Arduino Model Rocket Motor Test Stand

Nick Cinquino and Len Johnson, 10/2015

Combining an Arduino Uno microcontroller, an LCD keypad shield, an HX711 breakout board, a 10Kg loadcell, and a custom made engine mount yields a sensitive and accurate test stand or dynamometer for testing the thrust in Newtons of model rocket motors. The circuit is also usable as a stand-alone gram scale.

Functionally, a model rocket motor is affixed to a loadcell, which generates a very small analog voltage difference, proportional to the force applied. The small voltage is applied to an HX711 breakout board, which amplifies the voltage and converts to serial digital data. The serial data is received by the Arduino microcontroller, which applies a calibration factor so that output is shown as grams of force. When a motor is affixed to the loadcell, and the igniter is placed and wired, AND the momentary pushbutton is activated, a 10-second warning time begins, with flashing of a bright yellow LED. When the 10 seconds is up, the yellow LED turns off, a red LED turns on, and a relay is closed allowing the full 12V from a gel-cell battery to be applied to the igniter. Immediately afterwards, the Arduino logs the time, and begins acquiring force data at 80 samples per second. After 800 readings are acquired, the relay shuts down, acquisition stops and the stop time is logged. In the meantime, the Arduino is also displaying gram-force on the LCD display, and showing the status of the acquisition run.

Beginning with the loadcell, they are a beam with strain gages attached, in a Wheatstone Bridge arrangement. Loadcells can be purchased new from several sources including Sparkfun, or Aliexpress.com, or may be liberated from old, nonfunctional industrial scales.
Fig.1 shows a collection of loadcells from old scales…the electronics break down long before the loadcell does! The large one at center is rated to 2400Lb!

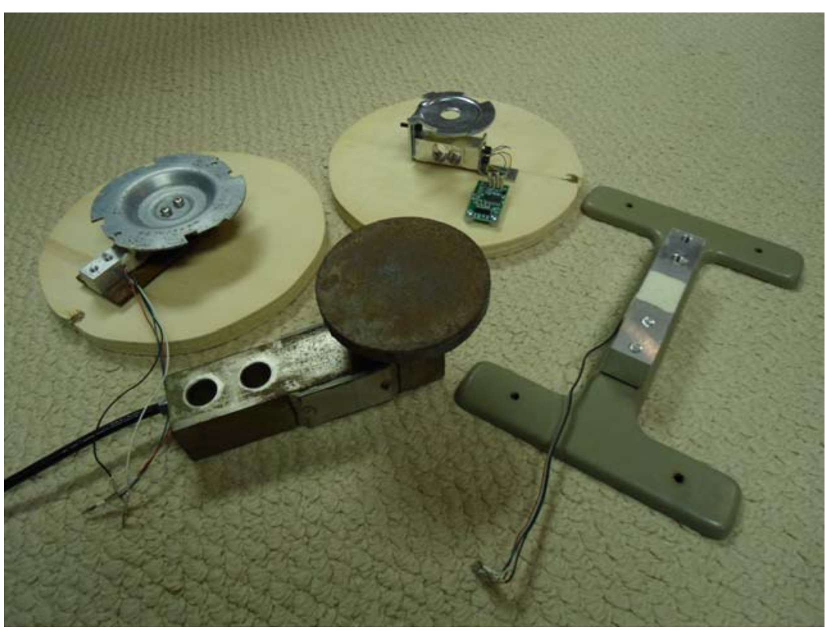

Fig.1

Next is the HX711 breakout board. The one in use is the YL-112 module from Aliexpress.com. The cable between the HX711 board and Arduino should be at least 4’. One easy hack is necessary: to change the sample rate from 10 samples per second to 80 samples per second, pin 15 needs to be desoldered from ground, and attached to +5 instead. This is done by using a fine tipped iron, and using a needle to pull up the pin the instant the solder melts. After the pin is bent up, run a short jumper from the pin to any +5V spot on the board. An even easier option is to use the Sparkfun HX711 board which is configurable for either 10 or 80 sa/sec. Fig.2 shows the HX711 hack:

