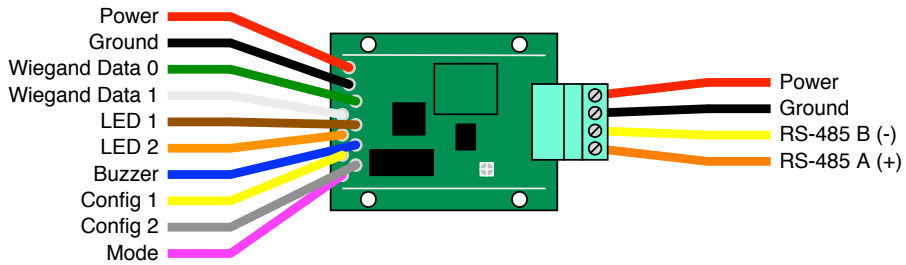
OSM-CPI Technical Documentation

The OSM-CPI is an OSDP to legacy "Control Panel Interface" module, which communicates with a single OSDP peripheral device.

Signals and Pinout

The OSM-CPI has a single 4 position terminal block, and a 10 wire flying lead interface. The signals of the terminal blocks and flying leads are named according to the diagram below



Power

Power can be supplied to the OSM-CPI by either the red and black flying leads, or pins 1 and 2 of the terminal block. These connections are common, and are intended to be used to pass power through the OSM-CPI from the flying leads to a peripheral device connected to the terminal blocks. **Do not connect two sepatate power supplies to these connections.**

The OSM-CPI can operate using between 6 and 36 volts DC input, and draws at most 200 mA at 12 V.

RS-485

The OSM-CPI uses the DS3695 RS-485 transceiver to interact with the RS-485 bus and to communicate with other OSDP compliant devices. The OSM-CPI supports communication speeds of 9600, 19200, 38400, 57600, and 115200 bits per second, and defaults to 9600 bps. The communication speed can be changed by issuing an `osdp_COMSET` command to the OSM-CPI in `PD` mode.

The RS-485 B and A signals are connected to the terminal block pins 3 and 4, respectively.

Wiegand

The OSM-CPI's Wiegand output D0 and D1 signals are connected to the green and white flying leads, respectively. The Wiegand outputs are open-collector NPN transistors, pulled up to 5 volts internally through diodes and 680Ω resistors.

Wiegand transmissions are nominally 1000 bits per second, with a pulse-off time of about 100 µs.

The OSM-1000 can handle between 1 and 255 bits of Wiegand, and handles special cases for 8 bit burst keypad readers.

Other Digital I/O

The OSM-CPI has two dedicated digital inputs, two signals that can be either inputs or outputs, depending on configuration, and two dedicated analog inputs.

In the default configuation, the brown and orange flying leads are inputs that control the PD's LED, the blue flying lead is an input that controls the PD's buzzer, the yellow flying lead is an output that indicates tamper, and the purple and grey flying leads are analog inputs used for configuration.

Configuration

The OSM-CPI and compatible attached peripheral devices can be configured in various ways by shorting certain combinations of the flying leads together at power-up.

Pairing

An OSM-CPI connected peripheral device can be "paired" together, by shorting the purple flying lead to ground before power up. The OSM-CPI will initiate a secure channel session using `SCBK_D`, then issue an `osdp_KEYSET` command with a new random key. From then on, during normal operation, the OSM-CPI will initiate secure channel sessions using only that key.

Factory Default

An OSM-CPI connected to a compatible peripheral device can both be restored to a factory default state, by shorting the purple flying lead to ground, and shorting the green flying lead to the yellow flying lead before power up. The OSM-CPI will send manufacturer specific commands to the peripheral device that will cause it to be restored to a factory default state, then restore itself to a factory default state.

PD Mode

An OSM-CPI can be configured to emulate a peripheral device for further configuration, by shorting both the purple and grey flying leads to ground before power up.

Communication

When power is first applied, the OSM-CPI will indicate that it is powering up by cycling it's diagnostic LED through off, red, blue, green, and off (about about a half-second per color). The diagnostic LED will then be solid white for about 2.5 seconds, then solid blue again for about a half-second before polling begins.

After powering up, the OSM-CPI will first send an `odsp_POLL` command. It will continue to send `osdp_POLL` commands at about 1.2 Hz until it receives an `osdp_ACK` reply. The OSM-CPI will send an `osdp_ID` command, and will continue to do so at about 1.2 Hz until it receives an `osdp_PDID` reply. After the first correct reply has been received, the OSM-CPI will send an `osdp_CAP` command, expecting an `osdp_PDCAP` reply. If the reply to this command is not as expected, or never arrives, the OSM-CPI will restart the connection sequence from the beginning.

At this point, to initiate a secure channel session with the attached PD the OSM-CPI sends the `osdp_CHLNG` command, receiving and validating the `osdp_CCRYPT` reply from the `PD`, sending the `osdp_SCRYPT` command, and finally receiving and validating the `osdp_RMAC_I` reply. If the reply to any of these commands is not as expected, or never arrives, the OSM-CPI will restart the connection sequence from the beginning.

After communication initalization, the OSM-CPI will begin sending the `osdp_POLL` command at about 8 Hz until either the `PD`'s reply is not `osdp_ACK`, or one of the OSM-CPI's digital inputs change state.

If either of the OSM-CPI's LED inputs change state, it will send an `osdp_LED` command with one of four payloads depending on the states of the two LED inputs after the change. All configurations send a permanent LED command to reader 0, LED 0, with the color of the LED set according to the following table:

LED1	LED2	payload	Color
high	high	`0x00000000000000000000101010000`	Off
low	high	`0x00000000000000000000101010101`	Red
high	low	`0x00000000000000000000101010202`	Green
low	low	`0x00000000000000000000101010303`	Amber

If the OSM-CPI's buzzer input changes from high to low, it will send an `osdp_BUZ` command with the following payload: `0x0002010000`, setting the state of the `PD` 's reader 0's buzzer to a continous default tone.

If the OSM-CPI's input 1 line changes from low to high, it will send an `osdp_BUZ` command with the following payload: `0x0001000000`, setting the state of the `PD` 's reader 0's buzzer to off.

If the OSM-CPI receives an `osdp_RAW` reply, it will extract and output the reader data from that message using the Wiegand interface.

If the OSM-CPI receives an `osdp_KEYPAD` reply, it will extract and output the keypad data in 8 bit burst mode, detailed in the following table:

Key	Wiegand Sequence
0	`11110000`
1	`11100001`
2	`11010010`
3	`11000011`
4	`10110100`
5	`10100101`
6	`10010110`
7	`10000111`
8	`01111000`
9	`01101001`
clear/delete/*	`01011010`
enter/#	`00101101`