



**MoveAlong**

Hardware v1.1

Firmware v1.0

## **USER MANUAL and SPECIFICATIONS**

<b>Index</b>	<b>Page</b>
<b>1- Description</b>	<b>3</b>
<b>2- Hardware</b>	<b>3</b>
2.1- Hatch cable connector	<b>3</b>
2.2- Jumpers	<b>3</b>
2.3- LEDs	<b>4</b>
<b>3- I2C setup</b>	<b>4</b>
<b>4- I2C commands</b>	<b>5</b>
4.1- GET SPEED / DIRECTION	<b>5</b>
4.2- GET VERSION	<b>6</b>
4.3- SET ADDRESS	<b>6</b>
4.4- SET GAIN	<b>6</b>
4.5- SET HOLD / RELEASE	<b>7</b>
4.6- SET BLINK	<b>7</b>
<b>5- Analog setup</b>	<b>8</b>
<b>6- Specifications</b>	<b>8</b>

## 1- Description

MoveAlong senses direction and speed (actually, angular speed) of a human passing in front of it. It does this by means of a sensor similar to the ones used in burglar alarms.

A detection starts when a person enters the beam from left to right or vice versa (the movement should start outside of the beam).

The output value will be updated with the new one when the person is near the middle of the beam, this depends of the speed of the person and distance from the sensor.

If the axis projected from the middle off the sensor is called  $0^\circ$  in the horizontal plane; then, a detection starts at an angle of about  $\pm 11^\circ$  and the output value is updated at about  $0^\circ$ .

The end of the detection will take place once no movement has been detected.

If a second person enters the beam before the detection has ended, the measurement in progress will fail. The measurement will also fail if the person stops moving or starts moving very slow (the sensor can't detect slow movements)

The red LED will lit when a detection is in progress. It will turn off only when the sensor is ready for a new one.

The detection range is 5 meters.

Two interfaces are provided; a 0 to 3.3V analog one; and a I2C digital one.

In analog mode the sensor will output a voltage of 1.65V ( $= 3.3V / 2$ ) when no movement has been detected. Left to right movements will have voltages from 1.65V to 3.3V according to the angular speed. Right to left movements will have voltages from 1.65V to 0V according to the angular speed.

Similarly, in I2C mode, a 16 bit signed integer will read 0 with no movement, positive with left to right movements and negative with right to left movements according to the speed.

## 2- Hardware

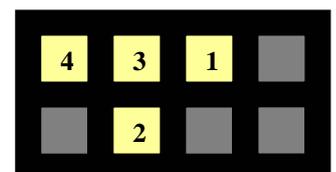
### 2.1- Hatch cable connector

Pins assignment in I2C mode:

1- Power	red	VCC
2- I2C_data	grey	input/output, open drain, Rpull-up=4k7
3- I2C_clock	white	input/output, open drain, Rpull-up=4k7
4- Ground	shield	GND

Pins assignment in analog mode:

1- Power	red	VCC
2- Analog_out	grey	output 0 to 3.3V, Rout=2k2
3- Unused	white	
4- Ground	shield	GND

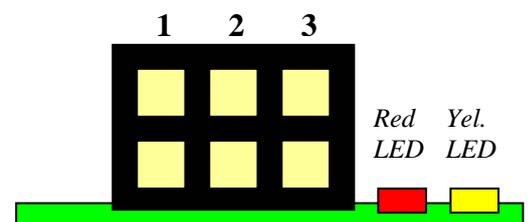


*Hack cable, 4 pins male connector ( front view)*

### 2.2- Jumpers

#### Jumper 1: Gain

Closed	Fast mode
Opened	Slow mode



*MoveAlong jumpers (side view)*

This jumper is used to select the gain value that will be used to amplify the sensors analog output.

In fast mode the gain is 2 (useful when sensing running or walking people)

In slow mode the gain is 4 (useful when sensing walking or slow walking people)

*This jumper is used only in analog mode (jumper 3 closed)*

#### **Jumper 2: Output Behaviour**

Closed            Hold mode

Opened           Release mode

In Hold mode the last output value is sustained in time until a new movement is detected.

In Release mode, the output value will return to 1.65V once the sensor is ready for a new measurement.

*This jumper is used only in analog mode (jumper 3 closed)*

#### **Jumper 3: Interface**

Closed            Analog mode

Opened           I2C mode

This jumper selects the interface to be used. Also, when the sensor is powered up with this jumper closed, all the I2C EEPROM configuration values will be set to the default value.

### **2.3- LEDs**

The red LED indicates that a detection is in progress.

The yellow LED indicates I2C communication.

