

Instruction Name	Length	Example / Info	History
{00} Nop	01	<p>00++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 } Nop; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</p>	08-02-2024 Newly Added
{01} Evt_end	02	<p>01 00++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x01 UCHAR zAlign; // Always Zero } Evt_end; This bytecode ends the current Main/Sub script.</p>	08-02-2024 Newly Added
{02} Evt_next	01	<p>02++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x02 } Evt_next; This bytecode moves to the next event.</p>	08-02-2024 Newly Added
{03} Evt_chain	02	<p>03 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x03 UCHAR Data1; // Data } Evt_chain; This bytecode chains the next event.</p>	08-02-2024 Newly Added
{04} Evt_exec	04	<p>04 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x04 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Evt_exec; This bytecode executes the event.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{05} Evt_kill	02	<p>05 ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x05 UCHAR Data1; // Data } Evt_kill; This bytecode kills the event.</pre>	08-02-2024 Newly Added
{06} Ifel_ck	04	<p>06 00 ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x06 UCHAR zAlign; // 00 USHORT data2; // Data } Ifel_ck; This bytecode checks the condition of an If-Else block.</pre>	08-02-2024 Newly Added
{07} Else_ck	04	<p>07 00 ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x07 UCHAR zAlign; // 00 USHORT data2; // Data } Else_ck; This bytecode checks the condition of an Else block.</pre>	08-02-2024 Newly Added
{08} Endif	02	<p>08 00++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x08 UCHAR zAlign; // 00 } Endif; This bytecode ends an If-Else block.</pre>	08-02-2024 Newly Added
{09} Sleep	04	<p>09 ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x09 UCHAR Data1; // Data USHORT data2; // Data } Sleep; This bytecode pauses the event for a specified duration.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{0A} Sleeping	03	<p>0A ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0A USHORT data2; // Data } Sleeping; This bytecode sets the sleeping state.</p>	08-02-2024 Newly Added
{0B} Wsleep	01	<p>0B++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0B } Wsleep; This bytecode sets the wake sleep state.</p>	08-02-2024 Newly Added
{0C} Wsleeping	01	<p>0C++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0C } Wsleeping; This bytecode sets the wake sleeping state.</p>	08-02-2024 Newly Added
{0D} For	06	<p>0D 00 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0D UCHAR zAlign; // 00 SHORT data2; // Data USHORT data4; // Data } For; This bytecode starts a For loop.</p>	08-02-2024 Newly Added
{0E} Next	02	<p>0E 00++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0E UCHAR zAlign; // 00 } Next; This bytecode ends a For loop.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{0F} While	04	<p>0F 00 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x0F UCHAR zAlign; // 00 SHORT data1; // Data } While; This bytecode starts a While loop.</p>	08-02-2024 Newly Added
{10} Ewhile	02	<p>10 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x10 UCHAR data1; // Data } Ewhile; This bytecode ends a While loop.</p>	08-02-2024 Newly Added
{11} Do	04	<p>11 00 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x11 UCHAR zAlign; // 00 SHORT data2; // Data } Do; This bytecode starts a Do loop.</p>	08-02-2024 Newly Added
{12} Edwhile	02	<p>12 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x12 UCHAR data1; // Data } Edwhile; This bytecode ends a Do-While loop.</p>	08-02-2024 Newly Added
{13} Switch	04	<p>13 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x13 UCHAR data1; // Data USHORT data2; // Data } Switch; This bytecode starts a Switch statement.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{14} Case	06	<pre>14 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x14 UCHAR data1; // Data USHORT data2; // Data } Case; This bytecode defines a Case in a Switch statement.</pre>	08-02-2024 Newly Added
{15} Default	02	<pre>15 00++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x15 UCHAR zAlign; // 00 } Default; This bytecode defines the Default case in a Switch statement.</pre>	08-02-2024 Newly Added
{16} Eswitch	02	<pre>16 00++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x16 UCHAR zAlign; // 00 } Eswitch; This bytecode ends a Switch statement.</pre>	08-02-2024 Newly Added
{17} Goto	06	<pre>17 ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x17 UCHAR data1; // Data UCHAR data2; // Data UCHAR zAlign; // Alignment byte SHORT data4; // Data } Goto; This bytecode performs a Goto operation.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{18} Gosub	02	<p>18 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x18 UCHAR data1; // Data } Gosub; This bytecode performs a Gosub operation.</p>	08-02-2024 Newly Added
