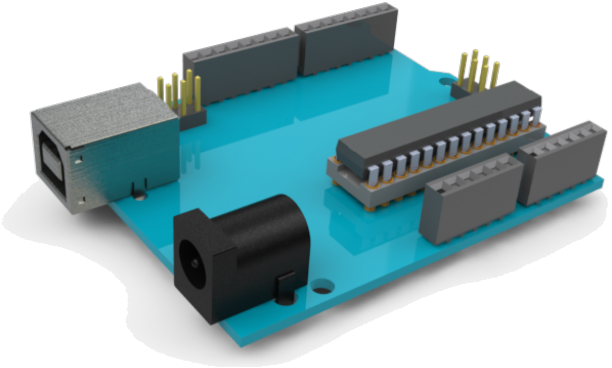
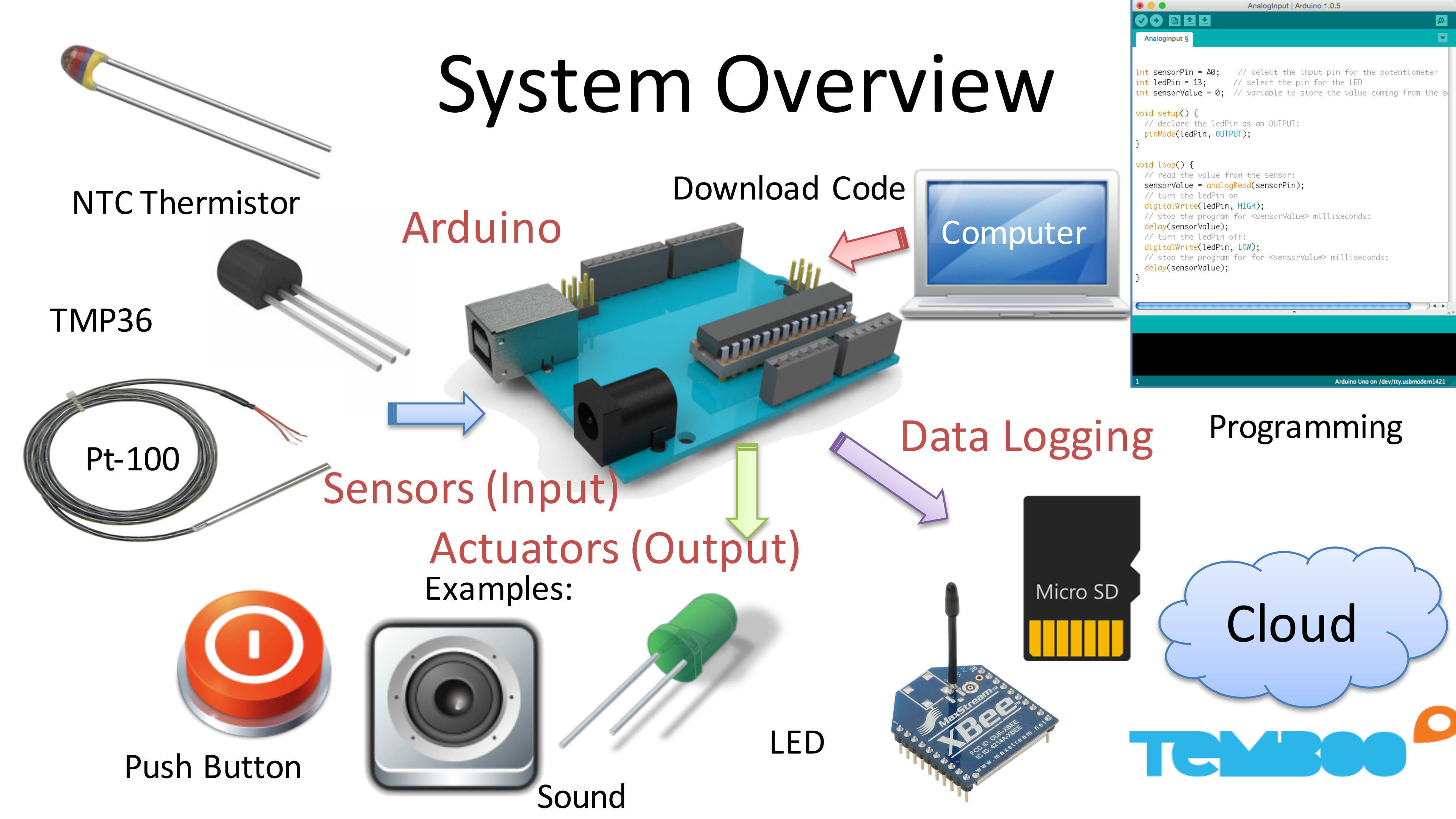
Sensors and Actuators with Arduino



