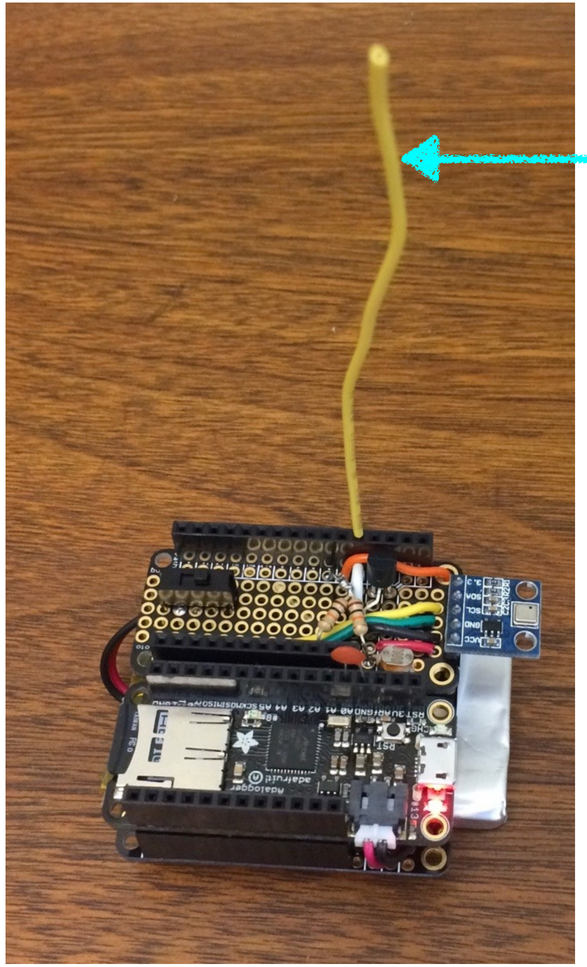
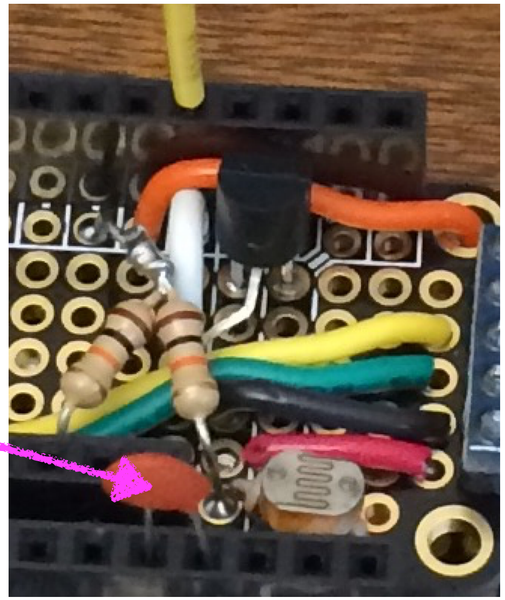
Defining Orders of Uncertainty in a Single Measured Quantity