{19} Return	02	<p>19 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x19 UCHAR data[3]; // Data } Return; This bytecode returns from a subroutine.</p>	08-02-2024 Newly Added
{1A} Break	02	<p>1A ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x1A CHAR data1; // Data } Break; This bytecode breaks out of a loop.</p>	08-02-2024 Newly Added
{1B} For2	06	<p>1B 00 ?? ?? 00 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x1B UCHAR zAlign0; // 00 SHORT data2; // Data UCHAR zAlign1; // 00 UCHAR data5; // Data } For2; This bytecode starts a secondary For loop.</p>	08-02-2024 Newly Added
{1C} Break_point	01	<p>1C++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x1C } Break_point; This bytecode sets a breakpoint.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{1D} Work_copy	04	<p>1D ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x1D UCHAR Source; // Source UCHAR Destination; // Destination UCHAR Typecast; // Typecast } Work_copy; This bytecode copies work data.</pre>	08-02-2024 Newly Added
{1E} Nop1E	01	<p>1E++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x1E } Nop1E; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</pre>	08-02-2024 Newly Added
{1F} Nop1F	01	<p>1F++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x1F } Nop1F; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</pre>	08-02-2024 Newly Added
{20} Nop	01	<p>20++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x20 } Nop; This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</pre>	08-02-2024 Newly Added
{21} Ck	04	<p>21 ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x21 UCHAR Flag; // System_flg, etc UCHAR Id; // Bit UCHAR OnOff; // On/Off } Ck; This bytecode checks a flag.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{22} Set	04	<pre>22 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x22 UCHAR Flag; // System_flg, etc UCHAR Id; // Bit UCHAR OnOff; // On/Off } Set; This bytecode sets a flag.</pre>	08-02-2024 Newly Added
{23} Cmp	06	<pre>23 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x23 UCHAR Flag; // Flag UCHAR Operator; // Operator SHORT Value; // Value } Cmp; This bytecode compares values.</pre>	08-02-2024 Newly Added
{24} Save	04	<pre>24 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x24 UCHAR Destination; // Destination SHORT Source; // Source } Save; This bytecode saves data.</pre>	08-02-2024 Newly Added
{25} Work_copy_2	03	<pre>25 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x25 UCHAR Data1; // Data UCHAR Data2; // Data } Work_copy_2; This bytecode copies work data with two parameters.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{26} Calc	06	<p>26 ?? ?? ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x26 UCHAR Data1; // Data UCHAR Data2; // Data SHORT Data3; // Data } Calc; This bytecode performs a calculation.</pre>	08-02-2024 Newly Added
{27} Calc2	04	<p>27 ?? ?? ?? ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x27 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Calc2; This bytecode performs a secondary calculation.</pre>	08-02-2024 Newly Added
{28} Sce_rnd	01	<p>28++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x28 } Sce_rnd; This bytecode generates a random scenario.</pre>	08-02-2024 Newly Added
{29} Cut_chg	02	<p>29 ??++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x29 UCHAR Data1; // Data } Cut_chg; This bytecode changes the cutscene.</pre>	08-02-2024 Newly Added
{2A} Cut_old	01	<p>2A++</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x2A } Cut_old; This bytecode refers to an old cutscene.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{2F} Speed_set	04	<p>2F ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x2F UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Speed_set; This bytecode sets the speed.</p>	08-02-2024 Newly Added
{30} Add_speed	01	<p>30++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x30 } Add_speed; This bytecode adds speed.</p>	08-02-2024 Newly Added
{31} Add_aspeed	01	<p>31++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x31 } Add_aspeed; This bytecode adds angular speed.</p>	08-02-2024 Newly Added
{32} Pos_set	08	<p>32 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x32 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data } Pos_set; This bytecode sets the position.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{33} Dir_set	08	<pre>33 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x33 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data } Dir_set; This bytecode sets the direction.</pre>	08-02-2024 Newly Added