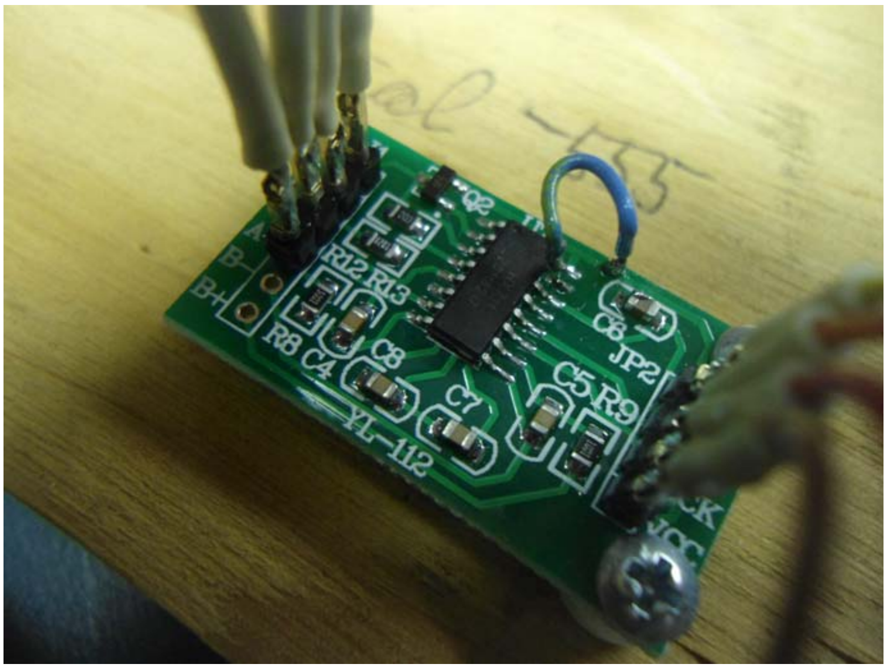

Fig.2: Desolder pin 15 from ground, jumper to +5V.

Arduino Uno/LCD Keypad shield: solder headers to the top row of the LCD shield and to the bottom 2 rows. Pin identifications on the top row of the shield, from left to right are D13, D12, D11, D3, D2, D1, D0 to the Arduino. Plug the LCD shield into the Arduino and add the remainder of the circuit <u>except for the 12V/igniter section</u>, fig.3:

