# Command Definitions

## CMD\_GET\_GPIO\_PORT\_COUNT (0xE0)

This command returns the number of GPIO ports present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_GPIO\_PORT\_NAME (0xE1)

This command returns the name of a GPIO port on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Name | | | | | | | |
| … |

**Name:** variable length string (UTF-8)

The index should be less than the number of GPIO ports present on the device (see CMD\_GET\_GPIO\_PORT\_COUNT).

## CMD\_GET\_GPIO\_PORT\_INFO (0xE2)

This command returns the pin information for a specific GPIO port.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 20 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Input Pins | | | | | | | |
| 1 |
| 2 |
| 3 |
| 4 | Output Pins | | | | | | | |
| 5 |
| 6 |
| 7 |
| 8 | Floating Pins | | | | | | | |
| 9 |
| 10 |
| 11 |
| 12 | Loaded Pins | | | | | | | |
| 13 |
| 14 |
| 15 |
| 16 | Overridden Pins | | | | | | | |
| 17 |
| 18 |
| 19 |

**Input Pins:** 32 bits unsigned

**Output Pins:** 32 bits unsigned

**Floating Pins:** 32 bits unsigned

**Loaded Pins:** 32 bits unsigned

**Overridden Pins:** 32 bits unsigned

Input pins should not have their mode changed, and their read value cannot be controlled internally. Output pins can have their mode changed. Floating pins are not connected to anything. Loaded pins are output pins that may not respond properly to pull-up or pull-down modes. Overridden pins are controlled by dedicated hardware functionality that must be disabled with CMD\_DISABLE\_GPIO\_PORT\_OVERRIDES before their modes can be changed.

The index should be less than the number of GPIO ports present on the device (see CMD\_GET\_GPIO\_PORT\_COUNT).

## CMD\_GET\_GPIO\_PORT\_VALUES (0xE3)

This command returns the pin values for a specific GPIO port.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 4 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Pin Values | | | | | | | |
| 1 |
| 2 |
| 3 |

**Pin Values:** 32 bits unsigned

The index should be less than the number of GPIO ports present on the device (see CMD\_GET\_GPIO\_PORT\_COUNT).

## CMD\_SET\_GPIO\_PORT\_MODES (0xE4)

This command sets the mode for a group of pins on a specific GPIO port.

Outgoing Parameters:

**Total length:** 6 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 2 | Mode | | | | | | | |
| 2 | Pins | | | | | | | |
| 3 |
| 4 |
| 5 |

**Index:** 8 bits unsigned

**Mode:** 8 bits unsigned

**Pins:** 32 bits unsigned

Incoming Parameters:

None

The index should be less than the number of GPIO ports present on the device (see CMD\_GET\_GPIO\_PORT\_COUNT).

The pin modes are defined in the following table:

|  |  |  |
| --- | --- | --- |
| Pin Mode # | Definition Name | Description |
| 0 | GPIO\_PIN\_MODE\_HI\_Z | The pin is left floating |
| 1 | GPIO\_PIN\_MODE\_PULL\_DOWN | The pin is pulled down |
| 2 | GPIO\_PIN\_MODE\_PULL\_UP | The pin is pulled up |
| 3 | GPIO\_PIN\_MODE\_LOW | The pin is driven low |
| 4 | GPIO\_PIN\_MODE\_HIGH | The pin is driven high |

## CMD\_DISABLE\_GPIO\_PORT\_OVERRIDES (0xE5)

This command disables all GPIO port overrides, so overridden pins (as reported by CMD\_GET\_GPIO\_PORT\_INFO) can be manipulated. The only way to restore the device to normal operation is a complete reset (e.g. CMD\_RESET).

