Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

OFN Module (#27903)

The Optical Finger Navigation (OFN) Module can add a unique human interface component to your BASIC Stamp or Propeller projects. OFN Modules are quickly becoming popular as user input devices in many smart phones as a replacement for trackballs, which are subject to mechanical wear and tear.

OFN technology is very similar to the technology used in optical mice, and movement across the sensor can be read by any microcontroller using I^2C communication.

Features

- Built-in center select button
- Onboard red LED which lights when finger movement is detected
- Easy I^2C communication interface with virtually any microcontroller
- Onboard voltage regulator conditions I/O to 3 V max for compatibility with 3.3 V devices even when 5.5 V is supplied to Vdd
- User-selectable 500 or 1000 counts per inch (cpi) resolution
- User-definable 0 or 90° module orientation
- Breadboard-friendly package with 0.1" pin spacing

Key Specifications

- Power Requirements: 3.3-5.5 VDC; 25 mA active, 1 mA standby
- Communication: Two-Wire Serial I^2C (400 kHz)
- Operating temperature: 32 to 158 °F (0 to 70 °C)
- Dimensions: 1.0 x 1.0 x 0.55 in (25.4 x 25.4 x 13.97 mm)

Application Ideas

- Video game input
- Mouse replacement
- User input for computing devices

Quick-Start Guide

The instructions that follow outline how to wire and test that your OFN Module is operating correctly when used with the BASIC Stamp 2 and Propeller microcontrollers. The simple test code provided will read the product ID and display it on a serial terminal.

NOTE: Before using your OFN module, be sure to remove the plastic film placed over the optical sensor to allow the device to function properly.

Quick-Start Circuit

For use with the example programs included on the OFN Module product page and in the test programs included below.

