 } Vib_set0; This bytecode sets the specified vibration parameters.</p>	08-02-2024 Newly Added
{8B} Vib_set1	06	<p>8B ID VA LU VA LU++ typedef struct { UCHAR Opcode; // 0x8B UCHAR Id; // Vibration ID USHORT Value1; // Vibration Value 1 USHORT Value2; // Vibration Value 2 } Vib_set1; This bytecode sets the specified vibration parameters.</p>	08-02-2024 Newly Added
{8C} Vib_fade_set	08	<p>8C 00 ?? ?? ?? ?? ??++ typedef struct { UCHAR Opcode; // 0x8C UCHAR zAlign; // zAlign UCHAR data0; // data0 UCHAR data1; // data1 USHORT data2; // data2 USHORT data3; // data3 } Vib_fade_set; This bytecode sets the specified vibration fade parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{8D} Item_aot_set2	24	<p>8D ID ?? ?? ?? ?? ?? XX XX ZZ ZZ WW WW DD DD ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++</p> <pre> typedef struct { UCHAR Opcode; // 0x8D UCHAR Aot; // Id UCHAR SCE; // Type (0x02) UCHAR SAT; // Atari UCHAR nFloor; // Height UCHAR Super; // Matrix SHORT X; // Position SHORT Z; // Position USHORT W; // Size USHORT D; // Size USHORT iItem; // Item Id USHORT nItem; // Item Amount USHORT Flag; // argv[1] for Flg_on() and Flg_ck() UCHAR MD1; // MD1 File Id - 0xFF if no MD1 UCHAR Action; // Predefined action taken to obtain UCHAR data16; // UCHAR data17; // } Item_aot_set2; </pre> <p>This bytecode sets the specified item AOT parameters.</p>	08-02-2024 Newly Added
{8E} Sce_em_set2	24	<p>8E 00 ID ?? ?? SI ZE ?? ?? ?? XX XX YY YY ZZ ZZ DY DY ?? ?? ?? ?? ??++</p> <pre> typedef struct { UCHAR Opcode; // 0x8E UCHAR zAlign; // Alignment byte, always zero UCHAR Aot; // UCHAR EMD; // USHORT Type; // UCHAR nFloor; // UCHAR SeType; // UCHAR ModelType; // UCHAR EmSetFlg; // SHORT X; // SHORT Y; // SHORT Z; // SHORT DirY; // USHORT Timer0; // USHORT Timer1; // USHORT data16; // } Sce_em_set2; </pre> <p>This bytecode sets the specified enemy parameters.</p>	08-02-2024 Newly Added

From:

<https://classicmodification.com/> - **Classic RE Modification**

Permanent link:

https://classicmodification.com/doku.php?id=re2_opcodes

Last update: **2024/08/04 06:23**

