MEASUREMENT OF FORCE USING STRAIN GAUGES

A COURSE-BASED PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT FOR THE LAB EVALUATION OF SENSORS AND TRANSDUCERS LAB

FOR

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS & INSTRUMENTATION ENGINEERING

Submitted by

M. BALA KRISHNA	23071A10A5
P.DHANANJAY	23071A10B9
S. TEJA	23071A10C1

Estd. 1995

DEPARTMENT OF ELECTRONICS & INSTRUMENTATION ENGINEERING

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

AICTE Approved; UGC Autonomous; JNTUH Affiliated; UGC "College with Potential for Excellence"; NAAC "A++" Grade
ISO 9001:2015 Certified; QS I.GAUGE "Diamond" Rated; NIRF 2019: 109th Rank Engineering (151–200 Band Overall)
NBA Accredited: CE, CSE, ECE, EEE, EIE, IT, ME, AE; JNTUH-Recognised Research Centres: CE, CSE, ECE, EEE, ME

2023-2024

MEASUREMENT OF FORCE USING STRAIN GAUGES

I. Introduction

A. Theory

A method often used for weighing is the strain-gauge balance. A strain gauge is a (often zigzag shaped) conductor, etched out of thin metal film. It uses the effect of change in the electrical resistance when it is subjected to mechanical stress, caused by change in length, cross section and specific resistance. Usually, four strain gauges are glued to a "spring body" (flexural component), which is manufactured from a single part, as shown below with its electrical circuit:

The strain gauges are wired as a Wheatstone-bridge to compensate for temperature changes. When the system is not loaded all four resistors are the same and the input of the amplifier is zero. When an object is placed on the pan R1 and R4 are compressed and their resistance decreases, R2 and R3 are strained and their resistance is increased. This causes a voltage difference at the input of the amplifier, proportional to the weight of the object.

The shape if this spring body is comparable with the basic construction of the electronic balance with two guides. The strain-gauge method of measurement has its limitations for high resolution weighing machines, which are primarily due to creep in the spring material and the adhesive between the spring body and the strain gauges. The moisture sensitivity of the adhesive and the low output signal also cause difficulties. The major advantages of this method are the compact design, cost and its easy adaptability to various maximum capacitance

B. Objective

This project outlines the operation of a strain gauge-based load cell system integrated with an Arduino microcontroller, LCD display, HX711 amplifier, and two buttons for user interaction. Strain gauges mounted on the load cell measure mechanical strain induced by applied force, causing a change in electrical resistance. The HX711 amplifier amplifies and converts this resistance change into a measurable electrical signal compatible with the Arduino. The Arduino processes this signal and calculates the applied force or weight. User interaction is facilitated through two buttons, allowing functions such as calibration or unit selection. The measured weight data is displayed on the LCD screen, providing real-time feedback to the user. This project showcases the utilization of strain gauge technology in conjunction with Arduino-based electronics for weight measurement applications, offering versatility and ease of implementation in various scenarios, including industrial automation, smart scales, and embedded systems.

The major features of this project are:

1. Design a strain gauge.
2. Automatic mass calibration adding.
3. Monitoring the weight on LCD display.
4. To achieve this task using Arduino UNO microcontroller.

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers. Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

The project “STRAIN GAUGE” is an exclusive project that can detect the weight using loadcell will be display on LCD.

II. Implementation

A. Hardware Requirements

The main blocks of this project are:

1. Power Supply.
2. Arduino UNO Micro Controller.
3. HX711 Load cell.
4. LCD display with driver.

1. Arduino UNO:

The Arduino Uno is a microcontroller board which has ATmega328 from the AVR family. There are 14 digital input/output pins, 6 Analog pins and 16MHz ceramic resonator.

USB connection, power jack and also a reset button is used. Its software is supported by a number of libraries, which makes the programming easier.

Features

• High Performance, Low Power AVR® 8-Bit Microcontroller

• Advanced RISC Architecture

– 131 Powerful Instructions

– Most Single Clock Cycle Execution

– 32 x 8 General Purpose Working Registers

– Fully Static Operation

– Up to 20 MIPS Throughput at 20 MHz

– On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory Segments

• **Peripheral Features**

– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Real Time Counter with Separate Oscillator

– Six PWM Channels – 8-channel 10-bit ADC in TQFP and QFN/MLF package

Temperature Measurement

– 6-channel 10-bit ADC in PDIP Package

Temperature Measurement

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Byte-oriented 2-wire Serial Interface (Philips I2C compatible)