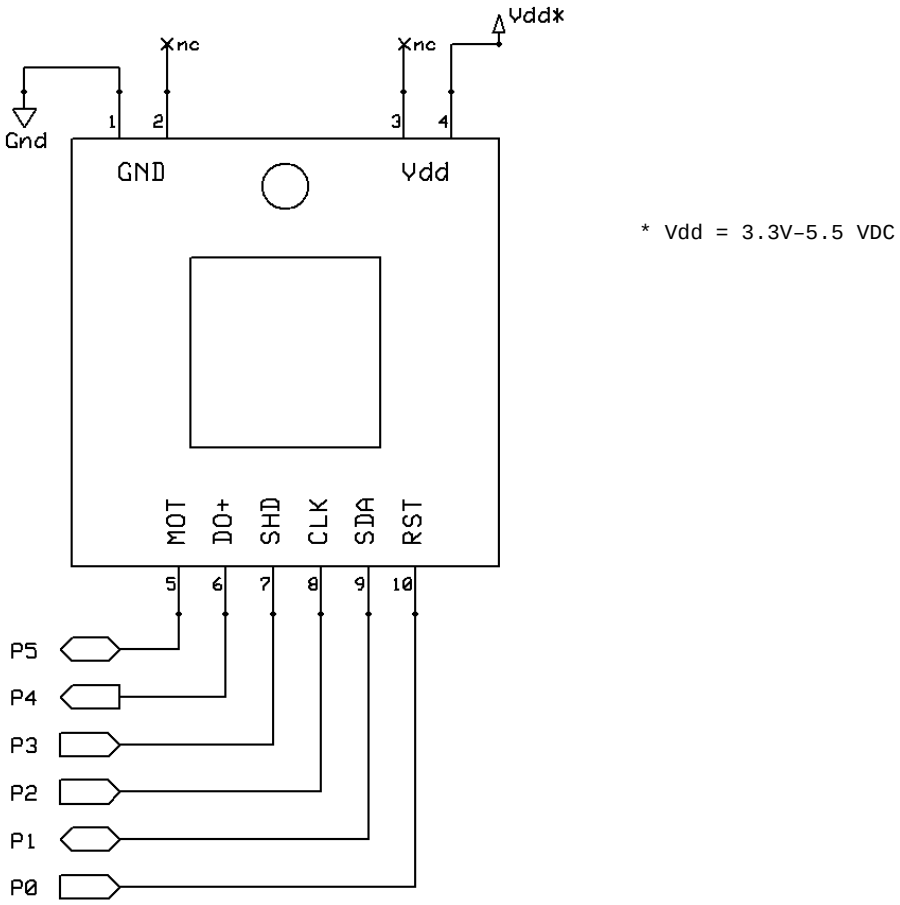

Figure 1: OFN Module connection diagram

BASIC Stamp 2 Test Code

This is a simple program designed to test serial communication between the BASIC Stamp 2 and the OFN Module. If working properly, the OFN Module should return $83 (the product ID). This program uses the Debug Terminal, which is built into the BASIC Stamp Editor software. The software is a free download from www.parallax.com/basicstampsoftware.

```
' OFNModule_Simple.bs2
' Tests serial communication between BS2 and OFN Module.

' {$STAMP BS2}
' {$PBASIC 2.5}

SDA       PIN     1                     ' Serial Data IO
CLK       PIN     2                     ' Serial Clock

ID        VAR     Byte                  ' Variable space for Product ID
I2CAck    VAR     Bit                   ' Acknowledge bit

INPUT SDA                               ' I2C Start Condition
```

```
INPUT CLK
LOW   SDA

SHIFTOUT SDA, CLK, MSBFIRST, [%01100110]  ' Set to write serial data
SHIFTIN  SDA, CLK, MSBPRE, [I2CAck\1]

SHIFTOUT SDA, CLK, MSBFIRST, [0]          ' Send address 0 to read Product ID
SHIFTIN  SDA, CLK, MSBPRE, [I2CAck\1]

INPUT SDA                                 ' I2C Start Condition
INPUT CLK
LOW   SDA

SHIFTOUT SDA, CLK, MSBFIRST, [%01100111]  ' Set to read serial data
SHIFTIN  SDA, CLK, MSBPRE, [I2CAck\1]

SHIFTIN SDA,CLK,MSBPRE,[ID\8]             ' Get Product ID
SHIFTOUT SDA,CLK,MSBPRE,[1]

LOW   SDA                                 ' I2C Stop Condition
INPUT CLK
INPUT SDA

DEBUG "Product ID (Hex): ", HEX ID        ' Display Product ID (default 83)

END
```

Propeller P8X32A Test Code

This is a simple program designed to test serial communication between the Propeller P8X32A and the OFN Module. If working properly, the OFN Module should return $83 (the product ID).

Note: This application uses the I2C.spin object, zipped in the OFNModule_PropellerDemo included on the OFN Module product page. It also uses the Parallax Serial Terminal to display the device output. The PST.spin object and the Parallax Serial Terminal itself are included with the with the Propeller Tool v1.2.7 or higher, which is available from the Downloads link at www.parallax.com/Propeller.

```
{{ OFNModule_Simple.spin:
Tests serial communication between Propeller P8X32A and OFN Module. }}

CON

  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

OBJ

  Debug : "Parallax Serial Terminal"
  I2C   : "I2C"

PUB Main : ID

  Debug.start(57600)                      ' Start PST & set Baudrate as 57600
  waitcnt(clkfreq/100 + cnt)              ' Wait for PST to start up
  Debug.Clear                             ' Clear the screen

  WriteRegister($00)                      ' Write operation to obtain Product ID
  ID := ReadRegister                      ' Retrieve Product ID

  debug.str(string("Product ID (Hex): "))  ' Display Product ID as hexadecimal value
  debug.hex(ID, 2)
```

```
PUB WriteRegister(RegisterAddress)

  I2C.DeviceStart                         ' Start Condition
  I2C.SendOut(%01100110)                  ' Write operation
  I2C.SendOut(RegisterAddress)            ' Write register address

PUB ReadRegister : RData

  I2C.DeviceStart                         ' Start Condition
  I2C.SendOut(%01100111)                  ' Read operation
  RData := I2C.Receive_NAck               ' Read data - no acknowledge
  I2C.DeviceStop                          ' Stop Condition
```

Theory of Operation

Optical Finger Navigation (OFN) Technology is very similar to the technology used for optical mice. When a finger moves across the surface of the OFN Module, an onboard LED is activated to light up the surface of the finger. The onboard Image Acquisition System then obtains microscopic images of the finger surface, and those images are processed by the Digital Signal Processor.