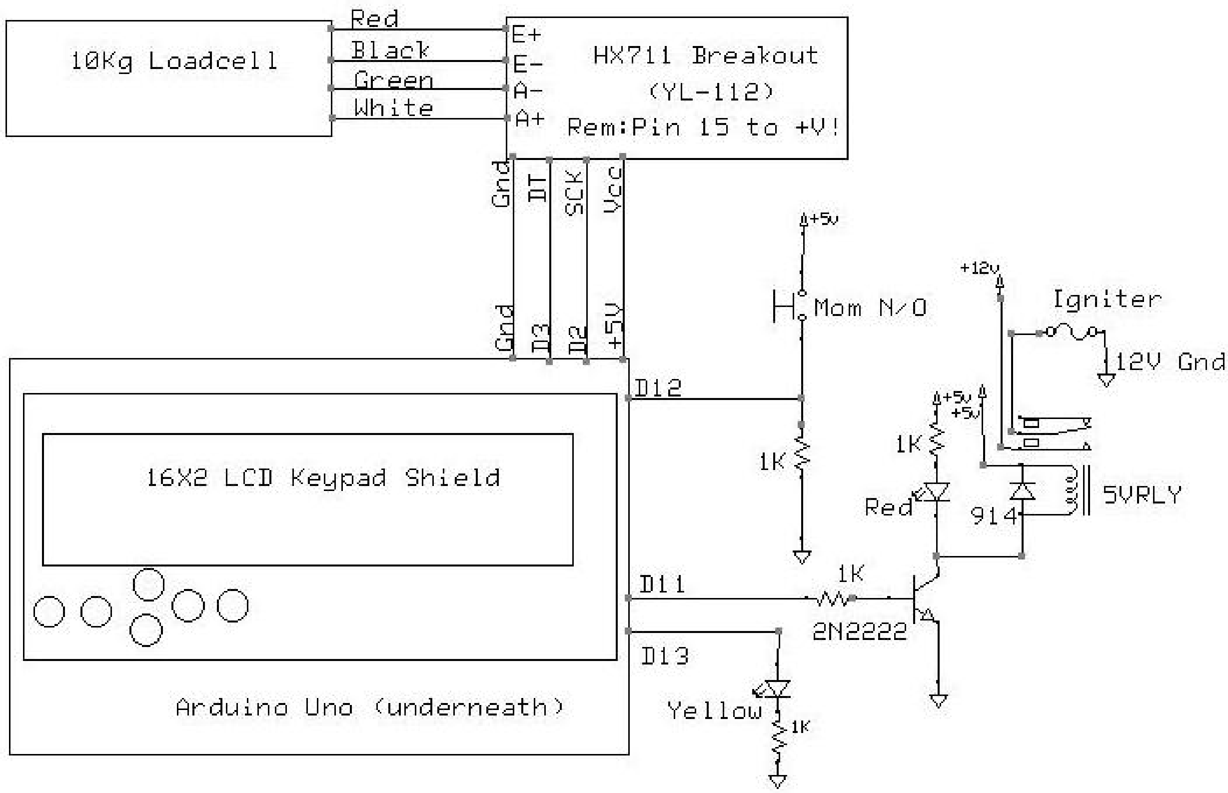

Fig.3 Circuit schematic

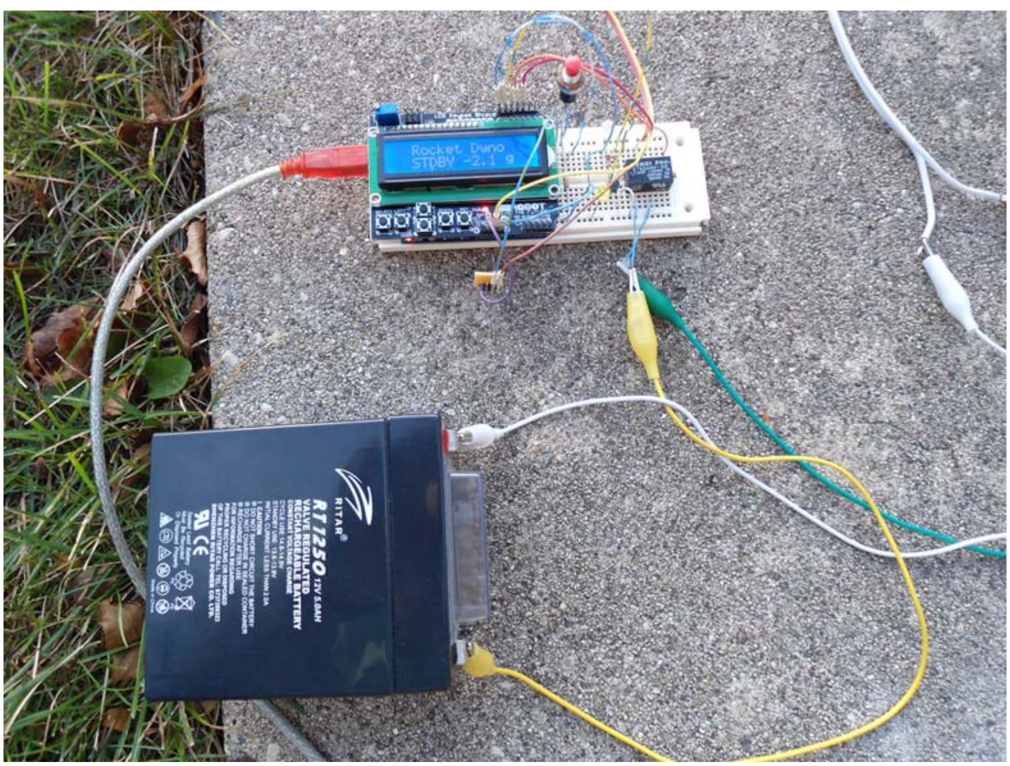

Fig.4 Arduino with LCD keypad shield and switch/LED's/relay/12V gel-cell.