LEDs activities can be turn off (and on) by an I2C command. In analog mode the LEDs activities is always on.

### **3- I2C setup**

If you are unfamiliar with the microdig I2C's commands, look for the host mode commands: "I2C PORT SET", "I2C WRITE" and "I2C READ" in the document "Wi-microdig communication protocol.pdf", before going on.

1- Close jumper 3; I2C mode.

2- The default I2C address is 1. If you have changed the address and don't remember it or if the sensor is not answering, follow the next procedure which will set the EEPROM values to the default ones:

A - Turn off MoveAlong and open jumper 3 (analog mode)

B - Turn on MoveAlong, the address is now 1.

C - Turn off MoveAlong and close jumper 3 (I2C mode)

Default values:

Address	1
Gain	1
Hold/Release	0 ( Hold output value )
Blink	1 ( Blink LEDs )

3- Plug the hatch cable 4 pins male connector (power & I2C bus) to the microdig.

4- Power up the microdig.

5- Reset the microdig.

6- Set the microdig in host mode.

7- Configure/select the microdig ports where the I2C bus has been plugged with the command "I2C PORT SET". This command should be sent each time you want to use a different I2C port before sending any command to MoveAlong in that I2C bus.

## 4- I2C commands

If you are using a microdig as I2C master, the following commands should be sent using the "I2C WRITE" and "I2C READ" commands.

The MoveAlong I2C commands are based on getting or setting registers values. The register's size may vary according to the data stored in it.

### 4.1- GET SPEED / DIRECTION (read register 0, whose size is 2 bytes)

The data returned is the speed and direction in a 16 bit signed integer (2's complement)  
The range used by the sensor is [ -512 : 511], or [ FE00h : 01FFh], where the most significant bit is the sign.

511	01FFh
510	01FEh
...	
1	0001h
0	0000h
-1	FFFFh
...	
-511	FE01h
-512	FE00h

Example: "Get speed / direction" (decimal notation)

Send:

240, 125, 0 {DEV}, 127 {I2C READ}, 1 {address of I2C device}, 0, 0 {register 16\*0+0=0}, 2 {bytes to be read}, 247

Receive:

240, 125, 0 {DEV}, 127 {I2C READ}, 1 {address of I2C device}, 0, 0 {register 16\*0+0=0}, 2 {bytes to be read}, 15, 14, 12, 9 {SPEED=4096\*15+256\*14+16\*12+1\*9=65225}, 247

As 65225 (FEC9h) is greater than 32767 (7FFFh) (bit 16 is '1') the value is negative, and it's 2's complement needs to be computed:

First subtract -1	FEC9h – 0001h = FEC8h
Then negate the value	~FEC8h = 0137h = 311
Add the negative sign symbol	-311

Example: "Get speed / direction" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Fh {I2C READ}, 01h {address of I2C device}, 00h, 00h {register 00h}, 02h {bytes to be read}, F7h

Receive:

F0h, 7Dh, 00h {DEV}, 7Fh {I2C READ}, 01h {address of I2C device}, 00h, 00h {register 00h}, 02h {bytes to be read}, 00h, 01h, 03h, 04h {SPEED=0134h}, F7h

The value is 0134h = 308

#### 4.2- GET VERSION (read register 1, whose size is 1 byte)

The byte returned is the software release. The first nibble is the mayor number an the second nibble, the minor.

Example: "Get version" (decimal notation)

Send:

240, 125, 0 {DEV}, 127 {I2C READ}, 1 {address of I2C device}, 0, 1 {register  $16*0+1=1$ }, 1 {bytes to be read}, 247

Receive:

240, 125, 0 {DEV}, 127 {I2C READ}, 1 {address of I2C device}, 0, 1 {register  $16*0+1=1$ }, 1 {bytes to be read}, 1, 0  
{VERSION =  $16*1+0=16$ }, 247

Example: "Get version" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Fh {I2C READ}, 01h {address of I2C device}, 00h, 01h {register 01h}, 01h {bytes to be read}, F7h

Receive:

F0h, 7Dh, 00h {DEV}, 7Fh {I2C READ}, 01h {address of I2C device}, 00h, 01h {register 01h}, 01h {bytes to be read}, 01h, 00h {VERSION = 10h}, F7h

#### 4.3- SET ADDRESS (write register 2, whose size is 1 byte)

This byte sets the sensor I2C address. The value received will be masked (and'ed) with 0X7F; then, the range is 1 to 127 (01h to 7Fh). The address will be stored into a non volatile memory.