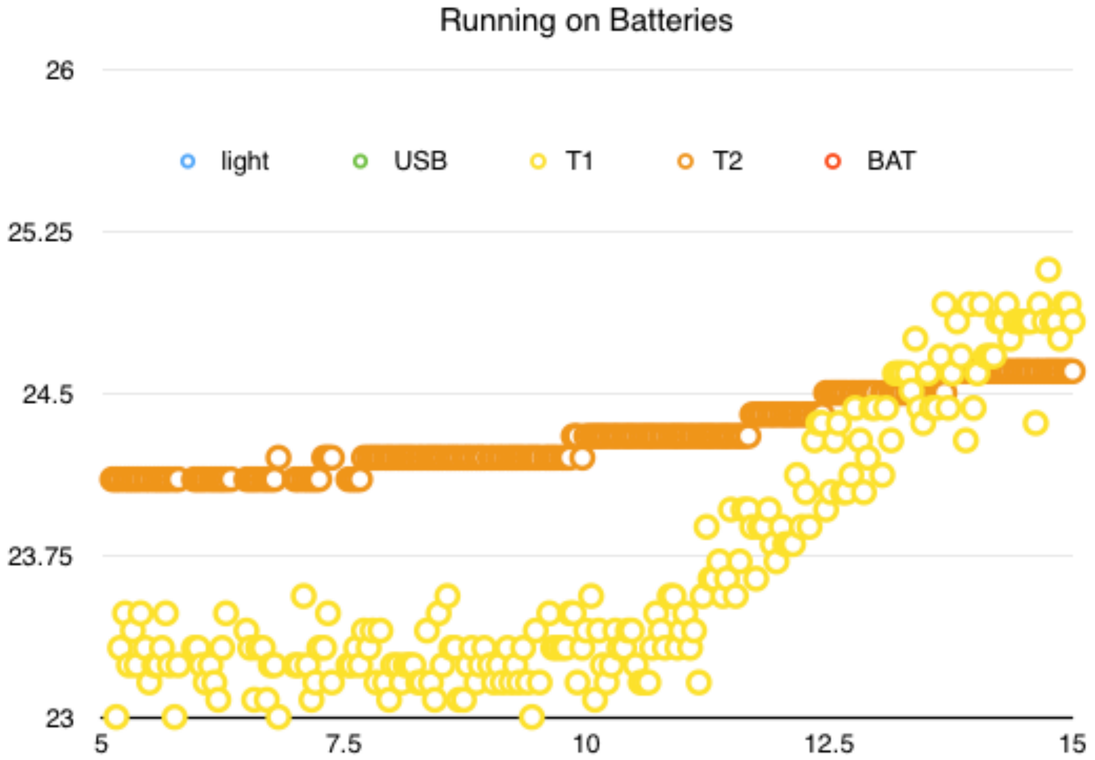
Rick Sellens



RF from room

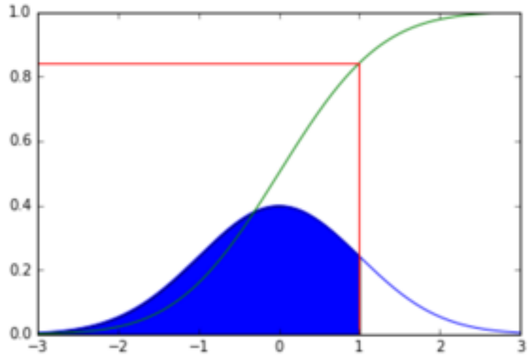
0.1 uF Output Capacitor





What is actual T(t)?

Assume a Gaussian Normal Distribution



- for everything random unless you have evidence for another distribution
- you will get something close to it combining multiple sources anyway
- 95% of the variation will lie within +/- 2 standard deviations of the mean
- You will reduce the variation by the square root of the number of samples when you average

Define Uncertainty based on 95% probability range

+/- 2 standard deviations

Assuming a Gaussian Normal Distribution

Watch out for “Standard Error” or “Typical”, probably 1 SD

Zero Order Uncertainty

- plus or minus one unit of **instrument resolution**
  - one analog to digital conversion step
  - one step in the least significant digit of a digital display
  - half a gradation on an analog dial
- not enough to know what is going on

Instrument Uncertainty

- uncertainty due to noise and calibration issues
- often approximated by data sheet specifications
- can be reduced by calibration with a more accurate standard
- can be reduced by noise reduction through averaging

Design Stage Uncertainty

- before we actually assemble any hardware, so only **first order** — more effects, higher order

- estimates need to come from data sheets

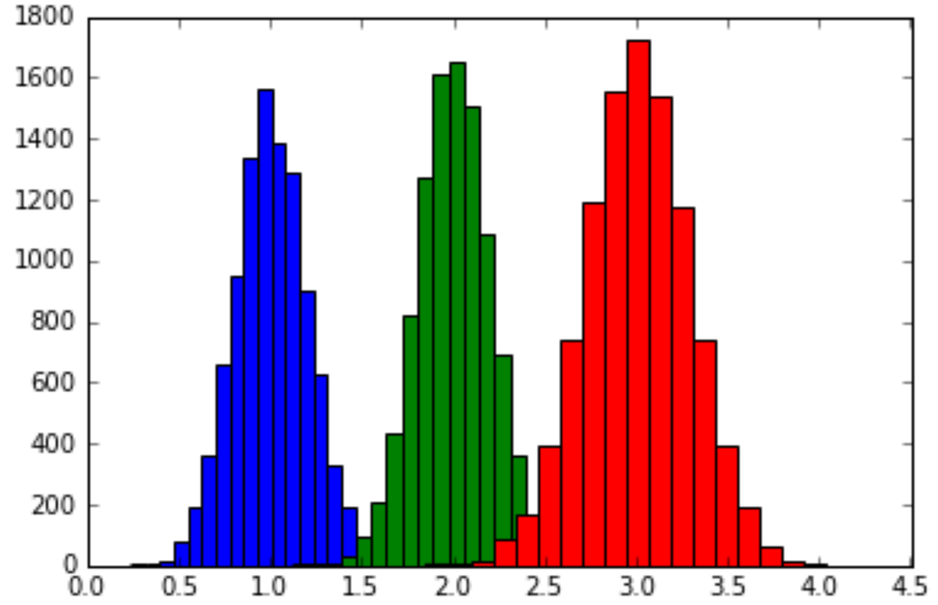
- combine resolution and instrument uncertainty by root mean square formula