Programming/Sketch: The HX711 library is needed; download from Github (see references) and add to your library folder. Then copy/paste the following sketch into your Arduino IDE window:

```
/*
 Model Rocket Engine Dynamometer - Nick Cinquino 9/1/15 - V.5 Original 10Kg loadcell
 Modified from Nathan Seidle, SparkFun Electronics, November 19th, 2014
 Output data in grams (g). There are 101.97 gram-force per Newton, for conversion.
 Your calibration factor may be very positive or very negative. It all depends on the
setup of your individual loadcell, and the direction the sensors deflect from zero state.
Need to experiment!
 Hack the HX711 board to set pin 15 high, for 80 samples per second.
 This example code uses bogde's excellent library: https://github.com/bogde/HX711
 bogde's library is released under a GNU GENERAL PUBLIC LICENSE
 Arduino pin 2 - HX711 Clock
 pin 3 - Serial Data In
 5V - VCC
 GND - GND
 Yellow LED on pin 13, igniter transistor/relay on pin 11, momentary pushbutton on pin
12.
 Most any pin on the Arduino Uno will be compatible with DOUT/CLK.
 The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power is fine.

 */

#include "HX711.h"
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,4,5,6,7);
const int buttonPin = 12;      // start sequence button
const int ledPin = 13;        //LED indicator and/or buzzer
const int igniterPin = 11;   //igniter transistor circuit
int buttonState = 0;
#define DOUT  3
#define CLK  2
HX711 scale(DOUT, CLK);
float calibration_factor = -560; //-560 works for my 10kg loadcell.


void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(igniterPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("HX711 Rocket Motor Dynamometer, V.5");
  Serial.println("Affix motor nozzle up. Place igniter in nozzle. Move away from test
stand.");
  Serial.println("Press start button to initialize ignition sequence.");
  lcd.begin(16, 2);
  lcd.clear();
```

```
  lcd.setCursor(0,0);
  lcd.print("  MODEL ROCKET");
  lcd.setCursor(0,1);
  lcd.print("  DYNAMOMETER");
  delay(2000);
  scale.set_scale();
  scale.tare(); //Reset the scale to 0
  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale.
Useful in permanent scale projects.
  Serial.println(zero_factor);
  Serial.println(" ");
}

void loop() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  Rocket Dyno");
  lcd.setCursor(0,1);
  lcd.print("  STDBY " );
  scale.set_scale(calibration_factor);
  lcd.print(scale.get_units(),1);
  lcd.print(" g");
  delay(500);
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("  Rocket Dyno");
    lcd.setCursor(0,1);
    lcd.print("  STAND CLEAR!");
    Serial.println("IGNITION SEQUENCE ACTIVATED!");

    for (int i=0; i <= 50; i++){
      digitalWrite(ledPin, HIGH);
      delay (100);
      digitalWrite (ledPin,LOW);
      delay (100);
    }
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("  Rocket Dyno");
    lcd.setCursor(1,1);
    lcd.print(" AQUIRING DATA");
    digitalWrite(igniterPin, HIGH);
    Serial.print("Start time, ms: ");
    Serial.print (millis());
    Serial.println(" ");
    Serial.println();
```

```
  for (int i=0; i <= 800; i++){        //800 samples at 80sa/sec = 10 seconds theoretical
    scale.set_scale(calibration_factor); //Adjust to the calibration factor
    Serial.print(scale.get_units(), 1);
    Serial.println();
  }
  Serial.println();
  Serial.print("Stop Time, ms: ");
  Serial.print(millis());
  digitalWrite (ledPin,LOW);
  digitalWrite (igniterPin,LOW);
  Serial.println();
 }
}
```

Note the line highlighted in yellow: this is the calibration factor that the Arduino uses to convert the signal from the loadcell/HX711 into grams. This value varies widely, even for 2 apparently identical loadcells! You just have to place a known weight in grams on the loadcell and try some values until you zero in on the best value that reads your calibration weight in grams on the LCD.