Hans-Petter Halvorsen, M.Sc.

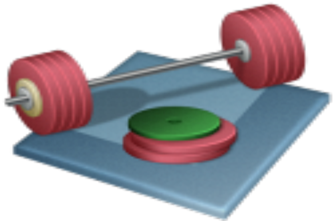
System Overview



Lab Topics

- Get Overview of the Arduino Platform
- Sensors and Actuators Overview in general
- Overview of Temperature Sensors, such as
  - Pt-100, Thermistor, Thermocouple
- Basic Data Acquisition (DAQ), Data Logging
- Calibration, Uncertainty, Resolution, Accuracy, Range, etc.
- Lowpass Filter implementation in Software
- Network Communication
- Reading Data sheets

Assignment Overview



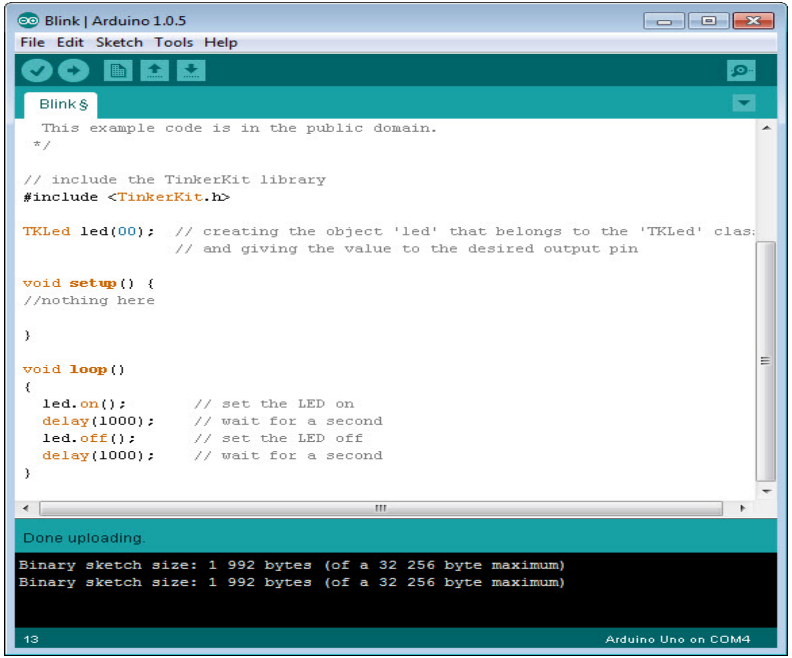
1. Arduino Basics: Explore the different Sensors and Actuators available with the Arduino Kit
2. Temperature Sensors
3. Pt-100: Create your own Pt-100 sensor with Transmitter from scratch and Read Temperature values using Arduino
4. Create a Temperature Data Logger/Embedded DAQ System. Select <u>one</u> of the following alternatives:
   a) Save Data using SD Card available on Arduino Shields
   b) Use an online web-based Data Logging Service like Temboo/Xively
   c) Use Wireless Communication to your PC using XBee Modules

See next slides for details...

Arduino Software

Programming with Arduino is simple and intuitive!