{34} Member_set	04	<pre>34 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x34 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Member_set; This bytecode sets a member.</pre>	08-02-2024 Newly Added
{35} Member_set2	03	<pre>35 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x35 UCHAR Data1; // Data UCHAR Data2; // Data } Member_set2; This bytecode sets a secondary member.</pre>	08-02-2024 Newly Added
{36} Se_on	12	<pre>36 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x36 UCHAR Data1[11]; // Data } Se_on; This bytecode turns on the SE (Sound Effect).</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{37} Sca_id_set	04	<pre> 37 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x37 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Sca_id_set; This bytecode sets the Sca ID. </pre>	08-02-2024 Newly Added
{38} Flr_set	03	<pre> 38 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x38 UCHAR Data1; // Data UCHAR Data2; // Data } Flr_set; This bytecode sets the floor. </pre>	08-02-2024 Newly Added
{39} Dir_ck	08	<pre> 39 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x39 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data } Dir_ck; This bytecode checks the direction. </pre>	08-02-2024 Newly Added
{3A} Sce_espr_on	16	<pre> 3A ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3A UCHAR Data1[15]; // Data } Sce_espr_on; This bytecode turns on the SCE ESPR. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{3B} Door_aot_set	32	<pre> 3B ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3B UCHAR Data1[31]; // Data } Door_aot_set; This bytecode sets a door AOT entry. </pre>	08-02-2024 Newly Added
{3C} Cut_auto	02	<pre> 3C ???++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3C UCHAR Data1; // Data } Cut_auto; This bytecode performs an automatic cut. </pre>	08-02-2024 Newly Added
{3D} Member_copy	03	<pre> 3D ?? ???++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3D UCHAR Data1; // Data UCHAR Data2; // Data } Member_copy; This bytecode copies a member. </pre>	08-02-2024 Newly Added
{3E} Member_cmp	06	<pre> 3E ?? ?? ?? ?? ?? ???++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3E UCHAR Data1; // Data UCHAR Data2; // Data USHORT Data3; // Data } Member_cmp; This bytecode compares a member. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{3F} Plc_motion	04	<pre> 3F ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x3F UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Plc_motion; This bytecode sets the motion for PLC. </pre>	08-02-2024 Newly Added
{40} Plc_dest	08	<pre> 40 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x40 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data } Plc_dest; This bytecode sets the destination for PLC. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{41} Plc_neck	10	<pre> 41 ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x41 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data UCHAR Data7; // Data UCHAR Data8; // Data UCHAR Data9; // Data } Plc_neck; This bytecode sets the neck for PLC. </pre>	08-02-2024 Newly Added
{42} Plc_ret	01	<pre> 42++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x42 } Plc_ret; This bytecode returns from a PLC operation. </pre>	08-02-2024 Newly Added
{43} Plc_flg	04	<pre> 43 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x43 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Plc_flg; This bytecode sets a flag for PLC. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{44} Sce_em_set	22	<pre> 44 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x44 UCHAR Data1[21]; // Data } Sce_em_set; This bytecode sets the scenario for EM. </pre>	08-02-2024 Newly Added
{45} Col_chg_set	05	<pre> 45 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x45 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data } Col_chg_set; This bytecode sets the color change. </pre>	08-02-2024 Newly Added
{46} Aot_reset	10	<pre> 46 ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x46 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data UCHAR Data7; // Data UCHAR Data8; // Data UCHAR Data9; // Data } Aot_reset; This bytecode resets the AOT. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{47} Aot_on	02	<p>47 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x47 UCHAR Data1; // Data } Aot_on; This bytecode turns on the AOT.</p>	08-02-2024 Newly Added
{48} Super_set	16	<p>48 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x48 UCHAR Data1[15]; // Data } Super_set; This bytecode sets a super parameter.</p>	08-02-2024 Newly Added