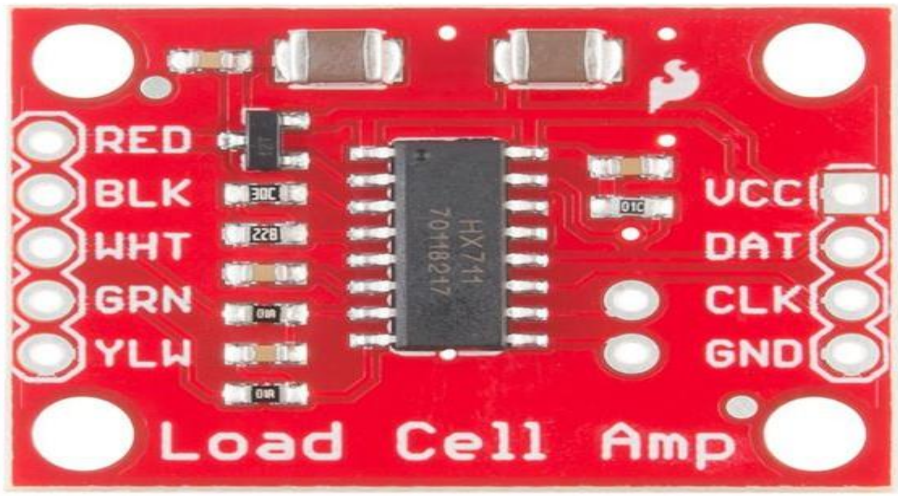
– Programmable Watchdog Timer with Separate On-chip Oscillator

– On-chip Analog Comparator

– Interrupt and Wake-up on Pin Change

2. HX711Loadcell:

The HX711 load cell amplifier is used to get measurable data out from a load cell and strain gauge. This Hookup Guide will show you how to get started with this amplifier using some of the various load cells we carry at SparkFun.



For this simple hook up guide we will just be hooking up a load cell with the HX711 amplifier, and showing how you would hook up four load cells with a combinator board and the HX711 amplifier. To follow along, you'll need:

• SparkFun Load Cell Amplifier - HX711

• Any Strain Gauge Based Load Cell:

If you are planning on using load sensors1 you will need to obtain or purchase four units. We recommend our Combinator Board to make it easy to turn the four strain gauges into a wheatstone bridge type load cell. (Single strain gauge load cells only have three wires instead of four.)

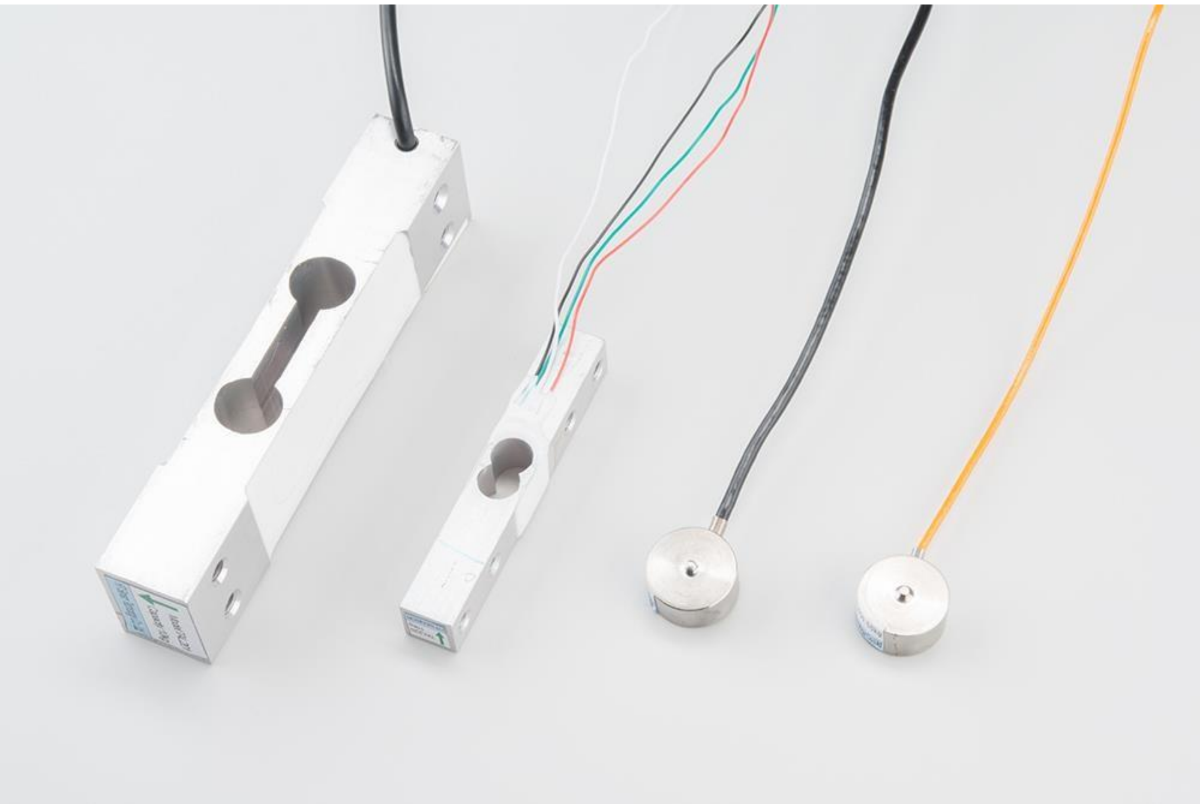
• (Optional) Combinator Board

Suggested Reading

If you aren’t familiar with the following concepts, we recommend reviewing them before beginning to work with the HX711 Load Cell Amplifier Board.

