Small Stepper Motor and Driver Board

See this product here:
See 1-turn Arduino Sketch HERE:

MOTOR DETAILS

This is a 5v 28YBJ-48 Stepper Motor with Gear Reduction, so it has good torque for its size, but relatively slow motion. These motors/drivers are made by the millions for A/C units, fans, duct controls etc. which is why they are so inexpensive.

4 Phase 5 Wire Connection

- 100% Brand New
- Phase : 4
- Current : 160 mA per winding (320 mA in 4-step mode) Measured: 250mA stopped, 200 mA running fast
- Resistance : 31 Ω per coil winding (from Red wire to any coil) (Some 24-28 ohms)
- Voltage : 5V DC
- Step Angle (8-Step sequence: Internal Motor alone): 5.625° (64 steps per revolution)
- Step Angle (4-Step sequence: Internal Motor alone): 11.25° (32 steps per revolution)
- Gear Reduction ratio: 1 / 64 (Not really exact: probably 63.68395.:1 )
- SO: it takes (64*64 = 4096 steps per output shaft revolution.. In 8-step sequence.
- SO: it takes (32*64 = 2048 steps per output shaft revolution.. In 4-step sequence.
- NOTE: Arduino "Stepper Library" runs in 4-step mode
- No-Load Pull-Out Frequency : 800pps
- No-Load Pull-In Frequency : 500pps
- Pull-In Torque : ≥ 78.4mN.m
- Wiring Instruction : A (Blue), B (Pink), C (Yellow), D (Orange), E (Red, Mid-Point)
- Weight : 30g
See wiring diagram, following...

**CONNECTION NOTES:**

**NOTE:** If your motor vibrates but does not turn, it is probably connected with the wrong sequence.

The Arduino pin connections need to have 4 pins connected to Motor Driver In1, In2, In3, In4 and then the pins entered in the software in the sequence 1-3-2-4 for proper sequencing. Also, The + and - pins near "5-12V" need to be connected: - to Arduino Ground, + to Arduino +5 (for one motor test only) or (best) to a separate +5V 1A power supply.

Example: Connect Arduino Pins 8,9,10,11 to In1,In2,In3,In4

Then **software** is initialized in 1-3-2-4 sequence:

```c
Stepper small_stepper(STEPS, 8, 10, 9, 11);
```

//Example Software Sketch below.

**STEP SEQUENCES:**

The motor moves in response to the sequence in which the internal electromagnets are turned on.
There are two possible sequences. In 4-step (Used by the Arduino Stepper Library) there are always 2 of the 4 magnet coils turned on, and only one coil changes each step.

The following refers to the letters A-B-C-D printed on the Stepper Driver Board which are controlled by the input pins 1-2-3-4. There are 4 LEDs next to the letters and they will follow the sequences. The test software sketch start out with a very slow sequence of the 4 step sequence. Push RESET on your Arduino to see that startup again.

4 Step Sequence:
AB-BC-CD-DA (Usual application using Arduino STEPPER Library)

The 8-step sequence uses only 1 coil on, then 2, then 1... etc
8 Step: A - AB - B - BC - C - CD - D - DA - A (Can be done with other Libraries).

Here is test code for this Driver Board and Motor. Cut and Paste into a blank Arduino IDE page. NOTE: It uses the 4-step sequence shown above.
2048 steps per output shaft revolution. This uses the standard built-in Stepper library: http://arduino.cc/en/Reference/Stepper. Other higher-performance libraries are available. See:

- **AccelStepper:**
  - Allows ramping up and down speeds, multiple motors, power down, more.
  - See Example 2-motor sketch below

- **Steve Bright's Stepper2**
  - Many functions, non-blocking modes, power control

See
INSTALLING LIBRARIES
HERE:

Test
Sketch:
4-step
sequence,
Then
1/2
turn
forward
slow
and
back
1/2
turn
fast
/*
YourDuino.com
Example
Software
Sketch
Small
Stepper
Motor
and
Driver
V1.3
11/30/2013
http://www.yourduino.com
/sunshop
/index.php?id=product_detail&p=126
Shows
4-step
sequence,
Then
1/2
turn
and
back
different
speeds
terry@yourduino.com
*/
/*-----(
#include