{49} Super_reset	08	<p>49 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x49 UCHAR Data1[7]; // Data } Super_reset; This bytecode resets a super parameter.</p>	08-02-2024 Newly Added
{4A} Plc_gun	02	<p>4A ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4A UCHAR Data1; // Data } Plc_gun; This bytecode sets the PLC gun.</p>	08-02-2024 Newly Added
{4B} Cut_replace	03	<p>4B ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4B UCHAR Data1; // Data UCHAR Data2; // Data } Cut_replace; This bytecode replaces a cutscene.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{4C} Sce_espr_kill	05	<p>4C ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4C UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data } Sce_espr_kill; This bytecode kills the SCE ESPR.</p>	08-02-2024 Newly Added
{4D} Door_model_set	22	<p>4D ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4D UCHAR Data1[21]; // Data } Door_model_set; This bytecode sets the door model.</p>	08-02-2024 Newly Added
{4E} Item_aot_set	22	<p>4E ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4E UCHAR Data1[21]; // Data } Item_aot_set; This bytecode sets an item AOT.</p>	08-02-2024 Newly Added
{4F} Sce_key_ck	04	<p>4F ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x4F UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Sce_key_ck; This bytecode checks the SCE key.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{50} Sce_trg_ck	04	<pre>50 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x50 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Sce_trg_ck; This bytecode checks the SCE trigger.</pre>	08-02-2024 Newly Added
{51} Sce_bgm_control	06	<pre>51 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x51 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data USHORT Data4; // Data } Sce_bgm_control; This bytecode controls the SCE BGM.</pre>	08-02-2024 Newly Added
{52} Sce_espr_control	06	<pre>52 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x52 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data USHORT Data4; // Data } Sce_espr_control; This bytecode controls the SCE ESPR.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{53} Sce_fade_set	06	<pre>53 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x53 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data USHORT Data4; // Data } Sce_fade_set; This bytecode sets the SCE fade.</pre>	08-02-2024 Newly Added
{54} Sce_espr3d_on	22	<pre>54 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x54 UCHAR Data1[21]; // Data } Sce_espr3d_on; This bytecode turns on the SCE ESPR 3D.</pre>	08-02-2024 Newly Added
{55} Member_calc	06	<pre>55 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x55 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data USHORT Data4; // Data } Member_calc; This bytecode performs a member calculation.</pre>	08-02-2024 Newly Added
{56} Member_calc2	04	<pre>56 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x56 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Member_calc2; This bytecode performs a secondary member calculation.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{57} Sce_bgmtbl_set	08	<pre>57 ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x57 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data } Sce_bgmtbl_set; This bytecode sets the SCE BGM table.</pre>	08-02-2024 Newly Added
{58} Plc_rot	04	<pre>58 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x58 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Plc_rot; This bytecode rotates the PLC.</pre>	08-02-2024 Newly Added
{59} Xa_on	04	<pre>59 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x59 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Xa_on; This bytecode turns on the XA sound.</pre>	08-02-2024 Newly Added
{5A} Weapon_chg	02	<pre>5A ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5A UCHAR Data1; // Data } Weapon_chg; This bytecode changes the weapon.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{5B} Plc_cnt	02	5B ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5B UCHAR Data1; // Data } Plc_cnt; This bytecode counts the PLC.	08-02-2024 Newly Added
{5C} Sce_shake_on	03	5C ?? ?++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5C UCHAR Data1; // Data UCHAR Data2; // Data } Sce_shake_on; This bytecode turns on the SCE shake effect.	08-02-2024 Newly Added
{5D} Mizu_div_set	02	5D ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5D UCHAR Data1; // Data } Mizu_div_set; This bytecode sets the Mizu division.	08-02-2024 Newly Added
{5E} Keep_Item_ck	02	5E ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5E UCHAR Data1; // Data } Keep_Item_ck; This bytecode checks the kept item.	08-02-2024 Newly Added
