

Connected Optical Security Synchronization (COSYS)

Protocol

By the **C**onected **O**ptical **L**ogistics and **I**nfrared **N**etworks Company.
January 22, 2025.

OVERVIEW

The Connected Optical Security Synchronization (COSYS) protocol is used by the Connected Optical Logistics and Infrared Networks (C.O.L.I.N.) Company in many of its products. This protocol communicates status that can be received by robo-guards and other mobile devices.

The protocol uses a 59.0 kHz IR modulation that can be received by off-the-shelf receivers, available with varying sensitivity. Optionally a user can simply use the IR light to detect the presence or location of one of our products (without demodulating the light or decoding the data).

Status information is communicated using a standard 600 baud rate signal. This document details the physical and protocol layers.

PHYSICAL LAYER

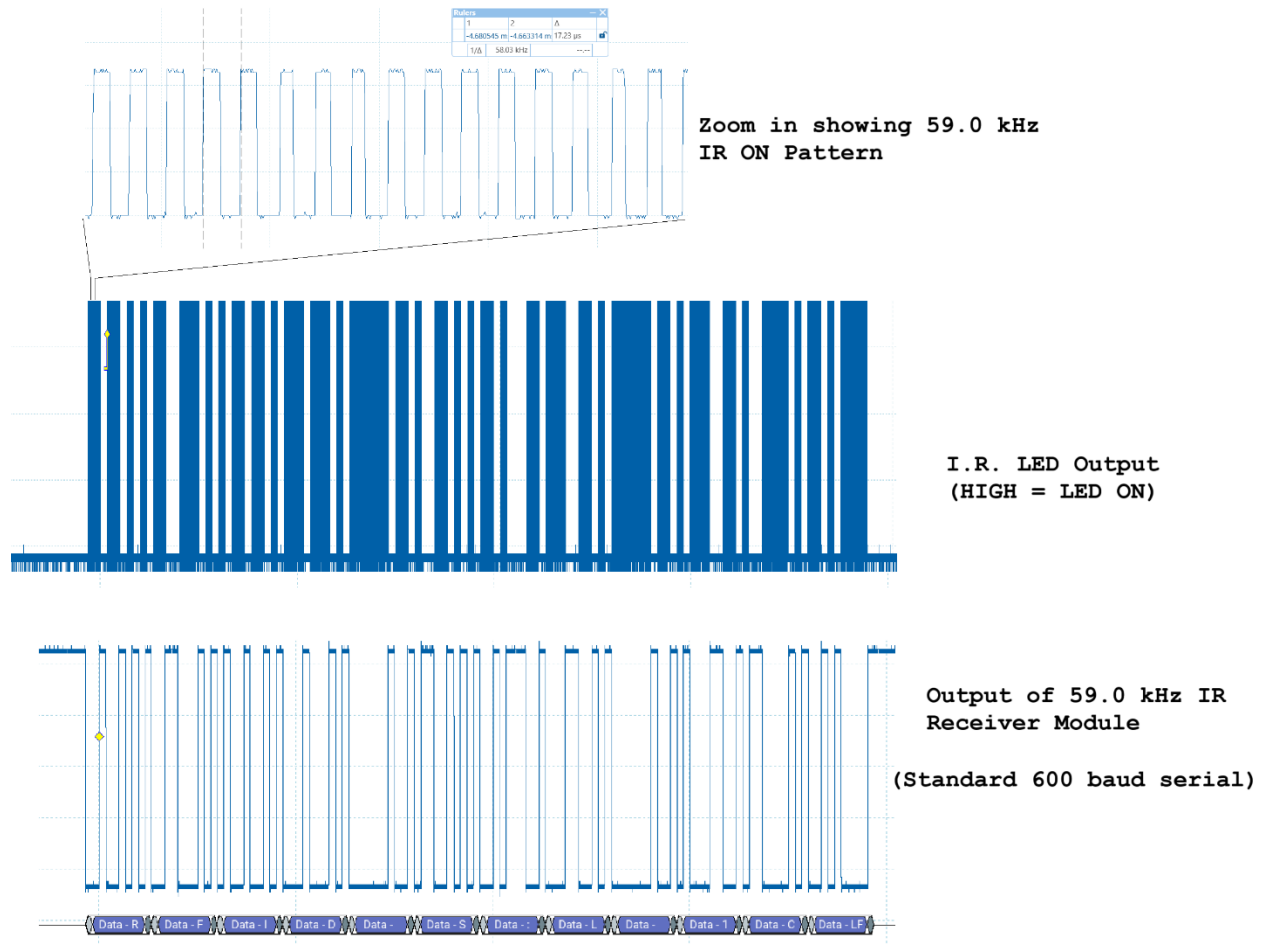
The physical layer uses a 59.0 kHz infrared (940nm) light. This is transmitted from an LED that is located 10mm from the floor. The protocol modulates serial data at 600 baud.

Data is sent approximately every 100 - 900 mS. Devices may stop sending data when performing other processing.

The use of 940nm LED allows usage for both intensity (analog) measurements as well as digital demodulation of the data.

A "0" is encoded as a 59.0 kHz IR pattern being transmitted, a "1" is encoded as the absence of the pattern.

The following show the over-the air waveform as well as the final decoded signal using a decoder module:



The standard 600 baud signal could be decoded by connecting to a microcontroller serial port line (such as Arduino RX pin).

Above the physical layer, a simple protocol layer is used to ensure reliable data transfer. This protocol is described in the next section.

PROTOCOL LAYER

All messages are transmitted as ASCII characters. Messages have the following format with length shown above:

Variable Bytes	1 Byte	2 Bytes	1 Byte
DEVICE MESSAGE	Space (0x20)	Checksum HEX	Newline (0x0A)

The device message is given in the datasheet for each device. In general the format of the device specific message is as follows:

IND STATUS

The first several characters identify the device, the rest of the message includes status or other information.

The format of an entire message (including checksum) is as follows:

IND 011111 OK 46\n

Note that 46 is in ASCII-encoded HEX format. That is the value of 46 is 0x46. See checksum calculation below.

All messages must end with a newline.

CHECKSUM CALCULATION

The checksum is calculated over the device message, NOT including the space separating the device message and checksum.

The checksum is the XOR of all message bytes, with an initial value of 0x00.

The following example Python code would take a message (such as the message from the RFID unlock system), calculate the checksum, and print the final message including the checksum and new line:

```
-----  
1:  s = "RFID S:U"  
2:  chsum = 0;  
3:  for g in s:  
4:      chsum ^= ord(g)  
5:  
6:  print(s + " %02x\n"%chsum)  
-----
```

The following is a similar example in C:

```
-----
```

```
1:  uint8_t chksum(char * data, unsigned char len){
2:      uint8_t sum = 0;
3:
4:      while(len){
5:          sum ^= *data++;
6:          len--;
7:      }
8:
9:      return sum;
10: }
```

Revision History

January 29, 2025: Addition of checksum details

January 22, 2025: Initial Release