AidF AIBus

# Basics

AIBus works similar to BMW’s I-BUS, but is balanced and runs at 115200. The structure is virtually identical.

The structure is as follows:

* Source Device ID: 1B, ID of the source device.
* Length: 1B, includes all following bytes.
* Destination Device ID: 1B, ID of the destination device (0xFF if broadcast).
* Data: Length given in the Length bytes, message.
* XOR Checksum: 1B, a simple XOR of all bytes in the message.

# Radio

1. ID 0x10
2. Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 10 | 1. 3 | 1. FF | 1. 04 E6 10 | 1. Sent to all audio devices when the radio is first connected. May also be sent to individual sources as a “repeat.” |
| 1. Any | 1. 3 | 1. 10 | 1. 01 01 xx | 1. Audio device “handshake.” Similar to Honda’s. Byte xx indicates device ID. May have a fourth byte to indicate subsource count. |
| 1. Any | 1. 2 | 1. 10 | 1. 01 02 | 1. Similar audio device “handshake,” indicates that the device has already made contact but needs to add extra “sub-sources.” (e.g. XM1, XM2) |
| 1. Any | 1. Varies | 1. 10 | 1. 01 23 nn [...] | 1. Name of the audio device, sub-source nn. |
| 1. 10 | 1. 3 | 1. Any | 1. 05 xx 02 | 1. Handshake reply from the radio. |
| 1. 10 | 1. 3 or 4 | 1. Any | 1. 40 10 xx nn | 1. Function change, xx indicates the ID of the device that is selected. A device can also send this message to request control (e.g. if a tape is loaded), the radio will send it back if it is accepted. If the source has multiple modes, nn is the mode. |
| 1. 10 | 1. 3 | 1. Any | 1. 40 01 xx | 1. Tells the source that device xx has text control. This control is automatically relinquished when the source receives a 40 10 message. |
| 1. Any | 1. 3 or 4 | 1. 10 | 1. 10 10 xx nn | 1. Request for control, i.e. if a tape or disc is loaded. |
| 1. Any | 1. 3 | 1. 10 | 1. 11 10 xx | 1. Acknowledgment by device xx. |
| 1. 10 | 1. 3 | 1. Any | 1. 70 10 xx | 1. Redundant heartbeat message to let device xx know it is still the selected source. |
| 1. 01, 57 |  | 1. 10, 57, 3B | 1. 1D nn | 1. Set whether this device should be sending clock data. If nn is 00, stop sending clock messages. |
| 1. 10 | 1. 4 | 1. 57 | 1. 1F 11 60 57 | 1. Query to the CANslator for where audio data should be written to the iMID screen. |
| 1. 57 | 1. 4 | 1. 10 | 1. 1F 11 10 nn | 1. CANslator reporting where audio data should be written. This may be sent with or without a query. States of nn:  * 00: Do not write audio data. * 01: Top bar. * 02: Top half of screen. * 04: Bottom half of screen. * 06: Full screen. |

# CD Changer

1. ID 0x06
2. Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 06 | 1. 8 | 1. 10, 11 | 1. 39 00 pp 00 3F 00 dd tt | 1. Disc and track, indicated by dd and tt, respectively. Whether the CD is playing is indicated by pp, pp = 02 if not playing, 09 if playing, 08 if seeking. |
| 1. 06 | 1. 5 | 1. 10, 11 | 1. 3B 00 00 ss ss | 1. Current track time. Bytes ss ss are a 16-bit representation of the total seconds. |
| 1. 06 | 1. Varies | 1. 10, 11 | 1. 23 61 bb [...] | 1. Song title in ASCII. Byte bb indicates the number of bits per character, 1 or 2. |
| 1. 06 | 1. Varies | 1. 10, 11 | 1. 23 62 bb [...] | 1. Artist name in ASCII, same format as title. |
| 1. 06 | 1. Varies | 1. 10, 11 | 1. 23 63 bb [...] | 1. Album name in ASCII, same format as title. |
| 1. 10 | 1. 3 | 1. 06 | 1. 30 00 00 | 1. Query the CD player for status. |
| 1. 10 | 1. 3 | 1. 06 | 1. 30 01 00 | 1. Trigger pause/resume. |
| 1. 10 | 1. 3 | 1. 06 | 1. 30 03 00 | 1. Trigger normal play (if FF/FR is in use). |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 04 00 | 1. Trigger fast forward. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 04 01 | 1. Trigger fast reverse. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 06 0n | 1. Go to disc n. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 06 10 | 1. Go to the previous disc. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 06 20 | 1. Go to the next disc. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 06 0n | 1. Go to disc n. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 07 nn | 1. Toggle scan mode:  * 00: Off * 01: On * 10: Toggle |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 08 nn | 1. Toggle random mode:  * 00: Off * 01: Random Disc * 02: Random All * 10: Toggle |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 09 nn | 1. Toggle repeat mode:  * 00: Off * 01: Repeat Track * 02: Repeat Disc * 10: Toggle |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 0A 00 | 1. Go to the next track. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 0A 01 | 1. Go to the previous track. |
| 1. 10 | 1. 3 | 1. 06 | 1. 38 0B nn | 1. Go to track nn. |