{5F} Xa_vol	02	5F ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x5F UCHAR Data1; // Data } Xa_vol; This bytecode sets the XA volume.	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{60} Kage_set	14	<pre>60 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x60 UCHAR Data1[13]; // Data } Kage_set; This bytecode sets the Kage parameter.</pre>	08-02-2024 Newly Added
{61} Cut_be_set	04	<pre>61 ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x61 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Cut_be_set; This bytecode sets the cutscene BE parameter.</pre>	08-02-2024 Newly Added
{62} Sce_Item_lost	02	<pre>62 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x62 UCHAR Data1; // Data } Sce_Item_lost; This bytecode sets the SCE item lost parameter.</pre>	08-02-2024 Newly Added
{63} Plc_gun_eff	01	<pre>63++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x63 } Plc_gun_eff; This bytecode sets the PLC gun effect.</pre>	08-02-2024 Newly Added
{64} Sce_espr_on2	16	<pre>64 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x64 UCHAR Data1[15]; // Data } Sce_espr_on2; This bytecode turns on the SCE ESPR 2.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{65} Sce_espr_kill2	02	<p>65 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x65 UCHAR Data1; // Data } Sce_espr_kill2; This bytecode kills the SCE ESPR 2.</p>	08-02-2024 Newly Added
{66} Plc_stop	01	<p>66++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x66 } Plc_stop; This bytecode stops the PLC.</p>	08-02-2024 Newly Added
{67} Aot_set_4p	28	<p>67 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x67 UCHAR Data1[27]; // Data } Aot_set_4p; This bytecode sets an AOT (Active Object Table) entry with 4 parameters.</p>	08-02-2024 Newly Added
{68} Door_aot_set_4p	40	<p>68 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x68 UCHAR Data1[39]; // Data } Door_aot_set_4p; This bytecode sets a door AOT entry with 4 parameters.</p>	08-02-2024 Newly Added
{69} Item_aot_set_4p	30	<p>69 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x69 UCHAR Data1[29]; // Data } Item_aot_set_4p; This bytecode sets an item AOT entry with 4 parameters.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{6A} Light_pos_set	06	<pre> 6A ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6A UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Light_pos_set; This bytecode sets the light position. </pre>	08-02-2024 Newly Added
{6B} Light_kido_set	04	<pre> 6B ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6B UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Light_kido_set; This bytecode sets the light intensity (Kido). </pre>	08-02-2024 Newly Added
{6C} Rbj_reset	01	<pre> 6C++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6C } Rbj_reset; This bytecode resets the RBJ. </pre>	08-02-2024 Newly Added
{6D} Sce_scr_move	04	<pre> 6D ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6D UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Sce_scr_move; This bytecode moves the SCE screen. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{6E} Parts_set	06	<pre> 6E ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6E UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Parts_set; This bytecode sets parts. </pre>	08-02-2024 Newly Added
{6F} Movie_on	02	<pre> 6F ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x6F UCHAR Data1; // Data } Movie_on; This bytecode turns on the movie. </pre>	08-02-2024 Newly Added
{70} Splc_ret	01	<pre> 70++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x70 } Splc_ret; This bytecode returns from a SPLC operation. </pre>	08-02-2024 Newly Added
{71} Splc_sce	01	<pre> 71++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x71 } Splc_sce; This bytecode sets the SPLC scene. </pre>	08-02-2024 Newly Added
{72} Super_on	16	<pre> 72 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x72 UCHAR Data1[15]; // Data } Super_on; This bytecode turns on the super mode. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{77} Sce_line_start	04	<pre>77 ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x77 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data } Sce_line_start; This bytecode starts the SCE line.</pre>	08-02-2024 Newly Added
{78} Sce_line_main	06	<pre>78 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x78 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Sce_line_main; This bytecode defines the main SCE line.</pre>	08-02-2024 Newly Added
{79} Sce_line_end	01	<pre>79++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x79 } Sce_line_end; This bytecode ends the SCE line.</pre>	08-02-2024 Newly Added