- We will always use 95% probability level and 2 standard deviations to avoid confusion

$$u_d = \sqrt{u_0^2 + u_c^2} \quad (P\%) \tag{5.3}$$

Two Gaussians -- how do they combine?

In [31]:
```
%matplotlib inline
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
Vbar = 1
sigmaV = .2
Ibar = 2
sigmaI = 0.2
V = Vbar + sigmaV * np.random.randn(10000)
I = Ibar + sigmaI * np.random.randn(10000)
VpI = V + I
n,bins,patches = plt.hist(V,20)
n,bins,patches = plt.hist(I,20)
n,bins,patches = plt.hist(VpI,20)
```



Potential Question

- What is the design stage uncertainty for a single temperature measurement made with a TMP36 transducer and recorded by an Arduino UNO?
- Would it be lower if we planned to average over multiple measurements? Why or why not?
- I would probably have to feed you critical information about TMP36 and UNO

TMP36 and Arduino UNO Resolution

- TMP36 is analog, so no resolution error
- UNO has 10 bit A/D so 1024 steps over the 1.1 volt range = 1.07 millivolt steps
- TMP36 sensitivity is 10 mv/C, so 1 mV —> 0.1 C
- Zero order, or resolution uncertainty
  u_0= 1.07 / 10 = 0.107 C
- If we used the normal 5 volt range we would get lower resolution and u_0 = 0.488 C

TMP36 Accuracy

The TMP35/TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the –40°C to +125°C temperature range.

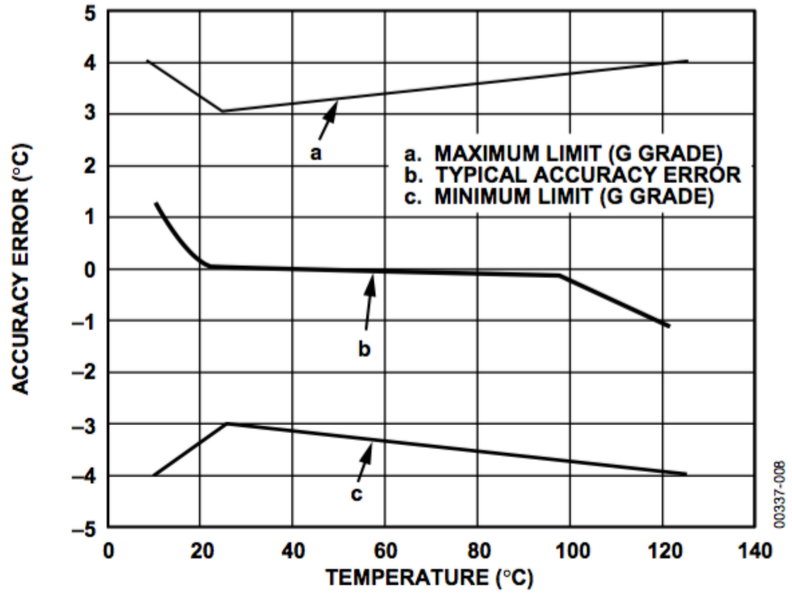


Figure 7. Accuracy Error vs. Temperature

- "Typical" makes me think 1 SD = 1 C
- "Max" makes me think 3 SD = 3 C
- Estimate u_C = 2 C @ 95%

TMP36 and Arduino UNO Design Stage Uncertainty

- Combine Resolution and Instrument Accuracy

$$u_d = \sqrt{u_0^2 + u_c^2} \quad (P\%) \tag{5.3}$$

- (0.107**2 + 2.0**2)**0.5 = 2.00

- The uncertainty is hugely dominated by the accuracy of the TMP36

- Averaging won't change it, because we haven't even considered noise yet

TMP36 and Arduino Noise

- We measured noise in the temperature signal from a typical setup to have a standard deviation of 0.14 C
  - we don’t know if the noise came from TMP36 or Arduino
- 2 standard deviations: u_n = 0.28 C
- Combine with design stage that ignored noise
- (2.0**2 + 0.28**2)**0.5 = 2.02 C for a single measurement
- TMP36 Accuracy still dominates

