by Opal Kelly

SYZYGY DNA Specification

Version 1.1

Software, documentation, samples, and related materials are

Opal Kelly Incorporated
Portland, Oregon
http://www.opalkelly.com

Revision History:

Date	Description
2018-04-19	Initial release.
2019-07-24	Added “IS_TXR4” attribute flag to differentiate TXR-2 and TXR-4 transceiver peripherals. Clarify guidance to carrier manufacturers in Smart VIO handling. Fix typos.

Contents

1. Introduction

SYZYGY DNA is a partner specification to the SYZYGY specification that describes how SYZYGY peripherals can communicate information to the carrier. This information generally includes the following:

- Allowable I/O voltage ranges
- Indentifying information such as manufacturer name, part number, serial number, etc.

This specification details both the I²C-compatible protocol used to communicate with the carrier as well as the memory layout of the associated DNA information.

1.1. Defined Terminology

- pMCU - The Peripheral MCU is the physical device attached to the I²C signals on the peripheral.
- DNA - The EEPROM contents that describes the peripheral personality.

2. Protocol

SYZYGY DNA is communicated using an I²C-compatible protocol between the SmartVIO controller on the carrier and the pMCU on the peripheral.

2.1. Geographical Address

Each port's geographical address is fixed at the carrier using a 1% resistor between the port's RGA pin and DGND. The peripheral has a 10 kΩ resistor between RGA and +3.3VDD forming a resistor divider with a resulting voltage at RGA. Each peripheral must determine its geographical address by reading the voltage at the RGA pin of the SYZYGY port.

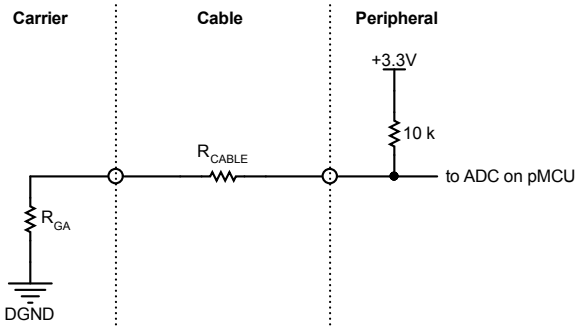

Figure 1. Schematic diagram of geographical addressing resistor network.

The pMCU should measure the voltage on the RGA pin shortly after startup and determine its I²C address based on the table below. The resistor values have been set to allow some degree of measurement error and to accommodate the small contribution from RCABLE. Standard 1% resistor values were chosen to provide approximately 200 mV between voltages. It is recommended that the pMCU accept any reading within 75 mV as a valid reading for the geographical address. This will leave about 50 mV of "dead band" ambiguity where the pMCU cannot make a geographical address determination.

Table 1: Carrier-side resistor values and their corresponding geographical addresses.

R_{GA} Resistor (kΩ)	Nominal R_{GA} Voltage	I²C Address (7-bit)	I²C Address (8-bit)
210	3.147	0x30	0b0110_000X
84.5	2.944	0x31	0b0110_001X
49.9	2.740	0x32	0b0110_010X
34.0	2.548	0x33	0b0110_011X
24.9	2.341	0x34	0b0110_100X
18.7	2.135	0x35	0b0110_101X
14.3	1.926	0x36	0b0110_110X
11.3	1.734	0x37	0b0110_111X
8.87	1.535	0x38	0b0111_000X
6.98	1.341	0x39	0b0111_001X
5.36	1.137	0x3A	0b0111_010X
4.02	0.933	0x3B	0b0111_011X
2.94	0.738	0x3C	0b0111_100X
2.00	0.541	0x3D	0b0111_101X
1.18	0.342	0x3E	0b0111_110X
0.487	0.153	0x3F	0b0111_111X

2.2. Read Operation

Read operations begin with a write transaction used to set the sub-address to read from. This is accomplished by using a write transaction (R/$\overline{W}$ bit low) to send the sub-address in two bytes, high byte first then low byte.