{7A} Sce_parts_bomb	16	<pre>7A ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7A UCHAR Data1[15]; // Data } Sce_parts_bomb; This bytecode sets the SCE parts bomb.</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{7B} Sce_parts_down	16	<pre> 7B ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7B UCHAR Data1[15]; // Data } Sce_parts_down; This bytecode sets the SCE parts down. </pre>	08-02-2024 Newly Added
{7C} Light_color_set	06	<pre> 7C ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7C UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Light_color_set; This bytecode sets the light color. </pre>	08-02-2024 Newly Added
{7D} Light_pos_set2	06	<pre> 7D ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7D UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Light_pos_set2; This bytecode sets the secondary light position. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{7E} Light_kido_set2	06	<pre> 7E ?? ?? ?? ?? ?? ++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7E UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Light_kido_set2; This bytecode sets the secondary light intensity (Kido). </pre>	08-02-2024 Newly Added
{7F} Light_color_set2	06	<pre> 7F ?? ?? ?? ?? ?? ++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x7F UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Light_color_set2; This bytecode sets the secondary light color. </pre>	08-02-2024 Newly Added
{80} Se_vol	02	<pre> 80 ++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x80 UCHAR Data1; // Data } Se_vol; This bytecode sets the SE volume. </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{81} Sce_Item_cmp	03	<p>81 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x81 UCHAR Data1; // Data UCHAR Data2; // Data } Sce_Item_cmp; This bytecode compares an SCE item.</p>	08-02-2024 Newly Added
{82} Sce_espr_task	03	<p>82 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x82 UCHAR Data1; // Data UCHAR Data2; // Data } Sce_espr_task; This bytecode sets an SCE ESPR task.</p>	08-02-2024 Newly Added
{83} Plc_heal	01	<p>83++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x83 } Plc_heal; This bytecode heals the PLC.</p>	08-02-2024 Newly Added
{84} St_map_hint	02	<p>84 ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x84 UCHAR Data1; // Data } St_map_hint; This bytecode sets the ST map hint.</p>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{85} Sce_em_pos_ck	06	<pre>85 ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x85 UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Sce_em_pos_ck; This bytecode checks the SCE EM position.</pre>	08-02-2024 Newly Added
{86} Poison_ck	01	<pre>86++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x86 } Poison_ck; This bytecode checks for poison.</pre>	08-02-2024 Newly Added
{87} Poison_clr	01	<pre>87++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x87 } Poison_clr; This bytecode clears poison.</pre>	08-02-2024 Newly Added
{88} Sce_Item_lost2	03	<pre>88 ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x88 UCHAR Data1; // Data UCHAR Data2; // Data } Sce_Item_lost2; This bytecode sets the SCE item lost parameter (version 2).</pre>	08-02-2024 Newly Added
{89} Evt_next2	01	<pre>89++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x89 } Evt_next2; This bytecode moves to the next event (version 2).</pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{8A} Vib_set0	06	<pre> 8A ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x8A UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Vib_set0; This bytecode sets vibration parameters (set 0). </pre>	08-02-2024 Newly Added
{8B} Vib_set1	06	<pre> 8B ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x8B UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data } Vib_set1; This bytecode sets vibration parameters (set 1). </pre>	08-02-2024 Newly Added

Instruction Name	Length	Example / Info	History
{8C} Vib_fade_set	08	<pre> 8C ?? ?? ?? ?? ?? ?? ?? ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x8C UCHAR Data1; // Data UCHAR Data2; // Data UCHAR Data3; // Data UCHAR Data4; // Data UCHAR Data5; // Data UCHAR Data6; // Data UCHAR Data7; // Data } Vib_fade_set; This bytecode sets the vibration fade parameters.</pre>	08-02-2024 Newly Added
{8D} Item_aot_set2	24	<pre> 8D ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x8D UCHAR Data1[23]; // Data } Item_aot_set2; This bytecode sets an item AOT entry (version 2).</pre>	08-02-2024 Newly Added
{8E} Sce_em_set2	24	<pre> 8E ??++ typedef struct { // Ptr // Description UCHAR Opcode; // 0x8E UCHAR Data1[23]; // Data } Sce_em_set2; This bytecode sets the scenario for EM (version 2).</pre>	08-02-2024 Newly Added

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

Permanent link:
https://classicremodification.com/doku.php?id=re2_opcodes&rev=1722656399

Last update: **2024/08/02 20:39**