# Nav Computer

1. ID 0x01
2. Audio Screen Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 10 (or audio source) | 1. 3 | 1. 01 | 1. 20 60 nn | 1. Clear area nn. Use FF to clear all areas. Note, for all 2x messages below, follow with ||10 to refresh the screen. |
| 1. 10 (or audio source) | 1. 3 | 1. 01 | 1. 20 61 nn | 1. Clear subtitle area nn. Use FF to clear all subtitle areas. |
| 1. 10 (or audio source) | 1. 3 | 1. 01 | 1. 20 62 nn | 1. Clear function key nn. Use FF to clear all function keys. |
| 1. 10 (or audio source) | 1. 3 | 1. 01 | 1. 20 6B nn | 1. Clear button nn. Use FF to clear all buttons. |
| 1. 10 (or audio source) | 1. 2 | 1. 01 | 1. 20 6F | 1. Clear the entire audio screen. |
| 1. 10 (or audio source) | 1. Varies | 1. 01 | 1. 23 60 nn [...] | 1. Write text to the main area indicated by nn. Only valid if the audio screen is visible. |
| 1. 10 (or audio source) | 1. Varies | 1. 01 | 1. 23 61 nn [...] | 1. Write text to the subtitle area indicated by nn. Only valid if the audio screen is visible. |
| 1. 10 (or audio source) | 1. Varies | 1. 01 | 1. 23 62 nn [...] | 1. Write text to the bottom function key nn. Second nibble indicates button 1-6. |
| 1. 10 (or audio source) | 1. Varies | 1. 01 | 1. 23 6B nn [...] | 1. Write text to on-screen button 1-3. If the text is cleared, the button is unused. |
| 1. 10 (or audio source) | 1. 2 | 1. 01 | 1. 23 6F | 1. Refresh the audio screen. |
| 1. Any | 1. 4 | 1. 01 | 1. 51 F0 rr rr | 1. Fill the screen with color rr. |

Drawing messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Any | 1. 12 | 1. 01 | 1. 51 02 x1 x1 y1 y1 x2 x2 y2 y2 rr rr | 1. Draw a line with points at x1, y1 and x2, y2, with color rr. |
| 1. Any | 1. 12 | 1. 01 | 1. 51 04 xx xx yy yy ww ww hh hh rr rr | 1. Draw a rectangle at xx, yy; with width and height ww, hh, of color rr. |
| 1. Any | 1. 12 | 1. 01 | 1. 51 F4 xx xx yy yy ww ww hh hh rr rr | 1. Draw a filled rectangle as above. |
| 1. Any | 1. 10 | 1. 01 | 1. 51 01 xx xx yy yy rr rr cc cc | 1. Draw a circle at xx, yy; with radius rr, of color cc. |
| 1. Any | 1. 10 | 1. 01 | 1. 51 F1 xx xx yy yy rr rr cc cc | 1. Draw a filled circle as above. |
| 1. Any | 1. 16 | 1. 01 | 1. 51 03 x1 x1 y1 y1 x2 x2 y2 y2 x3 x3 y3 y3 cc cc | 1. Draw a triangle at points x1, y1; x2, y2; and x3, y3; of color cc. |
| 1. Any | 1. 16 | 1. 01 | 1. 51 F3 x1 x1 y1 y1 x2 x2 y2 y2 x3 x3 y3 y3 cc cc | 1. Draw a filled triangle as above. |
| 1. Any | 1. 16 | 1. 01 | 1. 51 05 xx xx yy yy ww ww hh hh cc cc | 1. Draw an ellipse at xx, yy; with width and height ww, hh, of color cc. |
| 1. Any | 1. 16 | 1. 01 | 1. 51 F5 xx xx yy yy ww ww hh hh cc cc | 1. Draw a filled ellipse as above. |