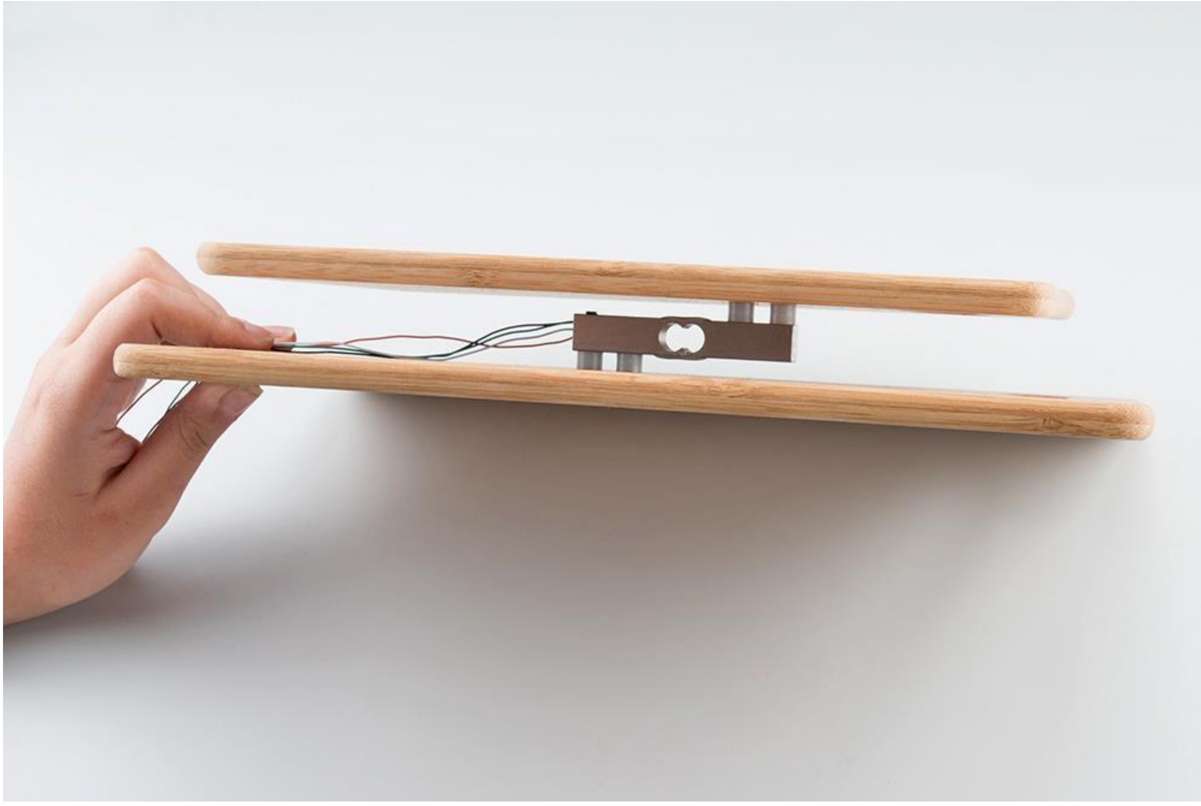
- Load Cell Basics
- Getting Started with Load Cells
- Installing the Arduino IDE
- How to Power Your Project
- Battery Technologies
- How to Solder