The read operation continues with another I²C start condition followed by the pMCU geographical address with the R/$\overline{W}$ bit set high. The pMCU must acknowledge the read transaction as the previous transaction did not result in any data being written to the microcontroller. Data bytes are read one byte at a time. Any number from 1 to 32 consecutive bytes may be read in a single transaction. The SmartVIO controller will acknowledge each received byte. After reading the desired number of bytes the SmartVIO controller ends the transaction with an I²C stop condition.

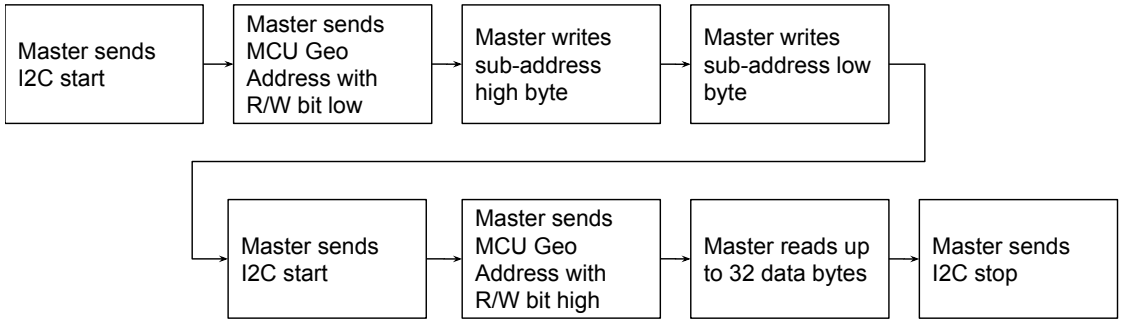

Figure 2. DNA Read Protocol

2.3. Write Operation

NOTE: Write operation to the pMCU is not specifically required by this specification but may be helpful for initial setup. This information is provided for guidance purposes only and may help peripheral manufacturers configure their peripherals.

Write operations begin with an I²C start condition followed by the pMCU geographical address with the R/$\overline{W}$ bit set low. The pMCU will acknowledge the write setup if it is ready to receive data.

With an acknowledged write operation, the SmartVIO controller then writes the high and low bytes of the desired sub-address to write to. The sub-address is followed by 1 to 32 sequential data bytes written one byte at a time. The MCU will acknowledge each received byte if the address location written to is a valid location, note that some register locations may be read-only. After sending the desired data the SmartVIO controller ends the operation with an I²C stop condition.

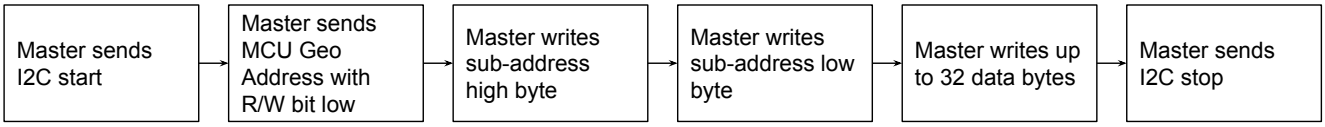

Figure 3. DNA Write Protocol

2.3.1. Completion Status

Due to flash write times on microcontrollers, the pMCU may require some time to complete a write operation. This can cause the pMCU to skip a subsequent write setup acknowledgment. The SmartVIO controller should poll the status of the pMCU by repeatedly writing the pMCU address with the R/$\overline{W}$ bit set low until the pMCU responds with acknowledgment.

3. Memory Organization

SYZYGY pMCU firmware memory is split into three sections, the firmware register section, the DNA data section, and a reserved area set aside for future use. The firmware register section contains firmware-specific registers that allow for direct communication between the SmartVIO controller and the pod microcontroller. The DNA memory contains information on the pod including acceptable voltage ranges for the SYZYGY SmartVIO.