Outgoing Parameters:

None

Incoming Parameters:

None

## CMD\_GET\_BUS\_COUNTS (0xE6)

This command returns the number of SPI and I2C buses present on the device. These numbers may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | SPI Count | | | | | | | |
| 1 | I2C Count | | | | | | | |

**SPI Count:** 8 bits unsigned

**I2C Count:** 8 bits unsigned

## CMD\_SET\_SPI\_CS\_MODE (0xE7)

This command sets the CS mode on a specific SPI bus.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | SPI Bus Index | | | | | | | |
| 1 | CS Mode | | | | | | | |

**SPI Bus Index:** 8 bits unsigned

**CS Mode:** 8 bits unsigned

Incoming Parameters:

None

The modes are defined in the following table:

|  |  |  |
| --- | --- | --- |
| CS Mode # | Definition Name | Description |
| 0 | SPI\_CS\_MODE\_LOW | CS is held low indefinitely |
| 1 | SPI\_CS\_MODE\_HIGH | CS is held high indefinitely |
| 2 | SPI\_CS\_MODE\_AUTO\_TRANSFER | CS is held low during a transfer |
| 3 | SPI\_CS\_MODE\_AUTO\_BYTE | CS is held low during each byte |

## CMD\_DO\_SPI\_TRANSFER (0xE8)

This command performs a raw low level SPI transfer.

Outgoing Parameters:

**Total length:** variable (≥2 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | SPI Bus Index | | | | | | | |
| … | Data to Transmit | | | | | | | |

**SPI Bus Index:** 8 bits unsigned

**Data to Transmit:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Received Data | | | | | | | |

**Received Data:** variable length byte array

The received data will have the same length as the transmitted data.

## CMD\_DO\_I2C\_WRITE (0xE9)

This command performs a single I2C write.

Outgoing Parameters:

**Total length:** variable (≥3 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | I2C Bus Index | | | | | | | |
| 1 | 7-bit I2C Address | | | | | | | |
| … | Data to Write | | | | | | | |

**I2C Bus Index:** 8 bits unsigned

**7-bit I2C Address:** the I2C address in 7-bit form.

**Data to Write:** variable length byte array

Incoming Parameters:

None

## CMD\_DO\_I2C\_READ (0xEA)

This command performs a single I2C read.

Outgoing Parameters:

**Total length:** 3 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | I2C Bus Index | | | | | | | |
| 1 | 7-bit I2C Address | | | | | | | |
| 2 | Bytes to Read | | | | | | | |

**I2C Bus Index:** 8 bits unsigned

**7-bit I2C Address:** the I2C address in 7-bit form.

**Bytes to Read:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Read Data | | | | | | | |

**Read Data:** variable length byte array

## CMD\_DO\_I2C\_WRITE\_READ (0xEB)

This command performs an I2C write, followed by a repeated start, then an I2C read.

Outgoing Parameters:

**Total length:** variable (≥4 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | I2C Bus Index | | | | | | | |
| 1 | 7-bit I2C Address | | | | | | | |
| 2 | Bytes to Read | | | | | | | |
| … | Data to Write | | | | | | | |

**I2C Bus Index:** 8 bits unsigned

**7-bit I2C Address:** the I2C address in 7-bit form.

**Bytes to Read:** 8 bits unsigned

**Data to Write:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Read Data | | | | | | | |

**Read Data:** variable length byte array

## CMD\_DO\_RADIO\_FIXED\_TEST (0xEC)

This command performs a test with the radio on a fixed radio channel for the specified duration (or indefinitely). NOTE: for devices relying on the radio for the communication with the host, the use of this command will terminate any active connection.

Outgoing Parameters:

**Total length:** 5 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel | | | | | | | |
| 1 |
| 2 | Duration (milliseconds) | | | | | | | |
| 3 |
| 4 | Radio Mode | | | | | | | |

**Channel:** 16 bits unsigned

**Duration:** 16 bits unsigned

**Radio Mode:** 8 bits unsigned

Incoming Parameters:

None

The duration governs how long the test will continue. After the duration has elapsed, the device will return to normal operation. If the duration is set to zero, the radio test will continue indefinitely.

The radio modes are defined in the following table:

|  |  |  |
| --- | --- | --- |
| Radio Mode # | Definition Name | Description |
| 0 | RADIO\_TEST\_MODE\_TX\_CW | Radio transmits an unmodulated carrier at the center frequency |
| 1 | RADIO\_TEST\_MODE\_RX | Radio has receiver enabled |
| 2 | RADIO\_TEST\_MODE\_TX\_MODULATED | Radio transmits a modulated signal |

Note that other modes may be defined by the firmware, to allow testing of specific hardware capabilities.

