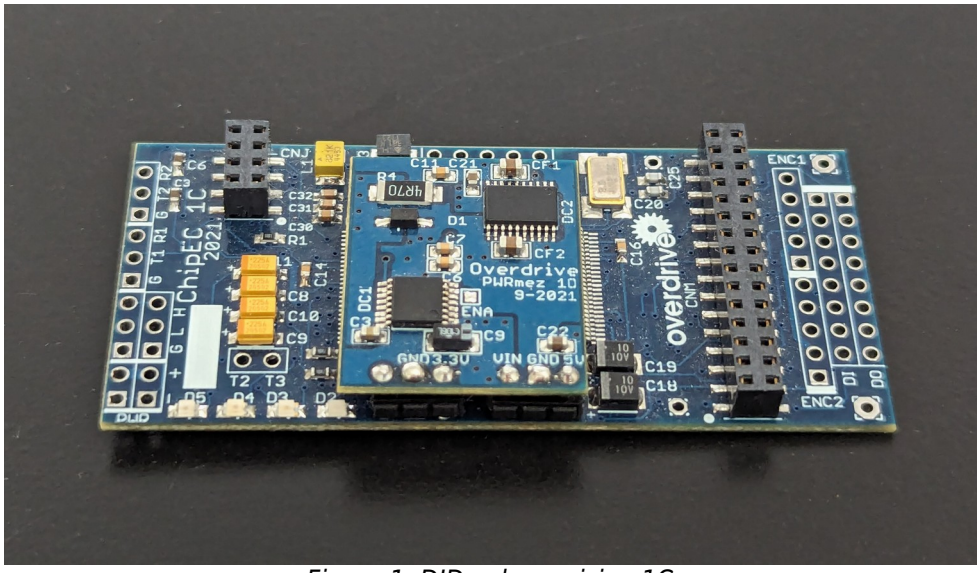



*Figure 1. DIDcoder revision 1C*

## Introduction

The DIDcoder is a dual-channel circuit board which counts incremental Encoder pulses and sends those values to a CAN bus master via CANopen protocol. The DIDcoder can also communicate with the Overdrive Control Computer (CC) or the Embedded Computer (EC) via CAN bus. The DIDcoder communicates on the CAN bus with a minimal implementation of the CANopen CiA-406 protocol. It supports 32-bit 4x quadrature decoding, pulse counting, index pulse, and value preset functions. LEDs indicate node ID and communications status. CAN bus termination is also available on the board.

DIDcoders have the capability of providing two single-ended quadrature encoder inputs and sixteen 5V TTL/CMOS digital inputs.

The DIDcoder is based on the Freescale/NXP 56F8355 DSC chip, further specifications can be found in that device's datasheet.

**Dimensions:** (per image above)

Length (horizontal): 2.7in (68.58mm)
Width (vertical): 1.26in (32mm)
Height: 0.40in (10.16mm) (power mezzanine may be removed for reduced height)

*~ The DIDcoder is static sensitive and may be damaged if its pins are touched without taking static dissipation safety precautions. ~*

## Electrical Specifications

Power for the DIDcoder can be in the range 4.3V - 16V

Quadrature signals are TTL/CMOS, and are tolerant of 5.5V levels maximum.
Digital IO signals are TTL/CMOS, and are tolerant of 5.5V levels maximum.

The device consumes a maximum of 820 mW, depending on power source.
Input voltage vs power consumption:

12V : 68 mA
9V : 83 mA
7.2V : 97 mA
4.8V : 135 mA

This is the power into the DIDcoder's on-board DC-DC converter. (The converter becomes less efficient as the input voltage gets higher. )

## CAN Bus

The DIDcoder is typically used as a CANopen input device on a CAN bus and it supports a minimal subset of CANopen CiA-406 protocol. Multiple communications bit-rates are supported. Because the DIDcoder CAN transceiver is non-isolated, thus it does not have a "CAN GND" pin.

CAN communications bit rates supported: 50k, 125k, 250k, 500k, 800k, 1Mbit.

## ID, Bit-rate, and Flashing

The DIDcoder's ID and baud rate are pre-configured during manufacturing but can also be configured using CANopen Layer Setting Services ("LSS").

This example shows CAN messages exchanged between a CANopen master and a DIDcoder.
The CAN-ID on the DIDcoder is changed from '1' to '5'.

LSS Set Node ID to ID 5:
0x7E5 [8] 11 05 00 00 00 00 00 00

response from DIDcoder:
0x7E4 [8] 11 00 00 00 00 00 00 00

The change of ID is immediate and the 'LSS Store configuration' message is not needed.