Load Cell Set Up

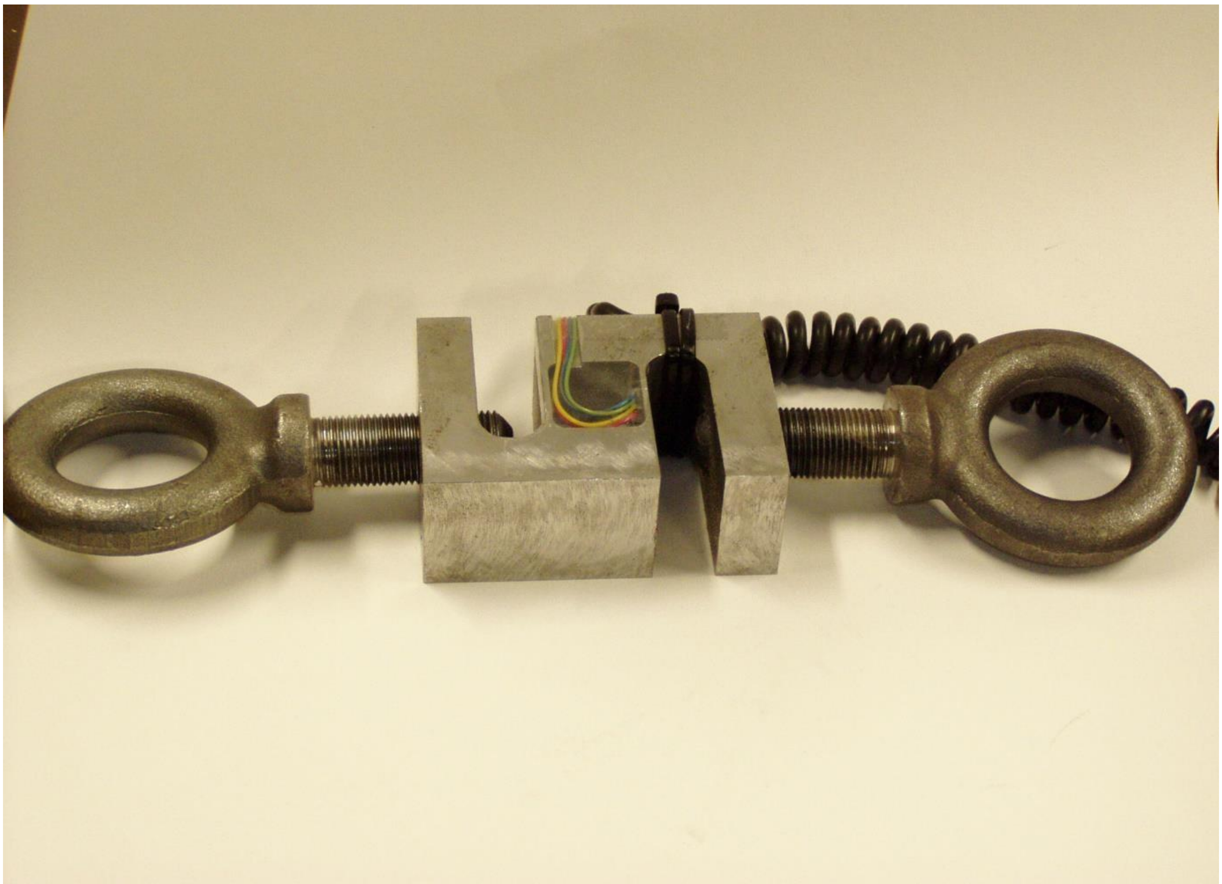


A selection of different load cells

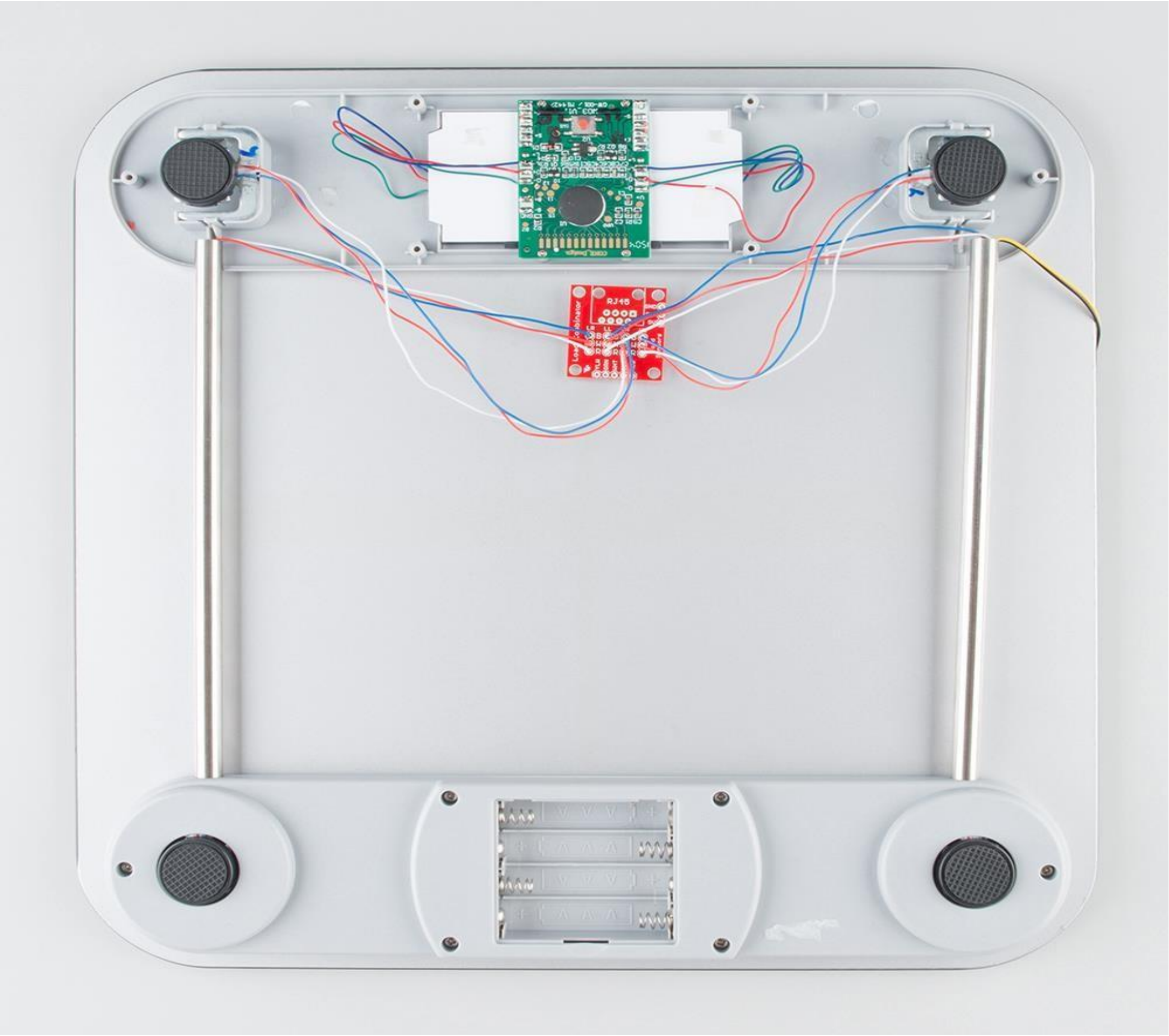
Depending on the type of load cell you are using, the configuration of how it should be hooked up to plates or surfaces will change. Below are a few different types of setups.