1. Menu Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. Any | 1. Varies | 1. 01 | 1. 2B 50 nn ll xx xx yy yy ww ww hh hh […] | 1. Create a new menu with nn rows. Bit 7 is 1 if the menu “loops.” Menu length is defined by ll, x and y coordinates are given by xx and yy. Bytes ww and hh define width and height of each entry. May be followed by the menu title or subtitle. |
| 1. Any | 1. Varies | 1. 01 | 1. 2B 51 nn […] | 1. Add menu entry at position nn. If the menu has already been defined, this changes the entry. |
| 1. Any | 1. 3 | 1. 01 | 1. 2B 52 nn | 1. Menu is complete, draw it with nn selected. |
| 1. Any | 1. Varies | 1. 01 | 1. 2B 53 […] | 1. Change the menu title. |
| 1. Any | 1. Varies | 1. 01 | 1. 2B 5A nn ll xx xx yy yy ww ww hh hh […] | 1. Create an audio-related menu defined as above. Note that this does not create a menu to be shown on the normal audio screen, it creates a setting overlay menu. |
| 1. Any | 1. 2 | 1. 01 | 1. 2B 40 | 1. Clear the menu. Clear an audio menu if 4A. |
| 1. Any | 1. 2 | 1. 01 | 2B 42 | 1. Query to the nav computer regarding whether a menu can be drawn. |
| 1. 01 | 1. 2 | 1. Any | 1. 2B 40 | 1. The menu was cleared, or a menu cannot be drawn. |
| 1. 01 | 1. 2 | 1. Any | 1. 2B 42 | 1. A menu can be drawn, sent after the destination device sends the same message. |
| 1. 01 | 1. 2 | 1. Any | 1. 2B 45 | 1. Request for the settings menu. |
| 1. 10 or 1 | 1. 2 | 1. Any Source | 1. 2B 4A | 1. Request for the audio menu. Typically, this is sent from 1 to 10, then 10 to the source. |
| 1. 01 | 1. 3 | 1. Any | 1. 2B 60 nn | 1. Entry nn was selected. |
| 1. 01 | 1. 3 | 1. Any | 1. 2B 6A nn | 1. Main audio screen entry nn was selected. |