The device communications bit-rate can also be changed via LSS, as follows:

LSS switch global mode:
0x07e5 8 0x04 0x01 0x00 0x00 0x00 0x00 0x00 0x00

LSS bit rate set to 500k bps:
0x07e5 8 0x13 0x00 0x02 0x00 0x00 0x00 0x00 0x00

LSS store configuration:
0x07e5 8 0x17 0x00 0x00 0x00 0x00 0x00 0x00 0x00

## LED indicators

The Green LED flashes when CAN traffic is present on the bus, signifying that the DIDcoder is communicating with the CAN Master.

The Red LED blinks to indicate the device Node ID which was flashed into the DIDcoder. One blink means an ID of one, two blinks means an ID of two, etc. A pause of 2 seconds occurs after the unit blinks its ID.

A Yellow LED is either an error indication, or, the completion of a flashing operation. This LED should not be illuminated under normal operating conditions – contact tech support if it illuminates.

# Quadrature

The DIDcoder counts incremental quadrature encoder pulses and sends values to a CAN master.
The DIDcoder has two quadrature ports for incremental encoder input. These are single-ended TTL/CMOS signals which support input voltage levels up to 15 volts.

By default, the DIDcoder sends two CANopen TPDOs, 0x180+ID and 0x280+ID, as in the example below where node ID = 1.

0x181 [2] FF FF
0x281 [8] AA AA AA AA BB BB BB BB

In the example above, 0x180+ID contains the 16 Digital Inputs. Sending of this TPDO can be disabled, see configuration notes below.

In the example above, 0x280+ID contains the encoder count value. The first 4 bytes (bytes 0 - 3) are the value of encoder 1 (highlighted blue above), bytes 4-7 the value of encoder 2 (highlighted green above). Sending of the second encoder value can be disabled in device settings.

The Index pulse can also be enabled via device settings, and its polarity can be changed. When an index pulse is triggered, the corresponding encoder value will be reset to the preset value.

### Configuration of Object Dictionary entries via CANopen SDO

| | |
|---|---|
| Preset value for encoder 1 | 0x601 [8] 2E 10 60 01 XX XX XX XX |
| Index | 0x6010 |
| Sub-Index | 1 |
| Data Type | Integer32 (bytes 4-7) |

*(Above: node ID=1, SDO is 8 bytes, Preset value is “XX XX XX XX”.)*

| | |
|---|---|
| Preset value for encoder 2 | 0x601 [8] 2E 10 60 02 XX XX XX XX |
| Index | 0x6010 |
| Sub-Index | 2 |
| Data Type | Integer32 (bytes 4-7) |

| | |
|---|---|
| Repeat NMT boot message | 0x601 [5] 2F 11 60 00 XX |
| Index | 0x6011 |
| Sub-Index | 0 |

| | |
|---|---|
| Data Type | Integer8 (byte 4)<br>0x00 = single boot message<br>0x01 = repeat boot messages |

| | |
|---|---|
| Digital Input TPDO | 0x601 [5] 2F 11 60 01 XX |
| Index | 0x6011 |
| Sub-Index | 1 |
| Data Type | Integer8 (byte 4)<br>0x00 = do not send TPDO<br>0x01 = send TPDO |

| | |
|---|---|
| Encoder TPDO | 0x601 [5] 2F 11 60 02 XX |
| Index | 0x6011 |
| Sub-Index | 2 |
| Data Type | Integer8 (byte 4)<br>0x00 = send only encoder value 1<br>0x01 = send both encoder values 1 and 2 |

| | |
|---|---|
| Index pulse | 0x601 [5] 2F 11 60 03 XX |
| Index | 0x6011 |
| Sub-Index | 3 |
| Data Type | Integer8 (byte 4)<br>0x00 = ignore Index pulse<br>0x01 = use index pulse to reset encoder value |