The Digital Signal Processor then mathematically determines the direction and magnitude of the finger's movement and calculates the delta-x and delta-y relative displacement values. A microcontroller can then read these values using simple I^2C communication. A block diagram of the OFN sensor can be seen in Figure 2, and an I^2C timing diagram can be found in Figure 3.

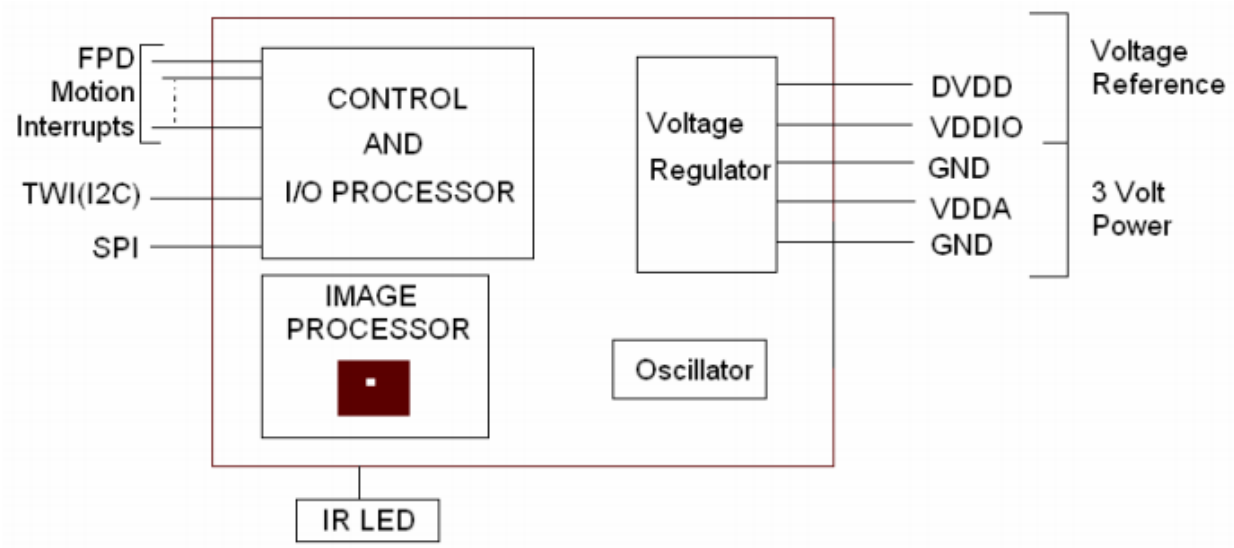

Figure 2: Optical Finger Navigation Senor Block Diagram

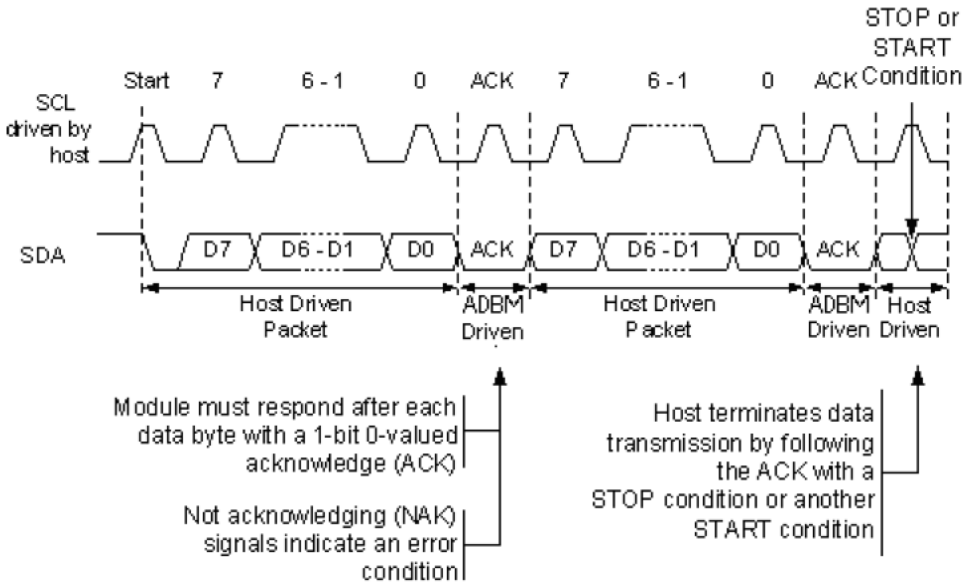

Figure 3: I²C Timing Diagram

Note that the OFN Module contains an onboard voltage regulator which conditions I/O to 3 V max, making the OFN module compatible with 3.3 V devices even when 5.5 V is supplied to Vdd.

Pin Definitions and Ratings

Pin	Name	Type	Function
1,2	GND	G	Ground -> 0V
3,4	Vdd	P	Supply Voltage -> 3.3 – 5.5 VDC
5	MOT	I/O	Motion Detect
6	DO+	O	Switch, Active Low
7	SHD	I	Shutdown
8	CLK	I	Serial Clock Input (400 kHz)
9	SDA	I/O	Serial Data Line
10	RST	I	Hardware Reset

Pin Type: P = Power, G = Ground, I = Input, O = Output

Register Table

Communicating with the OFN Module is done through an I^2C interface. A write operation is defined as data going from a microcontroller to the OFN Module, and a read operation is defined as data going from the OFN Module to a microcontroller. Below is a table of the most commonly used registers and their default values.

Address	Register	Read/Write	Default Value
0x00	Product_ID	R	0x83
0x01	Revision_ID	R	0x01
0x02	Motion	R/W	0x00
0x03	Delta_X	R	Any
0x04	Delta_Y	R	Any
0x05-0x10	Reserved	-	-
0x11	Configuration_Bits	R/W	0x00
0x12-0x39	Reserved	-	-
0x3a	Soft_RESET	W	0x00