Bar load cell between a two plate configuration



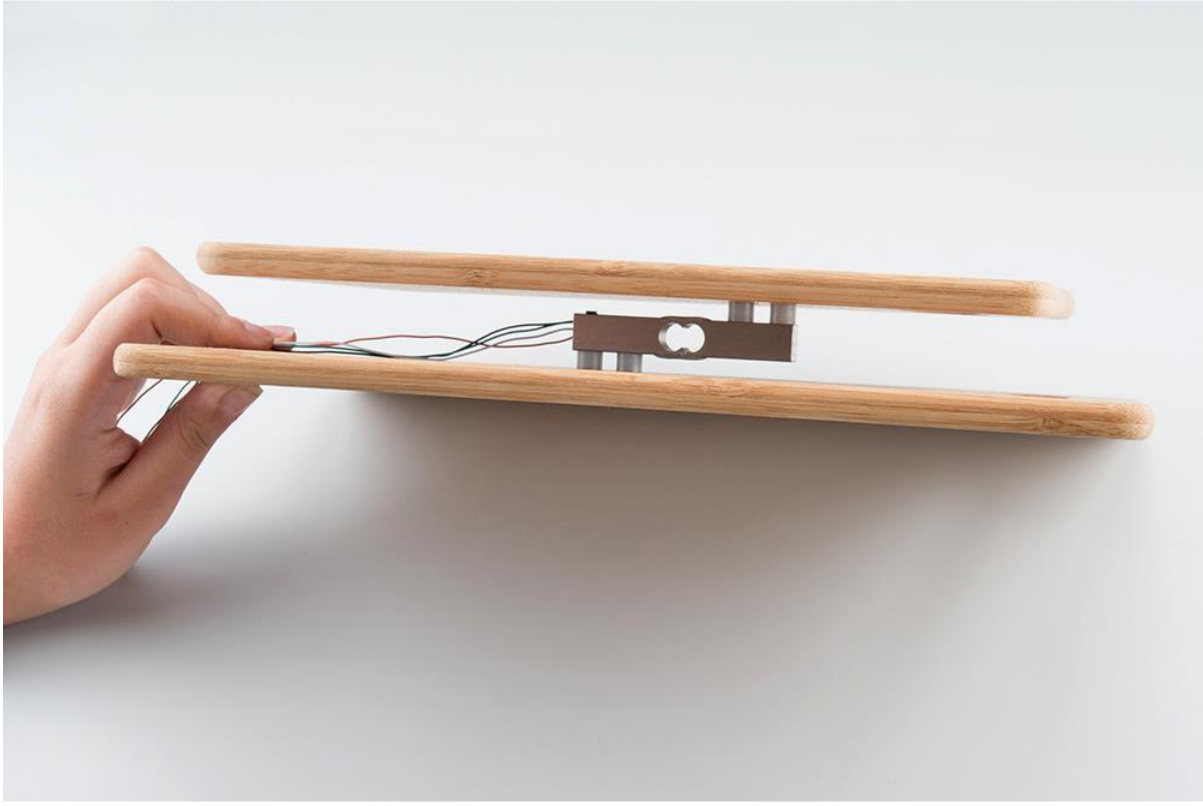
S load cell configuration



Possible four disc load cell configuration in something like a bathroom scale

Bar strain gauge based load cells

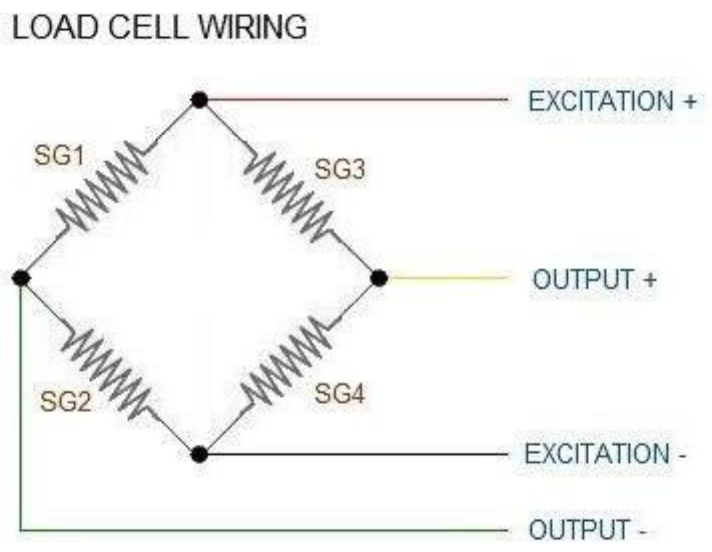
Usually with larger, non-push button bar load cells you will want to hook up the load cell between two plates in a “Z” shape, with fitting screws and spacers so that the strain can be correctly measured as shown below:



Note that only one side of the load cell is screwed into each board. This provides a moment of force, or torque, on the strain gauge rather than just compression force, which is easier to measure and much more accurate.