# Screen

ID 0x07

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 07 | 1. 3 | 1. 01 | 1. 30 nn tt | 1. Push or release button nn. Byte tt defines behavior as follows:  * 00: Pressed * 40: Held * 80: Released   Button Definitions:   * 06: Volume Knob * 07: Navigation Knob * 0A: Voice Command * 0B: Phone Send * 0C: Phone End * 11: Preset 1 * 12: Preset 2 * 13: Preset 3 * 14: Preset 4 * 15: Preset 5 * 16: Preset 6 * 20: Home * 21: Eject * 23: Source * 24: Skip Down * 25: Skip Up * 26: Audio * 27: Back * 28: Up * 29: Down * 2A: Left * 2B: Right * 30: FM * 31: AM * 32: Tape * 33: CD * 34: Aux * 35: XM * 36: FM/AM * 37: Tape/CD * 38: Media * 39: Scan * 44: Rewind * 45: FF * 46: Direction * 47: NR * 50: Phone * 51: Menu * 52: Tone * 53: Info * 54: Clock * 55: Map |
| 1. 07 | 1. 3 | 1. 01 | 1. 32 nn dt | 1. Knob turned. Byte nn = 06 if volume knob, 07 if navigation knob. Nibble t indicates number of steps, nibble d is 0 if the knob is rotated to the left, 1 if rotated to the right. |
| 1. 07 | 1. 5 | 1. 01 | 1. 35 xx xx yy yy | 1. Touch event at xx yy. |
| 1. 07 | 1. 2 | 1. 01 | 1. 38 nn | 1. Screen status. Byte nn = 00 if screen is closed, 01 if screen is opening, 02 if open, 03 if closing. FF if the screen cannot move. |
| 1. 01, 10 | 1. 2 | 1. 07 | 1. 38 nn | 1. Byte nn = 0, close the screen. If 2, open the screen. |
| 1. 07 | 1. 4 | 1. 01, 10 | 1. 47 xx yy cc | 1. Auxiliary screen of dimensions xx yy is present. Byte cc is formatted as follows:  * Bit 4: RGB Supported * Bit 3: Negative Mode * Bit 2: Dual Mode Supported |
| 1. 01, 10 | 1. Varies | 1. 07 | 1. 27 hh vv […] | 1. Print message to the auxiliary screen in standard monospace at horizontal indentation hh and vertical indentation vv. |
| 1. 01, 10 | 1. Varies | 1. 07 | 1. 2E ss […] | 1. Print message to the auxiliary screen in segmented form. Start at segment ss. |
| 1. 01 | 1. 2 | 1. 07 | 1. 10 pp | 1. Turn off display if pp is 00, on if pp is 11. |
| 1. Any | 1. 3 | 1. FF | 1. 77 ss nn | 1. A “Status” message sent every 5s. If receiver ID is FF, acknowledgment is not required. If this message stops, all screen controls default to the navigation computer.   If nn is ||20, audio controls (volume, preset buttons, source button, and all source controls unless they are requested by another device) are passed from the screen to the sending device (address ss). If ||80, “source” controls are passed to the device (seek/skip, direction control, and rewind/FF, and info). If ||10, all screen controls are passed (can only be sent by the radio). |
| 1. 7 | 1. Varies | 1. Any | 1. 31 30 nn […] | 1. A list of all buttons present on the screen. The bytes sent are the bytes that the buttons would trigger. All devices that can send messages to the screen can receive this message.   Byte nn is ||2 if there is a selection wheel present, ||4 if a volume knob is present, ||8 if a tuning knob is present, and ||1 if touch capabilities are present. |
| 1. Any | 1. 2 | 1. 7 | 1. 31 30 | 1. A request for the screen buttons. |

# Custom iMID

ID 0x11

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 11 | 1. 4 | 1. Any | 1. 3B 23 ww hh | 1. Announcement that a custom text field is available, of ww characters per line and hh lines. This message can be sent to FF. |
| 1. 11 | 1. Varies | 1. Any | 1. 3B 57 […] | 1. Announcement that an OEM text field is available. This is followed by the sources available in AIBus form (e.g. 0x10 for radio, etc.). |
| 1. Any | 1. 3 | 1. 11 | 1. 04 E6 3B | 1. Request for data sent with 3B. |
| 1. Any | 1. Varies | 1. 11 | 1. 20 60 nn […] | 1. Clear rows indicated by nn. There may be multiple. |
| 1. Any | 1. Varies | 1. 11 | 1. 23 60 xx yy […] | 1. Write a custom message at space xx and line yy. |
| 1. 10 | 1. 4 | 1. 11 | 1. 40 10 ss uu | 1. Change OEM display to source ss, subsource uu. |
| 1. 11 | 1. 2 | 1. 10 (or source) | 1. 60 11 | 1. A request for the radio to send the source name to the IMID. |
| 1. 11 | 1. 2 | 1. 10 (or source) | 1. 60 10 | 1. A request for the radio to send OEM source data to the IMID. |
| 1. 10 | 1. 6 | 1. 11 | 1. 67 rr rr pp ss hh | 1. FM or AM frequency rr, decimal point position pp. Byte ss is defined as follows:  * 0x: Mono * | 1x: Stereo * | 2x: HD1 * | 4x: HD2 * | 6x: HD3 * | 8x: Acquiring HD   First nibble defines preset.  Byte hh indicates the preceding letter of the frequency in Hz (e.g. 0x4D for M, 0x6B for k). |
| 1. 10 | 1. Varies | 1. 11 | 1. 63 61 […] | 1. RDS text. |