0x3b-0x3d	Reserved	-	-
0x60-0x76	Reserved	-	-
0x77	OFN_Orientation_CTRL	R/W	0x01

Writing Data to OFN Sensor Registers

Certain Register Addresses allow for the configuration of certain aspects of the OFN Module, which require specific values to be written to that address. A detailed listing of selected configuration options is included below.

Register Address 0x02: Motion

Bit	7	6	5	4	3	2	1	0
Field	MOT	Reserved	Reserved	OVF	Reserved	Reserved	Reserved	GPIO

Data Type: Bit field
Usage: Allows the user to determine if motion has occurred since the last time it was read. If the MOT bit is set, then the user should read registers 0x03 and 0x04 to get the accumulated motion. Read this register before reading the Delta_Y and Delta_X registers.

Writing anything to this register clears the MOT and OVF bits, Delta_Y and Delta_X registers. The written data byte is not saved.

Internal buffers can accumulate more than eight bits of motion for X or Y. If either of the internal buffers overflows, then absolute path data is lost and the OVF bit is set. This bit is cleared once motion has been read from the Delta_X and Delta_Y registers, and if the buffers are not at full scale. Since more data is present in the buffers, the cycle of reading the Motion, Delta_X and Delta_Y registers should be repeated until the motion bit (MOT) is cleared. Until MOT is cleared, either the Delta_X or Delta_Y registers will read either positive or negative full scale. If the motion register has not been read for long time, at

500 cpi it may take up to 16 read cycles to clear the buffers, at 1000 cpi, up to 32 cycles. To clear an overflow, write anything to this register.