For smaller, push-button or disc load cells, you will want to make sure to screw in the disc to a bottom plate (or surface you are measuring force against), and center the beam, plate, or whatever else you are wishing to measure the force of onto the "button" on the top. Usually another plate with a hole is used to make sure whatever you are measuring is hitting the same spot on the load cell each time, but it is not necessary. Make sure to read the datasheet for the load cell you are using and get the correct screws to fit into it. • Note: If you are hooking together four of the SparkFun Load Sensors using the Combinatorboard, you should position the four load sensors equidistant from each other, just like the bathroom scales shown in this tutorial. Load cell measurements can be off by +/- 5% due to a range of things including temperature, creep, vibration, drift, and other electrical and mechanical interferences. Before you install your scale, take a moment and design your system to allow for easy calibration or be able to adjust the code parameters to account for these variations.

The HX711 Load Cell Amplifier accepts five wires from the load cell. These pins are labeled with colors; RED, BLK, WHT, GRN, and YLW. These colors correspond to the conventional color coding of load cells, where red, black, green and white wires come from the strain gauge on the load cell and yellow is an optional ground wire that is not hooked up to the strain gauge but is there to ground any small outside EMI (electromagnetic interference). Sometimes instead of a yellow wire there is a larger black wire, foil, or loose wires to shield the signal wires to lessen EMI

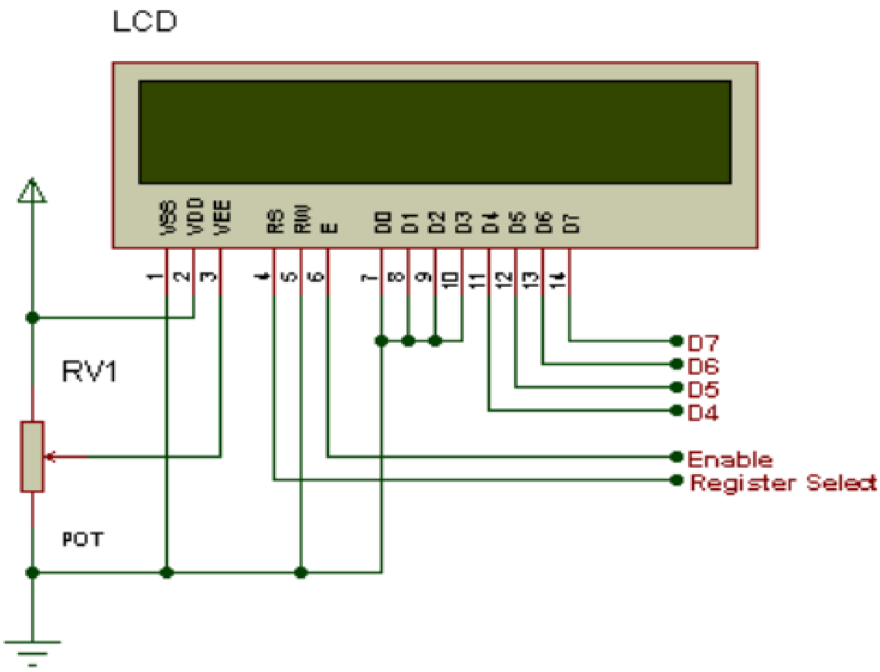


Four strain gauges (SG1 through 4) hooked up in a wheatstone bridge formation

3. LCD display:

LCD Background: One of the most common devices attached to a micro controller is an LCD display. Some of the most common LCD's connected to the many microcontrollers are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively. Basic 16x 2 Characters LCD

Figure 1: LCD Pin diagram



Pin description:

Pin No.	Name	Description
Pin no. 1	VSS	Power supply (GND)
Pin no. 2	VCC	Power supply (+5V)
Pin no. 3	VEE	Contrast adjust
Pin no. 4	RS	0 = Instruction input
D		1=Data input
Pin no. 5	R/W	0 = Write to LCD module
1		1 = Read from LCD module
Pin no. 6	EN	Enable signal
Pin no. 7	D0	Data bus line 0 (LSB)
Pin no. 8	D1	Data bus line 1
Pin no. 9	D2	Data bus line 2

Pin no. 10	D3	Data bus line 3
Pin no. 11	D4	Data bus line 4
Pin no. 12	D5	Data bus line 5
Pin no. 13	D6	Data bus line 6 27
Pin no. 14	D7	Data bus line 7 (MSB)