Loadcell/motor mount: This can be designed and built to suit; the general guidelines are, keep it light (no use in burning up dynamic range), and, importantly, nonflammable! Ours uses a thin sheet steel circular base (such as the cutout from a coffee can) bolted down to the loadcell, with a cemented motor tube with retaining clip attached to the center of the steel base. The motor tube is a half inch longer than the motor, with vent holes at the bottom so that on ejection, the motor moves forward against the retaining clip while ejection gases escape via the bottom vents. Fig.5 shows the loadcell with base:

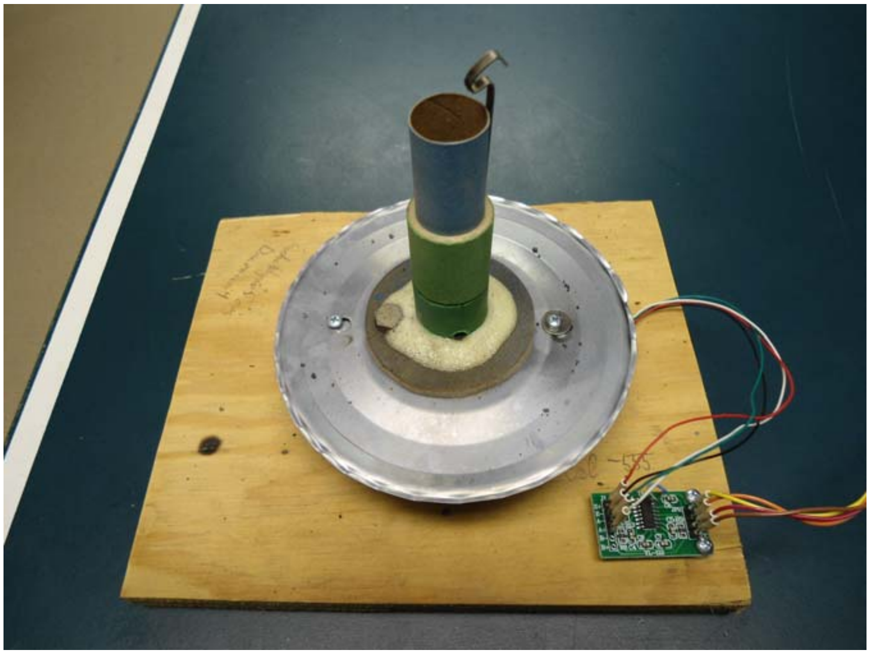

Fig.5 Motor mount on loadcell

When all systems are ready and connected and the Arduino is running the program, open the Arduino serial window on your laptop. Observe the LCD information. Push the momentary pushbutton. The yellow LED should start blinking, for 10 seconds. Then the red LED turns on and you should hear the relay click. Data should start streaming rapidly on the screen, and stop after 10 seconds. Relay clicks off. Take note of the start time and stop time on-screen. Subtract stop from start time to get delta T. Divide by 800 to get milliseconds per sample (should be around 11), useful for the chart X axis. Convert to samples per second (1/X) which should be between 80-90sa/sec. Copy only the weight data column, and paste into Excel. Make a line chart of this test run.

If everything above went well, you're ready for a burn test! Make sure the area all around is free of anything flammable. Set up your laptop, with a 6' USB cable to the Arduino section, then the 4' cable to the HX711/loadcell section. <u>Leave the 12V battery unconnected until the last step.</u> Open the Arduino serial window. Check status. Insert motor into your loadcell/motor mount, and add igniter. Connect igniter to igniter leads (battery is still unconnected). Move back to the Arduino section, check status on LCD and LED's. If everything is go, then connect the 12V battery, and push the momentary button, move away from the area.