Field Name	Description
MOT	Motion since last report 0 = No motion 1 = Motion occurred, ready to read Delta_X & Delta_Y
OVF	Motion overflow, ΔY and/or ΔX buffer has overflowed 0 = No overflow 1 = Overflow has occured
PIXFIRST	Reports GPIO status (read only) 0 = Low 1 = High

Register Address 0x11: Configuration_Bits

Bit	7	6	5	4	3	2	1	0
Field	RES	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Data Type: Bit field
Usage: Selectable 500 or 1000 cpi resolution.

Field Name	Description
RES	Sets Resolution 0 = 500 1 = 1000

Register Address 0x77: OFN_Orientation_CRTL

Bit	7	6	5	4	3	2	1	0
Field	XY_SWAP	Y_INV	X_INV	Reserved	Reserved	Reserved	$ORIENT_1$	$ORIENT_0$

Data Type: Bit field
Usage: Sets sensor orientation control

Field Name	Description
XY_SWAP	0 = Normal sensor reporting of DX, DY (default) 1 = Swap data of DX to DY and DY to DX
Y_INV	0 = Normal sensor reporting of DY (default) 1 = Invert data of DY only
X_INV	0 = Normal sensor reporting of DX (default) 1 = Invert data of DX only
$ORIENT_{1:0}$	Read only bits of orientation pin 0x00 = Mounted 90° clockwise (orient pin low) 0x01 = Mounted 0° (default, orient pin is high)

Resources and Downloads

Check for the latest version of this document, free software, and the example programs listed below from the OFN Module product page. Go to www.parallax.com and search 27903.

Module Orientation

Figure 4 below demonstrates the proper orientation of the OFN Module for use with the example programs included the next sections.

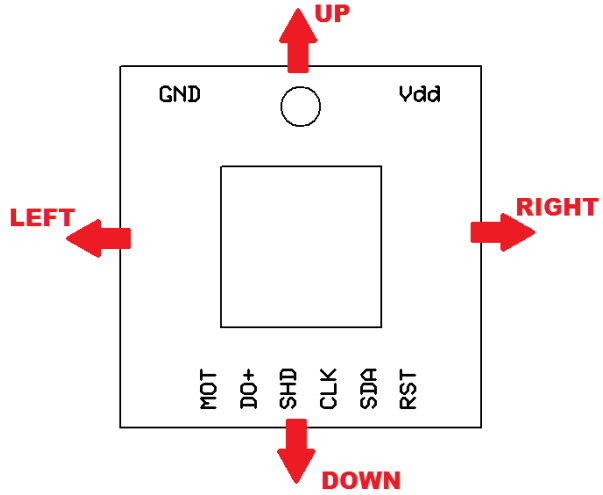

Figure 4: OFN Module Orientation for Example Programs

BASIC Stamp® Example Code

Also included on the OFN Module page is demo code to display movement along the surface of the OFN Module on the Debug Terminal. Figure 5 below shows the expected display on the Debug Terminal and the proper hardware orientation on a BASIC Stamp HomeWork Board. Visit the OFN Module's product page to download this code (OFNModule_Demo.bs2).