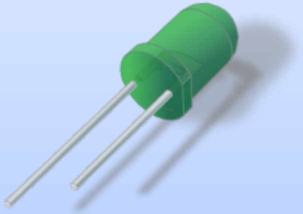
Arduino Sketch IDE



The syntax is similiar to C programming

Example:

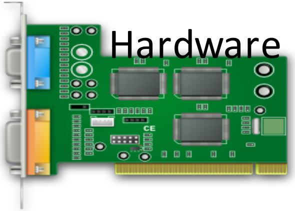
```
// include the TinkerKit library
#include <TinkerKit.h>

// creating the object 'led' that belongs to the 'TKLed' class
TKLed led(O0);


void setup()
 {
                //do something here
 }

void loop()
{
  led.on();          // set the LED on
  delay(1000);       // wait for a second
  led.off();         // set the LED off
  delay(1000);       // wait for a second
}
```



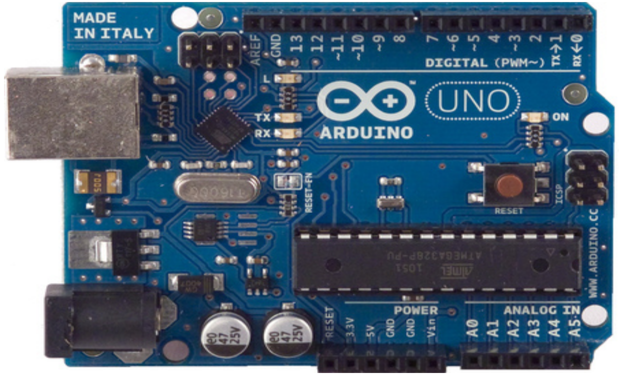
This program makes a LED blink

Software Installation: http://arduino.cc/en/Main/Software

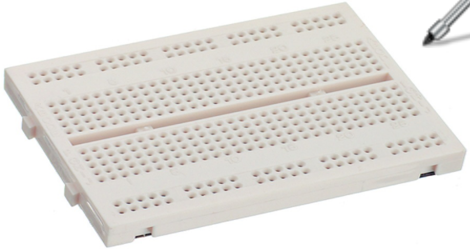
Hardware

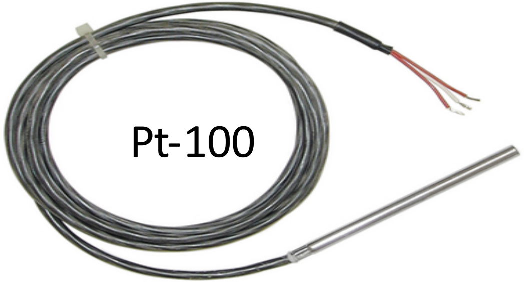

Hardware

Arduino UNO Device



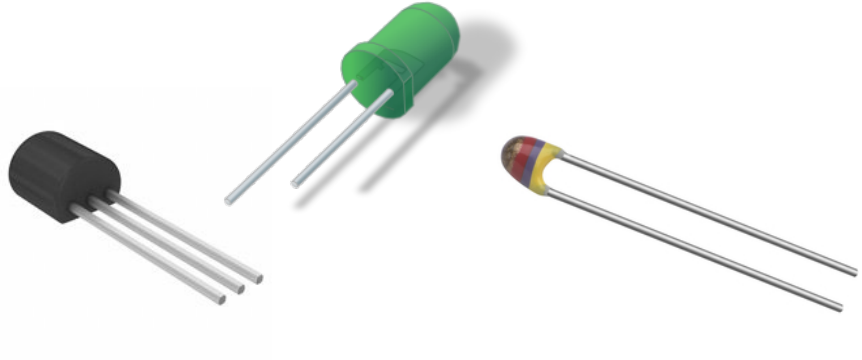
Breadboard




Pt-100

Tools

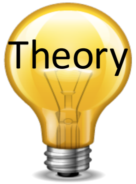
Sensors and Actuators




Multimeter

Pt-100 Transmitter

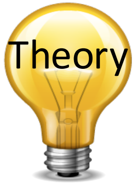
Sensors and Actuators



- A **Sensor** is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an (today mostly electronic) instrument.
- An **Actuator** is a type of motor for moving or controlling a mechanism or system. It is operated by a source of energy, typically electric current, hydraulic fluid pressure, or pneumatic pressure, and converts that energy into motion. An actuator is the mechanism by which a control system acts upon an environment.