Table 2: DNA Memory Map

Memory Locations	Contents
0x0000 - 0x7FFF	pMCU firmware registers
0x8000 - 0x8FFF	DNA EEPROM contents
0x9000 - 0xFFFF	Reserved

3.1. Firmware Register Map

The Firmware Register portion of the SYZYGY DNA specification occupies the first 32,768 addressable locations. The first 256 register locations are reserved for use as firmware registers defined by the SYZYGY DNA specification. The rest of the register space is considered firmware-specific and may be used by a firmware to perform special functions. Note that not all carriers will have full support for custom registers so these should be used sparingly.

Table 3: pMCU Firmware Register Map

Registers Contents	R/W	Location
FW major version	R	0x0000
FW minor version	R	0x0001
SYZYGY DNA major version	R	0x0002
SYZYGY DNA minor version	R	0x0003
EEPROM size (high byte)	R	0x0004
EEPROM size (low byte)	R	0x0005
Reserved	R/W	0x0006 - 0x02FF
Firmware specific (user defined)	R/W	0x0300 - 0x7FFF

3.2. DNA EEPROM

The SYZYGY DNA firmware stores information unique to a specific pod, including information on the pod itself, its manufacturer, versioning information, and the pod's power supply requirements. This data is referred to as the peripheral DNA (or simply DNA) data and begins at address 0x8000 of the pMCU sub-address space. Individual data within the DNA is stored either as individual bytes (uint8), 2-byte unsigned integers (uint16), or strings (ASCII without null termination). Multi-byte data is stored in a little endian format (least significant byte comes first in memory).

3.2.1. Front Matter

The DNA begins with the length of the DNA data section (in bytes) to provide the SmartVIO controller with enough information to read the full DNA section to memory.

The data length is followed by two bytes for the major and minor versions (respectively) of the DNA specification used to construct the data. For example, version 3.6 of the specification will contain '3' in the major version byte and '6' in the minor version byte.

A peripheral must specify the minimum required version using the same format as the version used to construct the data. The specification is considered forward compatible, so a minimum required version of 0.0 indicates that the DNA is compatible with any version of the specification.

The maximum load that the pod will place on the 5V and 3.3V power supplies is provided in two separate uint16 values. These current values are specified in singe milliamperes. The maximum load values can be used by a SmartVIO controller to determine if the carrier has enough capacity on each rail to power the pod.

Attribute flags is a 16-bit collection of binary attributes for the peripheral. This version supports two such flags:

- `IS_LVDS` - Set to `1` to indicate that the peripheral requires LVDS on the port. For some carriers, this may provide an additional SmartVIO constraint if the host FPGA requires it.
- `IS_DOUBLEWIDE` - Set to `1` to indicate that the peripheral is a double-wide peripheral. This will indicate to the carrier that the SmartVIO constraints apply to this port as well as a partner port.
- `IS_TXR4` - Set to `1` to indicate that the peripheral is TXR4 peripheral. If this is set, the required SYZYGY DNA major and minor revisions must be at least 1.1 to assure compatibility with supporting carriers.

3.2.2. Smart VIO

VIO compatibility is specified in up to four “VIO Ranges”. Each VIO range specifies an acceptable voltage range that all components on the pods VIO rail can operate under. For example, if a pod can operate from 1V to 2V or from 3V to 5V, two VIO ranges can be provided. The VIO voltage must be specified in 10’s of millivolts and should represent the nominal operating range of the device. If a particular range is unused, it should be specified with minimum and maximum set to 0. The VIO load is specified in single milliamperes.

Power shall be applied to the VIO group only when an acceptable Smart VIO solution has been found. Power must not be applied if any of the following conditions are true:

- The “required SYZYGY DNA major version” and “required SYZYGY DNA minor version” together do not satisfy the carrier’s SYZYGY DNA version support.
- The peripheral is incompatible with the port it is plugged into (for example, if a TXR2 peripheral is connected to a TXR4 port).
- The solver cannot find a suitable VIO voltages that satisfies at least one range for each of the peripherals.