The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

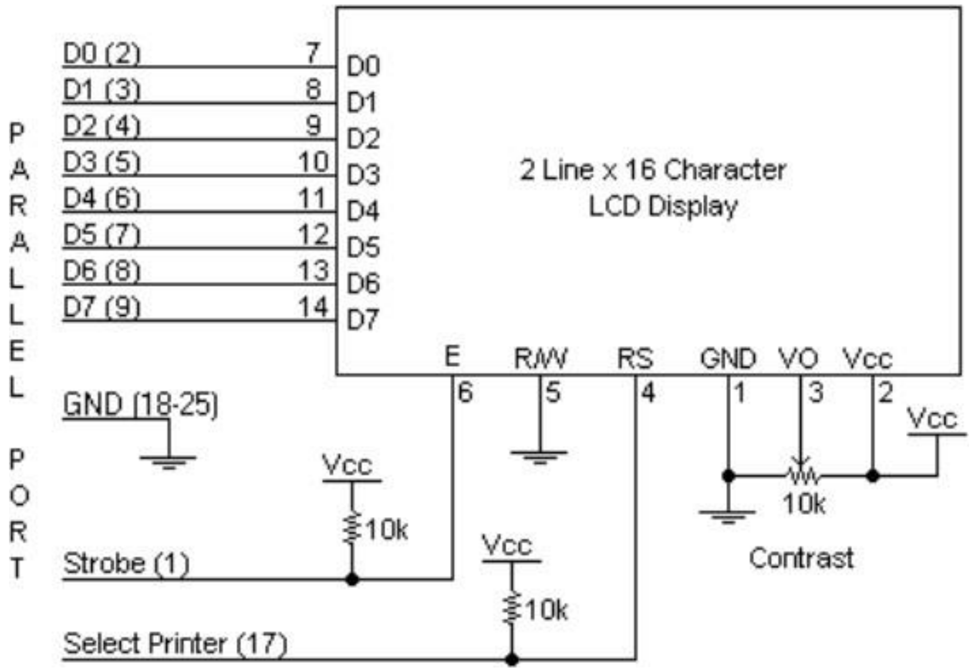
The three control lines are referred to as EN, RS, and RW.

The EN line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The RS line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands-- so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Schematic:

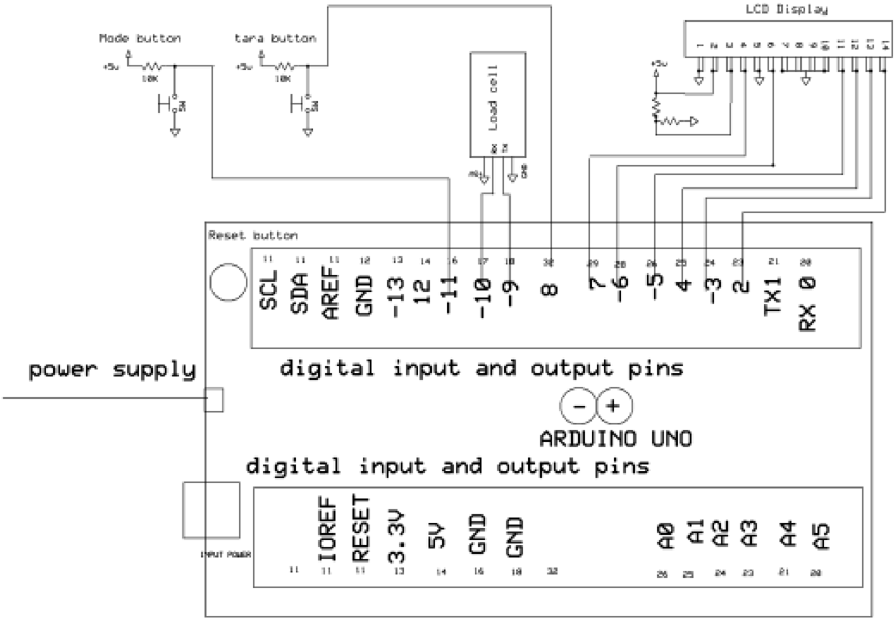


Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. We can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if we have trouble with the circuit working properly.

Diagram :



schematic diagram of STRAIN GAUGE

The above schematic diagram of STRAIN GAUGE explains the interfacing section of each component with microcontroller.