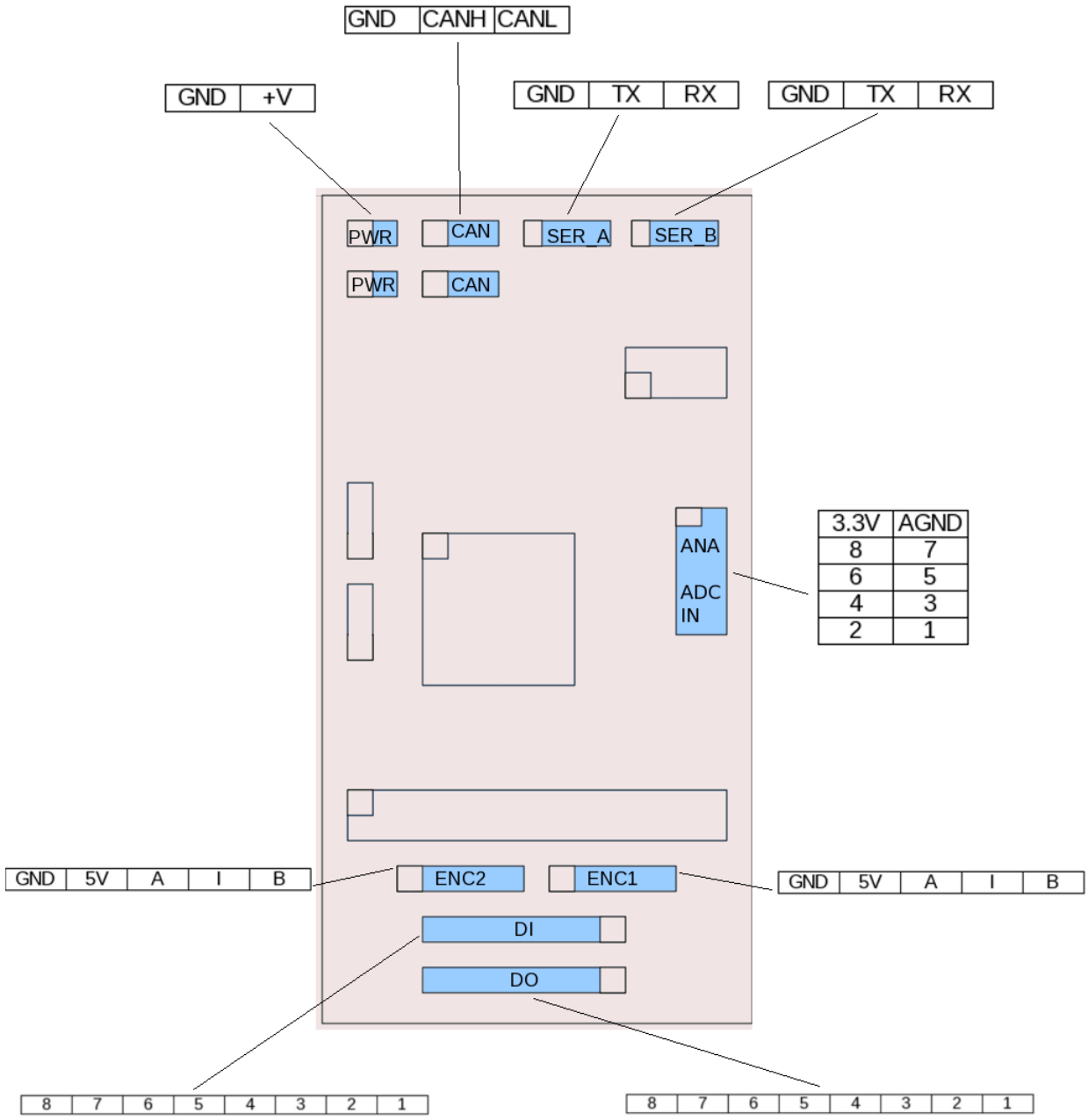
| | |
|---|---|
| Index pulse polarity | 0x601 [5] 2F 11 60 04 XX |
| Index | 0x6011 |
| Sub-Index | 4 |
| Data Type | Integer8 (byte 4)<br>0x00 = trigger on rising edge<br>0x01 = trigger on falling edge |

## Digital Input

The DIDcoder has two sets of eight Digital Input pads labeled “DIN 1-8” and “DO 1-8”, for a total of 16 digital inputs, which are 3.3V CMOS logic but are tolerant of 5V TTL/CMOS signals.

Digital inputs are pulled high by internal high-impedance pull-up resistors, so if left unconnected each bit will read a value of logical “1”. Connecting a DIN pad to GND asserts a logical “0”. When configured to do so, the DIDcoder will continuously send the state of the digital inputs over CAN upon reception of a SYNC from the master.

### *Physical Layout*




→ Pinout of encoder pads is seen in the above drawing on ports “ENC1”, “ENC2”.
→ Digital inputs are on pads “DI” (inputs 1-8) and “DO” (inputs (9-16).
→ CAN bus is available on two pass-through ports labeled “CAN” at the top of the board.
→ A solder-pad jumper which, when bridged, enables the onboard 120 ohm bus termination is available on the bottom of the board, labeled “TERM” *(not pictured above).*
→ Power is applied to the upper left “PWR” port, pads GND, +V.

*Contact technical support at for further interfacing information*
*conceptoverdrive.com*
*support@conceptoverdrive.com*
*Burbank, CA USA*
*818-980-6691*