To simplify carrier operation and behavior, the following rules should be applied when programming the SmartVIO ranges. If these rules are not followed, unspecified behavior may result from the carrier.

- Within a range, $V_{min} \leq V_{max}$
- VIO range 1 should be lower than VIO range 2 and so on. Furthermore, ranges should not overlap. That is, $V_{max,n} < V_{min,n+1}$

3.2.3. CRC Error Detection

The DNA data header is protected by a 16-bit cyclic redundancy check (CRC). This CRC protects the data header information but does not protect the string contents. The CRC calculation is defined by the following polynomial: $x^{16} + x^{12} + x^5 + 1$

The CRC is computed after initializing the CRC register with 0xFFFF and data is shifted in most-significant bit first. Source code (C) to compute the CRC is shown in Figure 4.

Note that the endianness of the CRC bytes in the header are not the same as the endianness of other 16-bit values. This is due to the arrangement of the CRC computation. When the correct CRC value is placed in these bytes, the CRC of the full data header can be computed and should be 0x0000 if the header is intact.

```
/// Computes the CRC-16/CCITT checksum using parallel computation without tables.
/// https://en.wikipedia.org/wiki/Computation_of_cyclic_redundancy_checks
///
/// Polynomial: 0x1021 (x^16 + x^12 + x^5 + x^0)
/// Initialization: 0xFFFF
/// Data "shifted" MSB first
///
/// \returns Computed 16-bit CRC.
unsigned short
szgComputeCRC(const unsigned char *data, unsigned int length)
{
        unsigned short x, crc = 0xffff;

        while (length--) {
                x = (crc>>8) ^ *data++;
                x ^= x>>4;
                crc = (crc<<8) ^ (x<<12) ^ (x<<5) ^ (x);
                crc &= 0xffff;
        }
        return(crc);
}
```

Figure 4. C Source Code for CRC-16 Computation

3.2.4. String Payloads

The final segment of the DNA contains a number of strings with information on the pod, its manufacturer, and model/revision information. The offset for each of these strings is computed by summing the appropriate lengths from the DNA header. The strings are composed of ASCII characters and do not have a null termination.

3.2.5. DNA Memory Layout

Table 4: DNA Memory Layout

Contents	Type	Offset	Note
DNA full data length (bytes)	uint16	0	
DNA header length (bytes)	uint16	2	
SYZYGY DNA major version	uint8	4	
SYZYGY DNA minor version	uint8	5	
Required SYZYGY DNA major version	uint8	6	
Required SYZYGY DNA minor version	uint8	7	
Maximum operating 5V load (mA)	uint16	8	
Maximum operating 3.3V load (mA)	uint16	10	
Maximum VIO load (mA)	uint16	12	
Attribute flags	uint16	14	[0]=IS_LVDS [1]=IS_DOUBLEWIDE [2]=IS_TXR4
Minimum operating VIO (10 mV steps)	uint16	16	VIO Range 1
Maximum operating VIO (10 mV steps)	uint16	18	
Minimum operating VIO (10 mV steps)	uint16	20	VIO Range 2
Maximum operating VIO (10 mV steps)	uint16	22	
Minimum operating VIO (10 mV steps)	uint16	24	VIO Range 3
Maximum operating VIO (10 mV steps)	uint16	26	
Minimum operating VIO (10 mV steps)	uint16	28	VIO Range 4
Maximum operating VIO (10 mV steps)	uint16	30	
Manufacturer name length	uint8	32	
Product name length	uint8	33	
Product model / Part number length	uint8	34	
Product version / revision length	uint8	35	
Serial number length	uint8	36	
RESERVED	uint8	37	
CRC-16 (most significant byte)	uint8	38	
CRC-16 (least significant byte)	uint8	39	
END DATA HEADER			
Manufacturer name	string	40	
Product name	string		Offset requires computation.
Product model / Part number	string		Offset requires computation.
Product version / revision	string		Offset requires computation.
Serial number	string		Offset requires computation.