Example: "Set address = 15" (decimal notation)

Send:

240, 125, 0 {DEV}, 126 {I2C WRITE}, 1 {address of I2C device}, 0, 2 {register  $16*0+2=2$ }, 1, 0  
{ADDRESS= $16*0+15=15$ }, 247

Receive:

(echo of the sent data)

Example: "Set address = 10h" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Eh {I2C WRITE}, 01h {address of I2C device}, 00h, 02h {register=02h}, 01h, 00h  
{ADDRESS=10h}, F7h

Receive:

(echo of the sent data)

#### 4.4- SET GAIN (write register 3, whose size is 1 byte)

The value written into this register will set the gain of the sensor. The range should be in between 0 to 3 (01h to 03h). The register value will be stored into a non volatile memory.

These four values [0, 1, 2, 3] correspond to gains of [1, 2, 4, 8] respectively.

Example: "Set gain = 2" (decimal notation)

Send:

240, 125, 0 {DEV}, 126 {I2C WRITE}, 1 {address of I2C device}, 0, 3 {register  $16*0+3=3$ }, 0, 2 {GAIN= $16*0+2=2$ }, 247

Receive:

(echo of the sent data)

Example: "Set gain = 03h" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Eh {I2C WRITE}, 01h {address of I2C device}, 00h, 03h {register 03h}, 00h, 03h {GAIN=03h}, F7h

Receive:

(echo of the sent data)

#### 4.5- SET HOLD / RELEASE (write register 4, whose size is 1 byte)

Select whether the output register will hold the last speed value or release it. Write 0 (00h) to hold the output value or write any other value to release it. This value is stored in non volatile memory.

Example: "Set Release mode" (decimal notation)

Send:

240, 125, 0 {DEV}, 126 {I2C WRITE}, 1 {address of I2C device}, 0, 4 {register  $16*0+4=4$ }, 0, 1 {HOLD\_REL =  $16*0+1=1$ }, 247

Receive:

(echo of the sent data)

Example: "Set Hold mode" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Eh {I2C WRITE}, 00h {address of I2C device}, 00h, 04h {register 04h}, 00h, 00h {HOLD\_REL = 00h}, F7h

Receive:

(echo of the sent data)

#### 4.6- SET BLINK (write register 5, whose size is 1 byte)

Enable or disable LEDs. Write 0 (00h) to disable the LEDs or write any other value to enable them. This value is stored in non volatile memory.

Example: "Set on blink mode" (decimal notation)

Send:

240, 125, 0 {DEV}, 126 {I2C WRITE}, 1 {address of I2C device}, 0, 5 {register  $16*0+5=5$ }, 0, 1 {BLINK= $16*0+1=1$ }, 247

Receive:

(echo of the sent data)

Example: "Set off blink mode" (hexadecimal notation)

Send:

F0h, 7Dh, 00h {DEV}, 7Eh {I2C WRITE}, 01h {address of I2C device}, 00h, 05h {register 05h}, 00h, 00h {BLINK=00h}, F7h

Receive:

(echo of the sent data)

## 5- Analog setup

- 1- Open jumper 3; analog mode (Please note that after doing this and powering the sensor, the I2C configuration values will be set to the default ones)
- 2- Set jumper 2 with the desired output mode.
- 3- Set jumper 1 with the desired gain value.
- 4- Plug the hatch cable male connector to the microdig.
- 5- Power up the microdig.
- 6- Reset the microdig.
- 7- Set the microdig in stand alone mode.
- 8- Configure the microdig ports as analog inputs with the desired resolution.

## 6- Specifications

<b>Sensors Parameters</b>	<b>min</b>	<b>typical</b>	<b>max</b>
Range	0.5m		5m
Beam angle		22°	
Detectable angular speed	10°/s		511°/s
Accuracy		$(\text{ang.speed } ^\circ/\text{s})^2 / 5371^\circ/\text{s}$	
Resolution		1°/s	

<b>Electrical Parameters</b>	<b>min</b>	<b>typical</b>	<b>max</b>
Supply voltage (VCC)	4.75V		12V
Supply Current		12mA	
I2C input, high voltage threshold	1.98V		
I2C input, low voltage threshold			0.66V
I2C pullup resistors		4K7Ω	
I2C output, low voltage (Io 10mA, Vcc=3.3V)			0.5V
I2C output, sink current (Vo 0.5V, Vcc=3.3V)			10mA
Analog output, voltage range	0V		3.3V
Analog output, DC equivalent resistance		2K2Ω	

<b>Timing</b>	<b>min</b>	<b>typical</b>	<b>Max</b>
I2C clock frequency			50kHz
I2C time between commands	10ms		
Analog output, response time (10 to 90%)		50ms	