When the run completes, as in the test, note the start and stop times and copy/paste the gram data into Excel. If desired, add a new column converted into Newtons (grams / 101.97 = Newtons). Generate an exact time scale from the stop – start calculation. Following are results from 2 runs of (old) C6-5 motors:

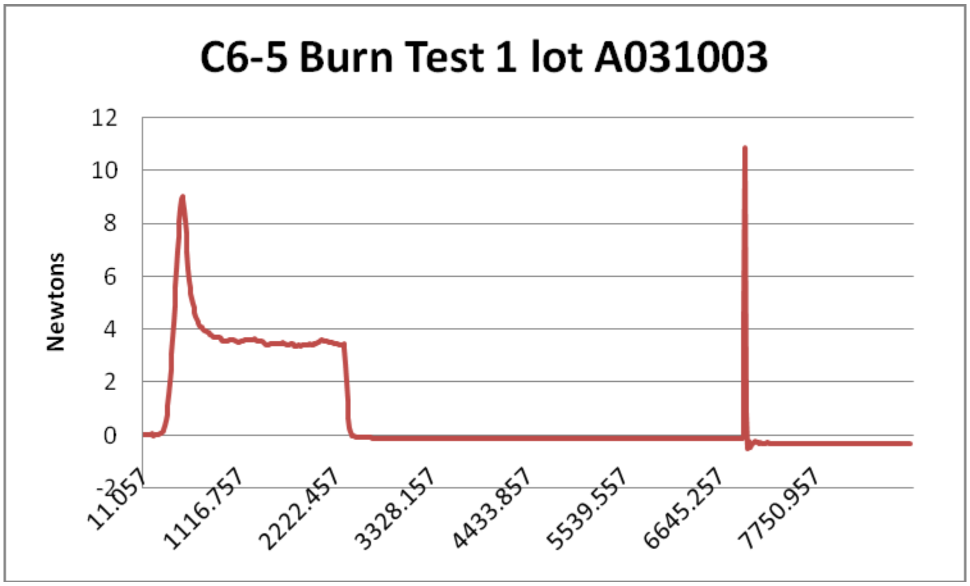