3.3. Example SYZYGY DNA Contents

The table below shows the contents of the POD-CAMERA DNA EEPROM. This content is transferred to the carrier during the carrier's boot sequence and helps it determine which ports have connected peripherals. The carrier can then use this information to determine a SmartVIO solution and enable the VIO regulators.

Table 5: Example DNA contents from the POD-CAMERA

Contents	Type	Value
DNA full data length (bytes)	uint16	101
DNA header length (bytes)	uint16	40
SYZYGY DNA major version	uint8	1
SYZYGY DNA minor version	uint8	0
Required SYZYGY DNA major version	uint8	0
Required SYZYGY DNA minor version	uint8	0
Maximum operating 5V load (mA)	uint16	0
Maximum operating 3.3V load (mA)	uint16	510
Maximum VIO load (mA)	uint16	50
Attribute flags	uint16	0x0001
Minimum operating VIO (10 mV steps)	uint16	180
Maximum operating VIO (10 mV steps)	uint16	330
Minimum operating VIO (10 mV steps)	uint16	0
Maximum operating VIO (10 mV steps)	uint16	0
Minimum operating VIO (10 mV steps)	uint16	0
Maximum operating VIO (10 mV steps)	uint16	0
Minimum operating VIO (10 mV steps)	uint16	0
Maximum operating VIO (10 mV steps)	uint16	0
Manufacturer name length	uint8	23
Product name length	uint8	10
Product model / Part number length	uint8	17
Product version / revision length	uint8	1
Serial number length	uint8	10
RESERVED	uint8	0
CRC-16 (most significant byte)	uint8	0x72
CRC-16 (least significant byte)	uint8	0xF9
END DATA HEADER		
Manufacturer name	string	`Opal Kelly Incorporated`
Product name	string	`POD-CAMERA`
Product model / Part number	string	`POD-CAMERA-AR0330`
Product version / revision	string	`A`
Serial number	string	`1743000ABC`

4. LEGAL NOTICE

This SYZYGY DNA Specification document (this “Document”) is proprietary to Opal Kelly Incorporated (“Opal Kelly”) and/or its successors and assigns.

4.1. CONDITIONS OF USE

Each party that receives a copy of this Document (“Recipient”) via any means and/or in any format acknowledges and agrees that each such Recipient is only permitted to implement and/or use this Document pursuant to a separate license agreement that the Recipient has entered into with Opal Kelly (the “Specification License”). The Document may not be used without first entering into that Specification License. Unauthorized use of this Document is strictly prohibited.

Recipient may make a reference to, or have a citation/footnote that cites, this Document provided the Recipient fully acknowledges Opal Kelly’s copyright ownership of this Document and gives Opal Kelly full copyright attribution. The proper copyright notice for this Document is as follows: “© 2017 OPAL KELLY INCORPORATED. ALL RIGHTS RESERVED.”

Without limiting the foregoing, Recipients of this Document are not permitted to do the following (without first obtaining Opal Kelly’s prior written consent): (i) revise, alter, modify, make derivatives of, or otherwise amend this Document in any way; or (ii) reproduce or copy the Document or transmit the Document to any third party.

SYZYGY™ and Opal Kelly® (among others) are trademarks, registered trademarks, service marks, and/or trade names (collectively “Marks”) owned by Opal Kelly and all rights are reserved therein. Absent an express grant set forth in a valid and signed Specification License between Opal Kelly and a particular Recipient, nothing in this Document grants (either expressly or impliedly) to any Recipient any right or license to use this Document, any information or other contents in this Document, or any other intellectual property owned or controlled by Opal Kelly (including, without limitation, any of the Marks).

4.2. PROVIDED “AS IS” ; DISCLAIMER

THIS DOCUMENT AND ALL INFORMATION CONTAINED HEREIN IS PROVIDED TO ALL RECIPIENTS ON AN “AS IS” BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, OPAL KELLY (ALONG WITH ALL CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NON-INFRINGEMENT.