MICROCHIP

MCP9808

±0.5°C Maximum Accuracy Digital Temperature Sensor

Features

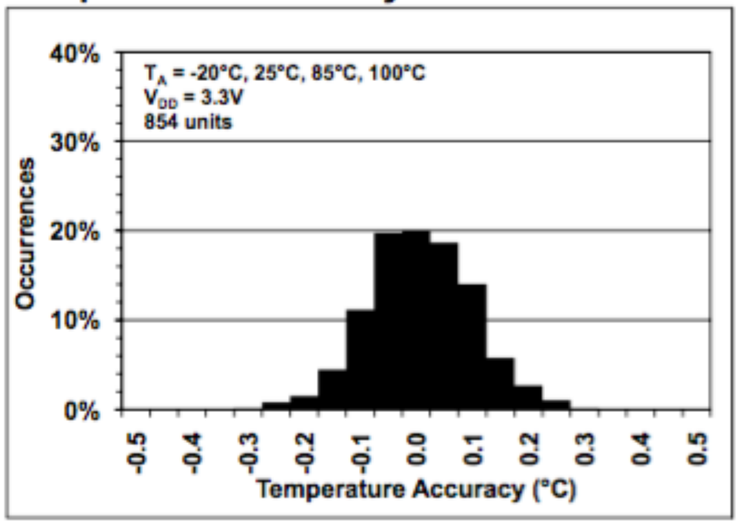
- Accuracy:
  - ±0.25 (typical) from -40°C to +125°C
  - ±0.5°C (maximum) from -20°C to 100°C
  - ±1°C (maximum) from -40°C to +125°C
- User-Selectable Measurement Resolution:
  - +0.5°C, +0.25°C, +0.125°C, +0.0625°C

Description

Microchip Technology Inc.'s MCP9808 digital temperature sensor converts temperatures between -20°C and +100°C to a digital word with ±0.25°C/±0.5°C (typical/maximum) accuracy.

The MCP9808 comes with user-programmable registers that provide flexibility for temperature sensing

Temperature Accuracy



What do you think is a good value to use for uncertainty? +/-0.25? +/- 0.5? +/-1.0? or?

MICROCHIP

MCP9808

±0.5°C Maximum Accuracy Digital Temperature Sensor

Features

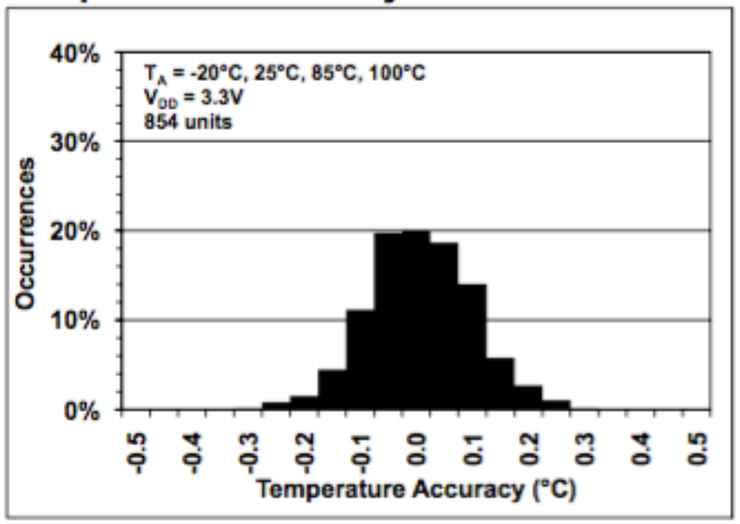
- Accuracy:
  - ±0.25 (typical) from -40°C to +125°C
  - ±0.5°C (maximum) from -20°C to 100°C
  - ±1°C (maximum) from -40°C to +125°C
- User-Selectable Measurement Resolution:
  - +0.5°C, +0.25°C, +0.125°C, +0.0625°C

Description

Microchip Technology Inc.'s MCP9808 digital temperature sensor converts temperatures between -20°C and +100°C to a digital word with ±0.25°C/±0.5°C (typical/maximum) accuracy.

The MCP9808 comes with user-programmable registers that provide flexibility for temperature sensing

Temperature Accuracy



Looks like an uncertainty around 0.2 C based on their stats