Fig.6 Actual test data

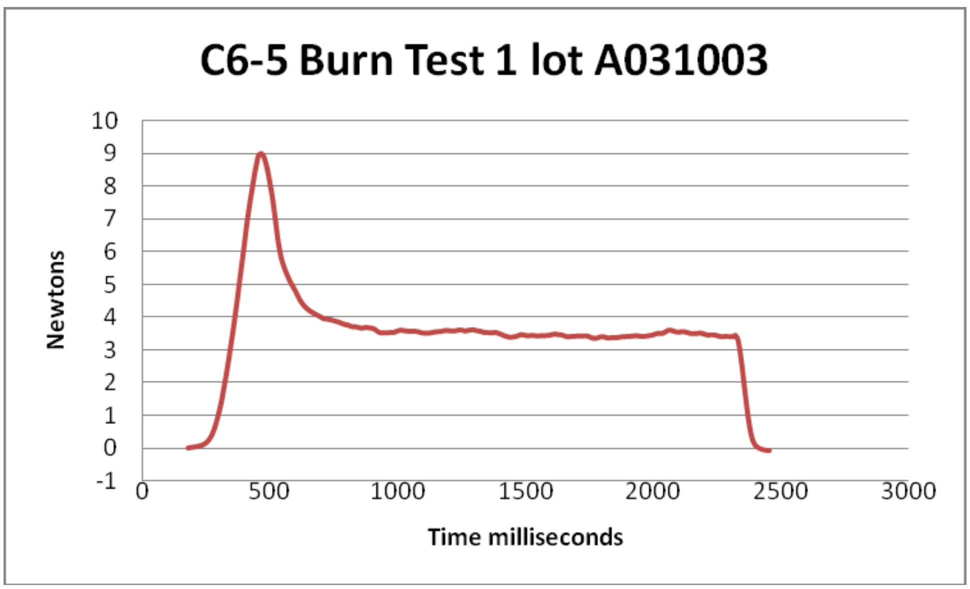

Fig.7 Closeup of burn section

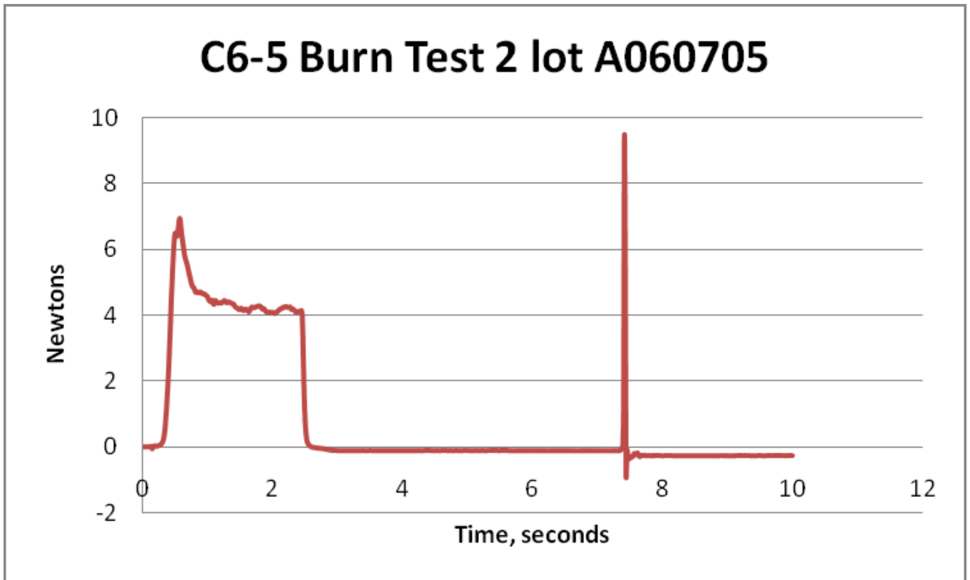

Fig.8 2nd run. Note lower peak thrust but higher sustained. Note oscillation.

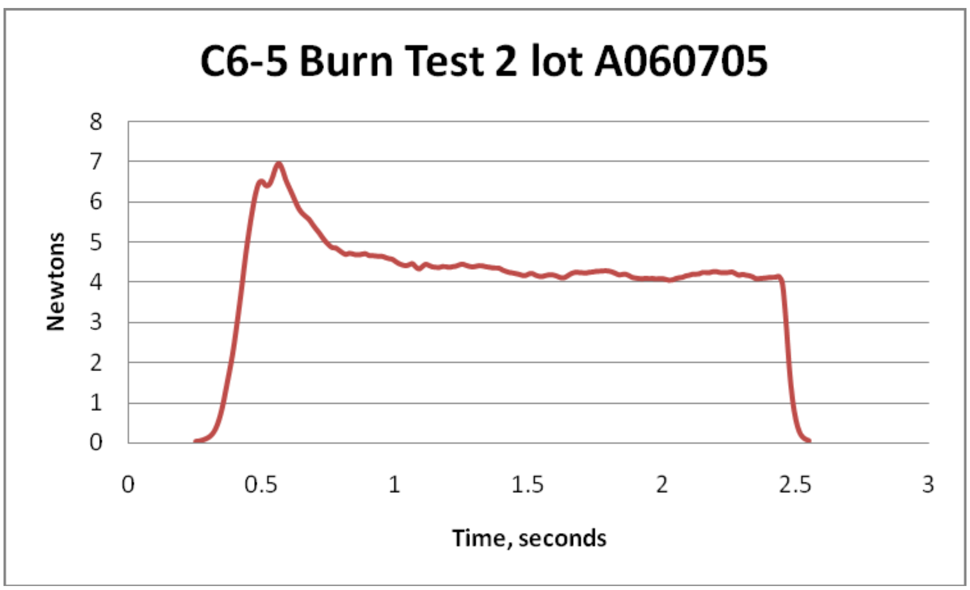

Fig.9 Closeup of burn time, test 2. Kind of a spooky looking peak thrust!