# Tape Deck

ID 0x13

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source** | **Length** | **Destination** | **Data Example** | **Interpretation** |
| 13 | 4 | 10 | 31 pp dd nn | Play status. Byte pp is defined as follows:   * 02 = Play * 05 = Rewind * 04 = FF * 07 = Reverse Search * 06 = Forward Search * 10 = Loading * 11 = Ejecting * 00 = Tape Out   Byte dd is defined bitwise:   * Bit 6: Reverse Mode * Bit 5: Repeat On * Bit 1: NR On * Bit 0: CrO2 tape   Byte nn indicates the number of tracks that are being searched. Only read if pp = 07 or 06. |
| 1. 10 | 1. 2 | 1. 13 | 1. 28 00 | 1. Query for tape deck status. |
| 1. 10 | 1. 2 | 1. 13 | 1. 28 01 | 1. Trigger direction change. |
| 1. 10 | 1. 3 | 1. 13 | 1. 28 04 00 | 1. Toggle fast forward. |
| 1. 10 | 1. 3 | 1. 13 | 1. 28 04 01 | 1. Toggle fast reverse. |
| 1. 10 | 1. 4 | 1. 13 | 1. 28 07 00 nn | 1. Toggle forward search. Byte nn indicates the number of tracks that are being searched. |
| 1. 10 | 1. 4 | 1. 13 | 1. 28 07 01 nn | 1. Toggle reverse search. Byte nn indicates the number of tracks that are being searched. |
| 1. 10 | 1. 2 | 1. 13 | 1. 28 09 | 1. Toggle repeat mode. |
| 1. 10 | 1. 2 | 1. 13 | 1. 28 0A | 1. Toggle NR. |

# Sirius Tuner

ID 0x19

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 19 | 1. 5 | 1. 10, 11 | 1. 39 00 hh hh pp | 1. The tuner is set to channel hh and preset pp. If pp|=0x80, the signal was lost. If pp|=0x40, the channel is not available. |
| 1. 19 | 1. Varies | 1. 10, 11 | 1. 23 61 bb [...] | 1. Song title in ASCII. Byte bb indicates the number of bits per character, 1 or 2. |
| 1. 19 | 1. Varies | 1. 10, 11 | 1. 23 62 bb [...] | 1. Artist name in ASCII, same format as title. |
| 1. 19 | 1. Varies | 1. 10, 11 | 1. 23 63 bb [...] | 1. Channel name in ASCII, same format as title. |
| 1. 19 | 1. Varies | 1. 10, 11 | 1. 23 64 bb [...] | 1. Genre in ASCII, same format as title. |
| 1. 10 | 1. 2 | 1. 19 | 1. 30 00 | 1. Request for tuner status. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 04 00 | 1. Go to the next channel. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 04 01 | 1. Go to the previous channel. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 05 hh hh | 1. Go to channel hh. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 06 pp | 1. Go to preset channel pp. Numbers higher than 6 can be used for XM2. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 07 00 | 1. Increment the preset channel. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 07 01 | 1. Decrement the preset channel. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 0A 00 | 1. Go to the next category. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 0A 01 | 1. Go to the previous category. |
| 1. 10 | 1. 3 | 1. 19 | 1. 38 16 pp | 1. Save current channel as preset pp. |

# CANslator

ID 0x57

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source** | **Length** | **Destination** | **Data Example** | **Interpretation** |
| 57 | 3 | FF | A1 02 kk | Key position kk. Defined as follows:   * 00 = Off * 01 = Acc 1 * 02 = Acc 2 * 03 = Cranking * 04 = Engine On   Byte kk is ||10 if the key (or fob) is in the ignition, ||20 if the car has pushbutton start. |
| 57 | 4 | FF | A1 10 ii nn | Illumination brightness, sent every few seconds (or upon change) if the appropriate CAN message is present. Brightness is given as ii, nn is defined as 00 if the system is in day mode, 80 if in night mode; ||1 if illumination should be on. |
| 57, 3B, 10 | 6 | FF | A1 1F 01 hh mm ss | Time. Hour, minute, and second. Always 24h. |
| 57, 3B, 10 | 7 | FF | A1 1F 02 yy yy mm dd | Date. Year, month, date. |
| 57 | Varies | FF | A1 1F 03 nn tt […] | Outside temperature. Byte nn defines units (bit 7 is high if Fahrenheit), number of decimal places (bits 6-4), and number of bytes (bits 3-0). |
| 57 | Varies | FF | A1 1F 04 nn tt […] | Vehicle speed. Byte nn defines units (bit 7 is high if mph), number of decimal places (bits 6-4), and number of bytes (bits 3-0). |
| 57 | 3 | FF | A1 43 dd | Door status. Byte dd is defined as follows:   * 20: Hood * 10: Trunk/Liftgate * 08: Driver Door * 04: Front Passenger Door * 02: Left Rear Door * 01: Right Rear Door |
| 01 | 2 | 57 | 45 nn | Request for vehicle information.  Byte nn is defined as follows:   * 01: Fuel Economy * 02: Odometer * 03: Outside Temperature * 04: Vehicle Speed |
| 57 | Varies | 01 | 46 nn yy […] | Vehicle information with byte nn defined as above. Placed in grid yy- column is defined by bit 7. Row is defined by remaining bits. Contains the string that will be printed to screen. |
| 57 | 1 | 01 | 4F | Refresh printed data. |