CODE :

```
#include <Q2HX711.h>
#include <Wire.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);



//Pins
const byte hx711_data_pin = 9;    //Data pin from HX711
const byte hx711_clock_pin = 10;   //Clock pin from HX711
```

```
int tara_button = 8;            //Tara button
int mode_button = 11;           //Mode change button


void setup() {
  lcd.begin(16,2);
  Serial.begin(9600);            // prepare serial port
  PCICR |= (1 << PCIE0);         //enable PCMSK0 scan
  PCMSK0 |= (1 << PCINT0);       //Set pin D8 trigger an interrupt on
state change.
  PCMSK0 |= (1 << PCINT3);       //Set pin D10 trigger an interrupt on
state change.
  pinMode(tara_button, INPUT_PULLUP);
  pinMode(mode_button, INPUT_PULLUP);


  delay(1000);                   // allow load cell and hx711 to settle


  // tare procedure
  for (int ii=0;ii<int(avg_size);ii++){
    delay(10);
    x0+=hx711.read();
  }
  x0/=long(avg_size);
  Serial.println("Add Calibrated Mass");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Add Calibrated ");
```

```
  lcd.setCursor(0,1);
  lcd.print("      Mass      ");
  // calibration procedure (mass should be added equal to y1)
  int ii = 1;
  while(true){
    if (hx711.read()<x0+10000)
    {
      //do nothing...
    }
    else
    {
      ii++;
      delay(2000);
      for (int jj=0;jj<int(avg_size);jj++){
        x1+=hx711.read();
      }
      x1/=long(avg_size);
      break;
    }
  }
  Serial.println("Calibration Complete");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  Calibration   ");
  lcd.setCursor(0,1);
  lcd.print("    Complete    ");
}

void loop() {
```

```
// averaging reading
long reading = 0;
for (int jj=0;jj<int(avg_size);jj++)
{
  reading+=hx711.read();
}
reading/=long(avg_size);



// calculating mass based on calibration and linear fit
float ratio_1 = (float) (reading-x0);
float ratio_2 = (float) (x1-x0);
float ratio = ratio_1/ratio_2;
float mass = y1*ratio;
mass=mass/15;


}
if(tara_pushed)
{
  tara = mass;
  tara_pushed = false;
  Serial.print("TARA");
  Serial.print(".");
  lcd.setCursor(0,0);
  lcd.print("      TARA      ");
  lcd.setCursor(0,1);
  lcd.print("      .         ");
  delay(300);
```

```
  Serial.print(".");
  lcd.setCursor(0,0);
  lcd.print("      TARA      ");
  lcd.setCursor(0,1);
  lcd.print("      ..        ");
  delay(300);
  Serial.println(".");
  lcd.setCursor(0,0);
  lcd.print("      TARA      ");
  lcd.setCursor(0,1);
  lcd.print("      ...       ");
  delay(300);
}
if(mode_pushed)
{
  mode = mode + 1;
  mode_pushed = false;
  if(mode > 2)
  {
    mode = 0;
  }
}

if(mode == 0)
{
  Serial.print(mass - tara);
  Serial.println(" g");
  lcd.clear();
  lcd.setCursor(0,0);
```

```
    lcd.print("     SCALE!     ");
    lcd.setCursor(0,1);
    lcd.print(mass - tara);
    lcd.print(" g");
  }
  else if(mode == 1)
  {
    Serial.print(mass - tara);
    Serial.println(" ml");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("     SCALE!     ");
    lcd.setCursor(0,1);
    lcd.print(mass - tara);
    lcd.print(" ml");
  }
  else
  {
    Serial.print((mass - tara)*oz_conversion);
    Serial.println(" oz");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("     SCALE!     ");
    lcd.setCursor(0,1);
    lcd.print((mass - tara)*oz_conversion);
    lcd.print(" oz");
  }

}//End of void loop
```

```
//interruption to detect buttons
ISR(PCINT0_vect)
{
  if (!(PINB & B00000001))
  {
    tara_pushed = true;           //Tara button was pushed
  }

  if (!(PINB & B00001000))
  {
    mode_pushed = true;           //Mode button was pushed
  }
}
```

III. ADVANTAGES DISADVANTAGES

ADVANTAGES:

1. Automatic weight measuring system.
2. Visible alerts using LCD display.
3. Fast response.
4. Wide detection range.
5. Efficient and low-cost design.

DISADVANTAGES

• Interfacing load cell to the Arduino is very critical.

APPLICATIONS

• Homes.

• Industries.

• Shops.

IV. RESULTS

Result:

Load Cell and Strain Gauge: The load cell contains one or more strain gauges, which are bonded to a flexible backing material. When force or weight is applied to the load cell, it deforms, causing a change in the electrical resistance of the strain gauges.

HX711 Amplifier:

The HX711 is a precision 24-bit analog-to-digital converter (ADC) specifically designed for weighing scales and load cells. It amplifies and converts the small electrical signals from the strain gauges into digital signals that can be read by the Arduino. 53 Arduino Microcontroller: The Arduino serves as the central control unit of the system. It communicates with the HX711 amplifier via a digital interface (e.g., SPI or GPIO pins) to receive the digital weight data.

LCD Display:

An LCD (liquid crystal display) screen is used to provide visual feedback to the user. It displays the measured weight, as well as any user prompts or messages.

Working Process:

When the load cell is subjected to force or weight, the strain gauges deform, causing a change in electrical resistance. The HX711 amplifier detects this change in resistance and converts it into a digital signal. The Arduino reads this digital signal from the HX711 and processes it to calculate the weight or force applied to the load cell. The Arduino then displays the measured weight on the LCD screen. The two buttons can be used to perform additional functions, such as resetting the scale to zero (taring) or selecting different units of measurement. The process repeats continuously, allowing the system to continuously measure and display the weight in real-time. Overall, this system provides a simple and effective way to measure weight using a strain gaugebased load cell, Arduino microcontroller, HX711 amplifier, LCD display, and two buttons for user interaction. It can be used in various applications, including kitchen scales, industrial weighing systems, and DIY projects.

Conclusion:

Integrating features of all the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC’s with the help of growing technology, the project has been successfully implemented. Thus, the project has been successfully designed and tested.

V. REFERENCES

[1] M. M. Hilal and S. H Mohamed, “A Strain Gauge Based System For Measuring Dynamic Loading on a Rotating Shaft,” International Journal Of Mechanics, Issue 1, Vol.5, 2011.

[2] S. Middelhoek, S. A. Audet, “Silicon Sensors,”Delft University of Technology, Department of Electrical Engineering. (Delft: Delft University Press) Lecture et 05-31, 1994.

[3] S.F Richard and E.B. Donald, “Theory and Design for Mechanical Measurements,” 5th ed. John Wiley & Sons, Inc. Chapter 11: pp. 466, 2011.

[4] "MIT - Department of Civil & Environmental Engineering Department of Civil Environmental Engineering, MIT".Cee.mit.edu. 2011-05-01. Retrieved 2011-09-29.

[5] H. K arl, “An Introduction to Measurements using strain gauges,” Germany: Druckerei Drach Press, Alsbach. Pp.1-32, 1987.

[6] Youtube video link https://youtu.be/LRd3W_p8PJ4?feature=shared.