Let’s Get Cracking! Force Sensor Instructions
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HX711 Amplifier (Weight Weighing Load Cell Conversion Module for Arduino Microcontroller)</td>
<td>$7.69</td>
</tr>
<tr>
<td>1</td>
<td>Mustcam 5 Megapixel USB Digital Microscope with Measurement Software for Windows/Mac</td>
<td>$39.99</td>
</tr>
<tr>
<td>1</td>
<td>Adafruit Motor/Stepper/Servo Shield for Arduino v 2.3 Kit</td>
<td>$19.95</td>
</tr>
<tr>
<td>1</td>
<td>Arduino UNO</td>
<td>$22.00</td>
</tr>
<tr>
<td>1</td>
<td>400mm Length Travel Linear Stage Actuator with Square Linear Rails + CBX1605 Ball Screw 1605 Ballscrew Motorized XY XYZ Linear Stage Table with Nema23</td>
<td>$132.28</td>
</tr>
<tr>
<td>1</td>
<td>Digi-Key AC/DC 3.3V 6.6W Power Supply</td>
<td>$13.47</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td><strong>$249.37</strong></td>
</tr>
</tbody>
</table>
TO STEPPER MOTOR
BLACK GND to GND
GREEN DT to A1
WHITE SCK to A0
RED VCC to 5V

TO DC BARREL JACK

50 KG LOAD CELL SENSOR
AC/DC 3.3V 6.6W POWER SUPPLY

HX711 AMPLIFIER
RED E+ BLACK E- WHITE A- GREEN A+

NEMA23 STEPPER MOTOR ON RAILS

ADAFRUIT MOTOR SHIELD V2.3 MOUNTED TO ARDUINO UNO

AC/DC 3.3V 6.6W POWER SUPPLY

VIN JUMPER REMOVED

FORCE SENSOR WIRING DIAGRAM
Version: June 2019
HX711 on PCB

Motor shield on Arduino UNO

Version: June 2019
Motor shield on Arduino UNO
The Software Side of Things…

1. Connect the Arduino UNO Board to your computer using the USB cable.

2. You will need to download and install two libraries (see https://www.arduino.cc/en/Guide/Libraries):
   - Adafruit_Motor_Shield_V2_Library-master.zip
   - HX711-master.zip
   ```
   #include <HX711.h>
   #include <Adafruit_MotorShield.h>
   ```

3. You will need to include them in the Arduino sketch.

4. Use the NoBonesAboutIt.ino code shown on the next page or write your own.

5. Check for compiling errors and then upload to the Arduino board.

6. Connect the power supply to the motor shield to power the motor.

7. Calibrate the scale using the procedure (in two pages).

8. After you set the scale calibration value in the code, you will need to re-compile NoBonesAboutIt.ino and upload to the Arduino board again.

9. Finally, open a Serial Window (newline) and type type U and D for up and down using the interleave stepping function or u and d to use the microsetepping function; in either case, you will be prompted to enter the number of steps. Type t for tare and m for measure.

Version: June 2019
```
#include <HX711.h>
#include <Adafruit_MotorShield.h>

//-------------
Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x60);
// Connect a stepper motor with 200 steps per revolution
// which is 1.8 degrees
// use motor port #2, M3 and M4

Adafruit_StepperMotor *M2 = AFMS.getStepper(20, 2);
// HX711.DOUT uses pin #A1
// HX711.PD_SCK uses pin #A0
HX711 scale(A1, A0);             // using default parameter gain

void setup() {
  AFMS.begin();           // use default frequency
  M2->setSpeed(10);  // 10 rpm
  Serial.begin(9600);
  scale.set_scale(79.1156);              // this value is obtained by calibrating the scale with
  scale.tare();                // reset the scale to 0
  Serial.println("Enter 'U', 'D', 'u', 'd', 't', or 'm'.");
}
```
1. Use Arduino to open the *NoBonesAboutIt* program.

2. Remove the `scale.set_scale()` number. You will replace it later at the end of this procedure.

3. Type `t` for tare. This is your (0, 0) starting point in the graph shown below.

4. Add known weights, beginning with a basket for holding weights and measure each successive force reading. I attached some embroidery floss to a plastic hummus cup as my basket.

5. After verifying that the results are linear, find the slope of the line. This is your scale factor.

6. Add the number to the `scale.set_scale(79.1156)` line and upload the updated code to the board.

**Calibrating the Scale**

<table>
<thead>
<tr>
<th>Item</th>
<th>Item Weight</th>
<th>Total Weight(g)</th>
<th>Force(g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Basket</td>
<td>23.9</td>
<td>23</td>
<td>2116</td>
</tr>
<tr>
<td>Plug</td>
<td>65.1</td>
<td>89</td>
<td>7203</td>
</tr>
<tr>
<td>Plug</td>
<td>64.9</td>
<td>153.9</td>
<td>12360</td>
</tr>
<tr>
<td>Plug</td>
<td>65.1</td>
<td>219</td>
<td>17507</td>
</tr>
<tr>
<td>Plug</td>
<td>64.9</td>
<td>283.9</td>
<td>22593</td>
</tr>
<tr>
<td>Plug</td>
<td>64.8</td>
<td>348.7</td>
<td>27750</td>
</tr>
<tr>
<td>Plug</td>
<td>65</td>
<td>413.7</td>
<td>32899</td>
</tr>
<tr>
<td>Plug</td>
<td>65.2</td>
<td>478.9</td>
<td>37890</td>
</tr>
<tr>
<td>Plug</td>
<td>65</td>
<td>543.9</td>
<td>43031</td>
</tr>
</tbody>
</table>

\[
\text{SCALE FACTOR} = \text{slope} = \frac{(43031 - 0)}{(543.9 - 0)} = 79.1156
\]

---

**void setup()**

- `AFMS.begin();` // use default frequency
- `M2->setSpeed(10);` // 10 rpm
- `Serial.begin(9600);`
- `scale.set_scale(79.1156);` // this value is obtained by calibrating the scale