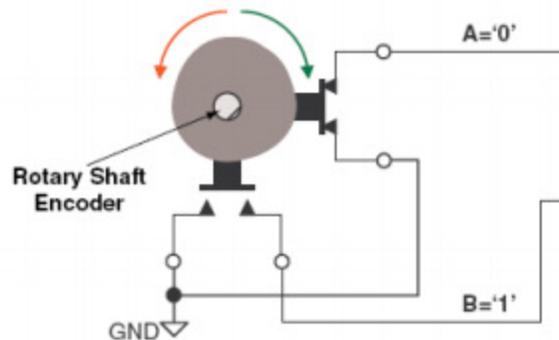


Introduction

The Digilent ENC library provides a programming interface for a device with a rotary push-button shaft encoder and a slide switch.

Overview

The rotary encoder behaves like a cam connected to a central shaft. Rotating the shaft operates two push-button switches, as shown in the below picture. Depending on which way the shaft is rotated, one of the switches closes before the other. Likewise, as the rotation continues, one switch opens before the other. When the shaft is stationary (detent position) both switches are open (logic 1)



Library Operation

Library Interface

The header file ENC.h defines the interfaces to the Encoder Module. The library is accessed via the methods and constants defined for the ENC object class. To instantiate an ENC object; simply include the library and instantiate an ENC object (e.g. myENC, or whatever name you want).

ENC Initialization

The ENC library is designed to use change notice pins to detect the position change on the rotary shaft. When an ENC object is instantiated, the change notice number is automatically associated with the physical change notice pin numbers that are located on each board. Before making calls to any other library functions, ENC.begin() must be called with the desired change notice numbers in order to enable change notice interrupts. To disable the change notice interrupt, simply call ENC.end(). For more information on the change notice number, please refer to the reference manual of your board.

Encoder Library Functions

Initialization Functions

void begin(unsigned int encA, unsigned int encB)

Parameters:

encA – the change notice number that is attach to pin A

encB – the change notice number that is attach to pin B

e.g. Using change notice 1 (CN1) on pin A, encA will simply be 1

Initializes the Change Notice interrupts

void end(void)

Parameters:

None

Return Value:

None

Turns Change Notice Interrupts off

int getDir(void)

Parameters:

None

Return Value:

Returns 1 of three defined values; left, right, and stationary.

ENC_Right -1

ENC_0 0

ENC_Left 1

After returning direction the direction will be set to ENC_0 until the encoder has been moved

void AttachInterrupt(void(*function)(int))

Parameters:

A pointer to a function with one parameter of int type

Assigns the passed in function to the end of the Interrupt Service Procedure so that every time the encoder turns the function will be called passing the direction the encoder

**void DettachInterrupt(void(*function)(int))**

Parameters:

A pointer to a function with one parameter of int type

Removes the function that was attached by AttachInterrupt().