# Steering Wheel Controls

ID 0x6F

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Source** | 1. **Length** | 1. **Destination** | 1. **Data Example** | 1. **Interpretation** |
| 1. 6F | 1. 3 | 1. 10 | 1. 30 nn tt | 1. Push or release audio button nn, formatted exactly as from ID 7, with the following exceptions: 2. Volume control: nn=6, tt |= 1 if volume up, tt |= 2 if volume down. |

# Android Auto Pi

ID 0x8E

Messages:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source** | **Length** | **Destination** | **Data Example** | **Interpretation** |
| 01 | 3 | 8E | 48 8E nn | Turn off (nn = 0) or turn on (nn = 1) the AMirror interface. |
| 8E | 3 | 01 | 48 01 nn | Same as above. |
| 01 | Varies | 8E | 52 […] | Send a BASH command to the Pi. |
| 8E | Varies | 01 | 52 […] | BASH command from the Pi. |
| 01 | 5 | 8E | 60 20 rr gg bb | Change background color to rgb. Normally sent when the Teensy display switches from day to night or the user changes the color. |
| 01 | 5 | 8E | 60 21 rr gg bb | Change text color to rgb. |
| 01 | 5 | 8E | 60 22 rr gg bb | Change header color to rgb. |
| 01 | 5 | 8E | 60 23 rr gg bb | Change selection rectangle color to rgb. |
| 01 | 5 | 8E | 60 24 rr gg bb | Change border color to rgb. |
| 8E | 2 | 01, 10, 57 | 30 pp | Type of phone connected. Byte pp: 03 = iPhone, 05 = Android. |
| 8E | 2 | 10 | 39 pp | Play/pause status. Byte pp = 02 if playing, 00 if paused. |
| 1. 8E | 1. 5 | 1. 10, 01 | 1. 3B 00 00 ss ss | 1. Current track time. Bytes ss ss are a 16-bit representation of the total seconds. |
| 1. 8E | 1. Varies | 1. 10, 01 | 1. 23 30 […] | 1. Name of phone. |
| 1. 8E | 1. Varies | 1. 10, 01 | 1. 23 60 [...] | 1. Music app name in ASCII, if present. |
| 1. 8E | 1. Varies | 1. 10, 01 | 1. 23 61 [...] | 1. Song title in ASCII. |
| 1. 8E | 1. Varies | 1. 10, 01 | 1. 23 62 [...] | 1. Artist name in ASCII. |
| 1. 8E | 1. Varies | 1. 10, 01 | 1. 23 63 [...] | 1. Album name in ASCII. |
| 1. 10 | 1. 3 | 1. 8E | 1. 30 00 00 | 1. Query Pi status. |
| 1. 10 | 1. 3 | 1. 8E | 1. 30 01 00 | 1. Trigger pause/resume. |
| 1. 10 | 1. 3 | 1. 8E | 1. 30 01 10 | 1. Force resume. |
| 1. 10 | 1. 3 | 1. 8E | 1. 30 01 20 | 1. Force pause. |
| 1. 10 | 1. 3 | 1. 8E | 1. 38 0A 00 | 1. Go to the next track. |
| 1. 10 | 1. 3 | 1. 8E | 1. 38 0A 01 | 1. Go to the previous track. |
| 1. 8E | 1. 3 | 1. FF | 1. 77 8E nn | 1. A “Status” message sent every 500ms. If nn is ||01, the video feed switches to the Pi. If the message is not sent or nn is 0, the feed switches back to the Teensy.   If nn is ||10, screen controls are passed to the Pi (8E). If nn is ||20, audio controls are passed as well. |

# Generic Messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source** | **Length** | **Destination** | **Data Example** | **Interpretation** |
| Any | 1 | Any | 80 | Acknowledgment byte. |
| Any | 1 | Any | 01 | Ping (if not followed by any other bytes). |
| Any | 1 | Any | 02 | Pong (if not followed). |