/*-----*/

Declare
Constants,
Pin
Numbers
)-----*/

//---

Number
of
steps
per
revolution
of
INTERNAL
motor
in
4-step
mode
)--

#define
STEPS_PER_MOTOR_REVOLUTION
32

//----

Steps
per
OUTPUT
SHAFT
of
gear
reduction
)--

#define
STEPS_PER_OUTPUT_REVOLUTION
32

* 64

//2048

/*-----*/

Declare
objects
)-----*/

//
create
an
instance
of
the
stepper
class,
specifying
//
the
number
of
steps
of
the
motor
and
the
pins
it's
//
attached
to
//The
pin
credentials
need
to
be
4
pins
connected
//
to
Motor
Driver
In1,
In2,
In3,
In4
and
then
the
pins
entered
//
here
in
the
sequence
1-3-2-4
for
proper
sequencing
Stepper
small_stepper(STEPS_PER_MOTOR_REVOLUTION,
8,
/*-----(*
Declare
Variables
)-----*/
int
Steps2Take;
void
setup()
/*----(*
SETUP:
RUNS
ONCE
)----*/
{
//
Nothing
(Stepper
Library
sets
pins
as
outputs)
}/*--(end
setup
)----*/
void
loop()
/*----(*
LOOP:
RUNS
CONSTANTLY
)----*/
{
small_stepper.setSpeed(1);
//
SLOWLY
Show
the
4
step
sequence
Steps2Take
=
4;
//
Rotate
CW
small_stepper.step(Steps2Take);
delay(2000);

Steps2Take = STEPS_PER_OUTPUT_REVOlUTION / 2;

// Rotate CW 1/2 turn
small_stepper.setSpeed(100);
small_stepper.step(Steps2Take);
delay(1000);

Steps2Take = - STEPS_PER_OUTPUT_REVOlUTION / 2;

// Rotate CCW 1/2 turn
small_stepper.setSpeed(700);
// 700 a good max speed??
small_stepper.step(Steps2Take);
delay(2000);

*/
--(end main loop)
*/
/*
 ( THE END )
 */[/code]

Test
Rotate 1 turn in each direction, repeat. So this is a repeatable "back and forth" motion and is not dependent on the exactly precise gear ratio.

/* YourDuino.com E Small Stepper Motor
http://arduino-steps.sourceforge.net*/

//---( Import necessary libraries
#include <Stepper.h>

//---( Declare constants
#define STEPS_PER_REVOLUTION 200
#define STEPS_PER_OCTAVE 40

// create an instance of the stepper motor connected to the number of stepper drivers
Stepper myStepper(STEPS_PER_REVOLUTION, STEPS_PER_OCTAVE);

void setup()
{
  // Nothing (Stepper setup)
}

void loop()
{
  // Nothing (Stepper loop)
}
Test Sketch:
Runs 2 Stepper motors in opposite directions, accelerates, decelerates, reverses. REQUIRES the Accelstepper Library HERE
See INSTALLING LIBRARIES HERE:
```c
#define FULLSTEP 4
#define HALFSTEP 8

// motor pins
#define motorPin1
#define motorPin2
#define motorPin3
#define motorPin4
#define motorPin5
#define motorPin6
#define motorPin7
#define motorPin8

/*-----( Declare o
NOTE: The seque
AccelStepper stepp
AccelStepper stepp
/*-----( Declare V

//none

void
{
stepper1.setMaxSl
stepper1.setAcce
stepper1.
stepper1.moveTo(
stepper2.setMaxS
stepper2.setAcce
stepper2.
stepper2.moveTo(
}
//--(end setup )-

void
{
stepper1.moveT
stepper2.moveT
stepper1.run();
stepper2.run();
}
//--(end main loo

/*-----( Declare U
//none
//*********( THE E
```