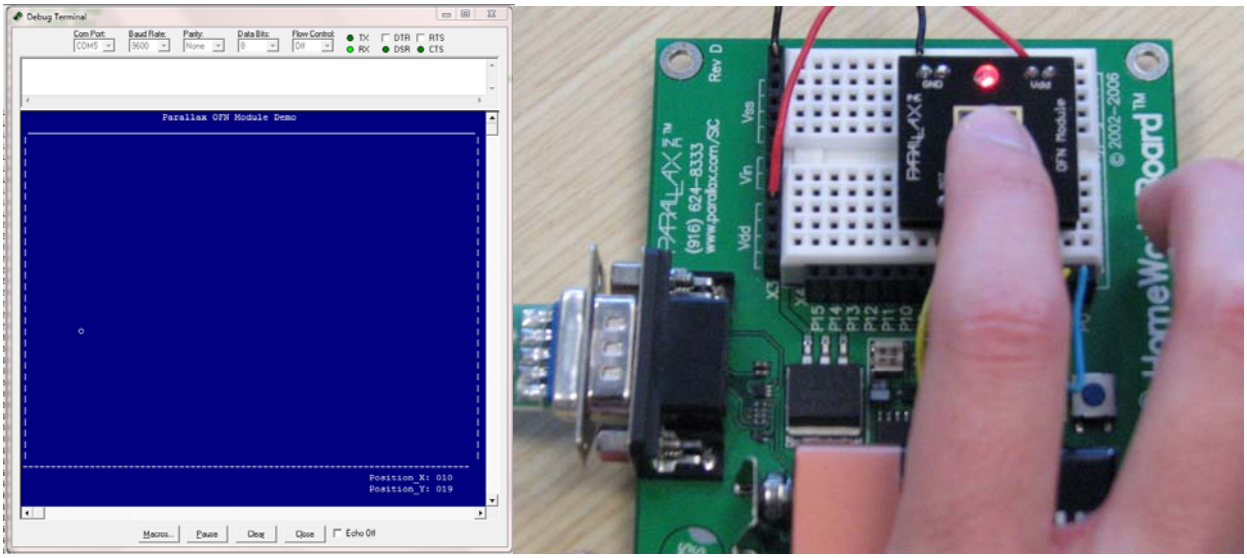

Figure 5: (Left) BS2 Debug Terminal Display (Right) BASIC Stamp HomeWork Board Hardware Orientation

Propeller™ P8X32A Example Code

Also included on the OFN Module page is demo code to display movement along the surface of the OFN Module on the Parallax Serial Terminal. Figure 6 below shows the expected display on the Parallax Serial Terminal and the proper hardware orientation on a Propeller Demo Board. Visit the OFN Module's product page to download this code (OFNModule_PropellerDemo.zip).

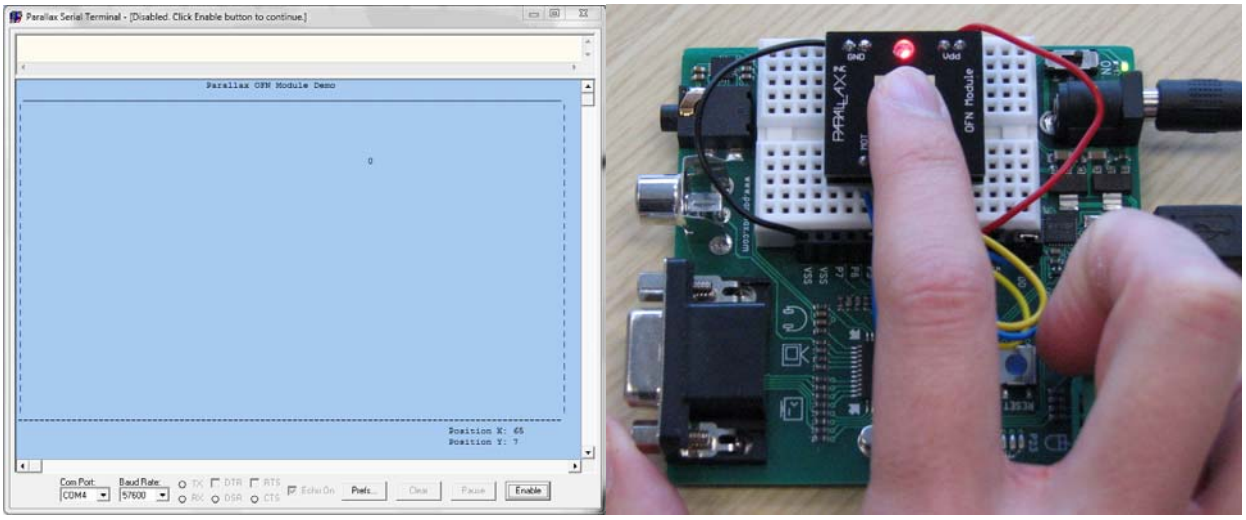

Figure 6: (Left) Propeller PST Display (Right) Propeller Demo Board Hardware Orientation

Revision History

Version 1.1

Section added: Writing Data to OFN Sensor Registers, page 6.