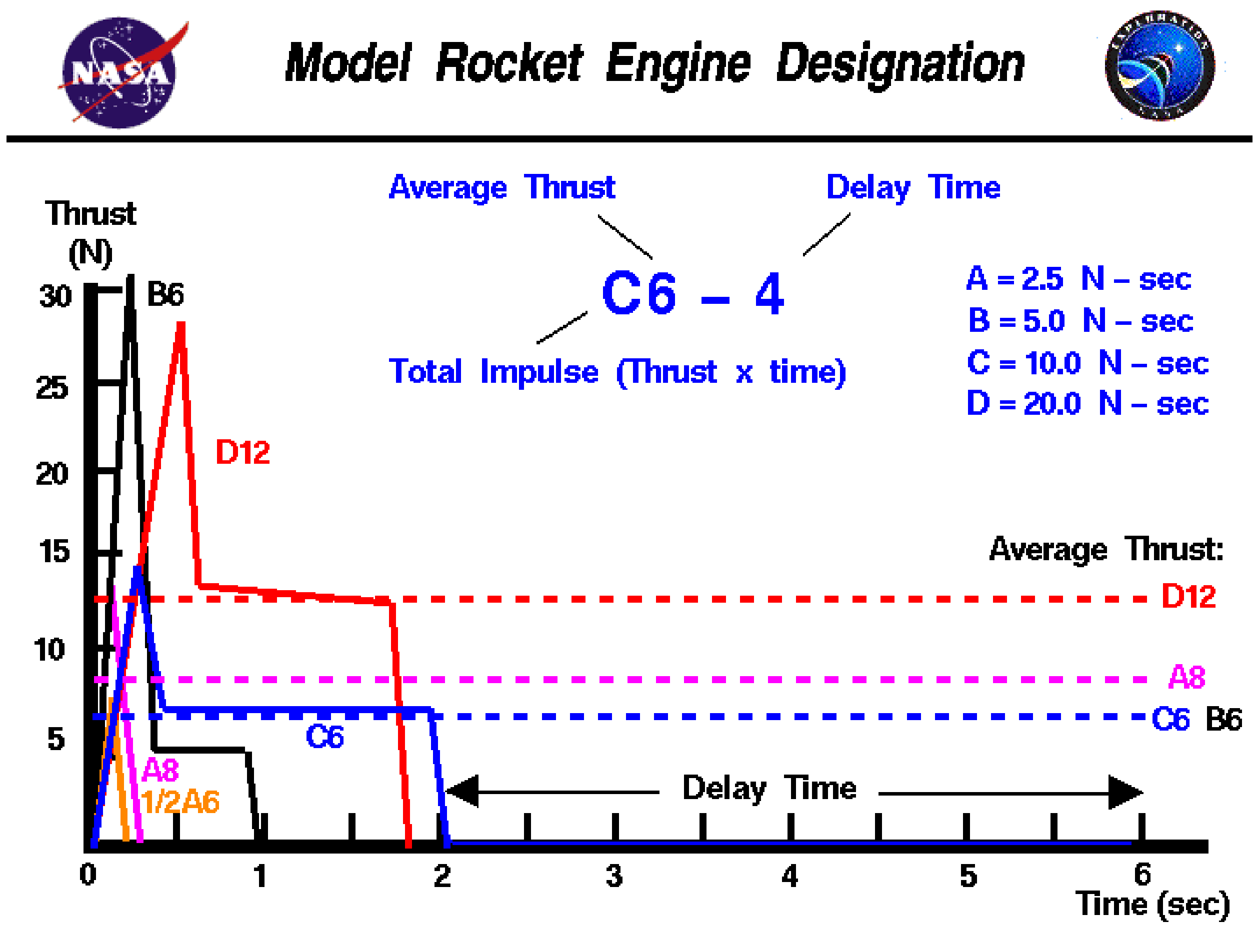

Fig.10 Anticipated thrust curve of a C6 motor; higher than what we found from old motors.
Note that the test motors burned slightly longer.

References:
Github source of HX711 library:
https://**github**.com/bogde/**HX711**

HX711 board:
https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide
http://www.aliexpress.com/item/For-Arduino-Dual-channel-HX711-Weighing-Pressure-Sensor-24-bit-Precision-A-D-Module/32296409056.html

Loadcell sources:
www.**aliexpress**.com/ (search loadcell on the site)
https://www.sparkfun.com/products/13329