## CMD\_DO\_RADIO\_SWEEP\_TEST (0xED)

This command performs a sweep with the radio over a contiguous set of channels. NOTE: for devices relying on the radio for the asphodel command communication, use of this command will terminate any active connection.

Outgoing Parameters:

**Total length:** 9 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Start Channel | | | | | | | |
| 1 |
| 2 | Stop Channel | | | | | | | |
| 3 |
| 4 | Hop Interval (milliseconds) | | | | | | | |
| 5 |
| 6 | Hop Count | | | | | | | |
| 7 |
| 8 | Radio Mode | | | | | | | |

**Start Channel:** 16 bits unsigned

**Stop Channel:** 16 bits unsigned

**Hop Interval:** 16 bits unsigned

**Hop Count:** 16 bits unsigned

**Radio Mode:** 8 bits unsigned

Incoming Parameters:

None

The radio will transmit on successive channels from start to stop, inclusive. It will dwell on each channel for the hop interval and then switch to the next in the sequence, wrapping from the stop channel back to the start channel. The device will make a total of hop count channel changes before disabling the test mode and returning to normal operation. If the hop count is zero, the sweep will continue indefinitely.

See the table in the CMD\_DO\_RADIO\_FIXED\_TEST definition for possible values for the radio mode parameter.

## CMD\_GET\_INFO\_REGION\_COUNT (0xF0)

This command returns the number of info regions present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_INFO\_REGION\_NAME (0xF1)

This command returns the name of a specific info region on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Name | | | | | | | |
| … |

**Name:** variable length string (UTF-8)

The index should be less than the number of info regions present on the device (see CMD\_GET\_INFO\_REGION\_COUNT).

## CMD\_GET\_INFO\_REGION (0xF2)

This command returns the data for a specific info region on the device. This data is device specific and only useful for device verification and testing.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Data | | | | | | | |
| … |

**Data:** variable length byte array

The index should be less than the number of info regions present on the device (see CMD\_GET\_INFO\_REGION\_COUNT).

## CMD\_GET\_STACK\_INFO (0xF3)

This command returns the current stack usage of the device (if known). The interpretation of this data is device specific and only useful for firmware development and verification.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 8 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Free Bytes | | | | | | | |
| 1 |
| 2 |
| 3 |
| 4 | Used Bytes | | | | | | | |
| 5 |
| 6 |
| 7 |

**Free Bytes:** 32 bits unsigned

**Used Bytes:** 32 bits unsigned

The returned values will be 0 on devices which do not support stack reporting.

## CMD\_ECHO\_RAW (0xFC)

This command returns the same data as was sent, overwriting the transaction ID and return command ID. This is used for testing, and is not needed for normal device interaction. It can be used to generate various invalid responses from the device.

Outgoing Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Data to Echo | | | | | | | |

**Data to Echo:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Echoed Data | | | | | | | |

**Echoed Data:** variable length byte array

Note: it is permissible for the data to be zero length. In this case, either a zero length packet or no packet will be sent in return (depending on the device implementation). The maximum data length is dictated by the lesser of the outgoing and incoming maximum packet sizes.

## CMD\_ECHO\_TRANSACTION (0xFD)

This command returns the same transaction data (all bytes after the transaction ID) as the transmitted parameters. This is used for testing, and is not needed for normal device interaction. It can be used to generate various invalid responses from the device.

Outgoing Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Data to Echo | | | | | | | |

**Data to Echo:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Echoed Data | | | | | | | |

**Echoed Data:** variable length byte array

Note: it is permissible for the data to be zero length. In this case, the transaction data will also be zero length.

## CMD\_ECHO\_PARAMS (0xFE)

This command returns the same parameter data as was sent. This is used for testing, and is not needed for normal device interaction. It can be used to verify that there are no transmission issues with specific packet lengths.

Outgoing Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Data to Echo | | | | | | | |

**Data to Echo:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| … | Echoed Data | | | | | | | |

**Echoed Data:** variable length byte array

Note: it is permissible for the data to be zero length. In this case, the echoed data will also be zero length. The maximum data length is dictated by the lesser of the outgoing and incoming maximum packet sizes.