http://en.wikipedia.org/wiki/Sensor http://en.wikipedia.org/wiki/Actuator

Sensors

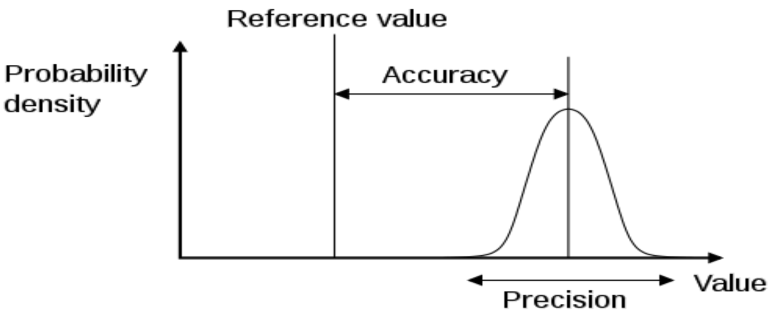


Calibration: A comparison between measurements. One of known magnitude or correctness made or set with one device and another measurement made in as similar a way as possible with a second device. The device with the known or assigned correctness is called the standard. The second device is the unit under test, test instrument, or any of several other names for the device being calibrated.

Accuracy: How close the measured value is the the actual/real value, eg., ±0.1 %

Resolution: The smallest change it can detect in the quantity that it is measuring. The followig formula may be used (where S is the measurment span, e.g., 0-100deg.C):

$$R = \frac{S}{2^n - 1}$$



In the assignment you need to deal with these parameters. You find information about these parameters in the Data sheet for your device

http://en.wikipedia.org/wiki/Calibration

http://en.wikipedia.org/wiki/Measurement_uncertainty

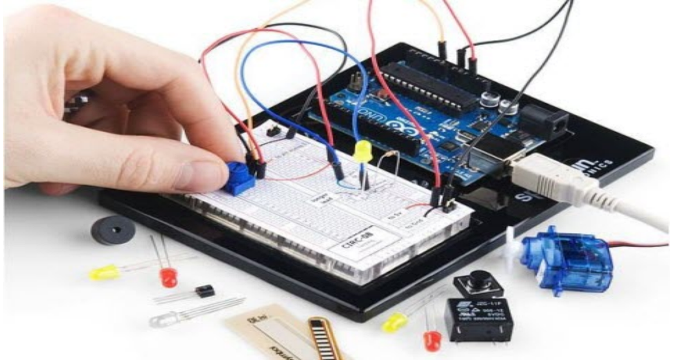
http://en.wikipedia.org/wiki/Accuracy_and_precision

Arduino Basics

Getting Started with Arduino

Hans-Petter Halvorsen, M.Sc.

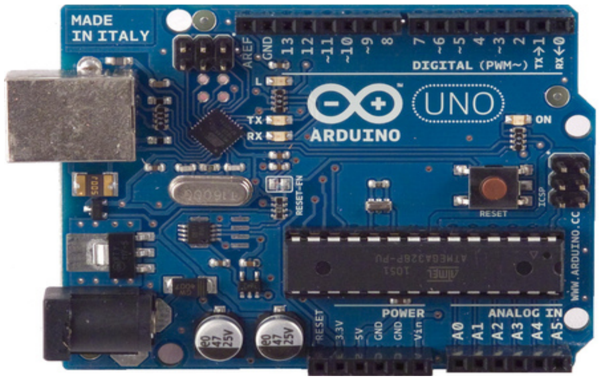
ARDUINO



- Arduino is an open-source physical computing platform designed to make experimenting with electronics and programming more fun and intuitive.
- Arduino has its own unique, simplified programming language and a lots of premade examles and tutorials exists.
- With Arduino you can easily explore lots of small-scale sensors and actuators like motors, temperature sensors, etc.
- The possibilities with Arduino are endeless.

http://www.arduino.cc

Arduino UNO



http://www.arduino.cc

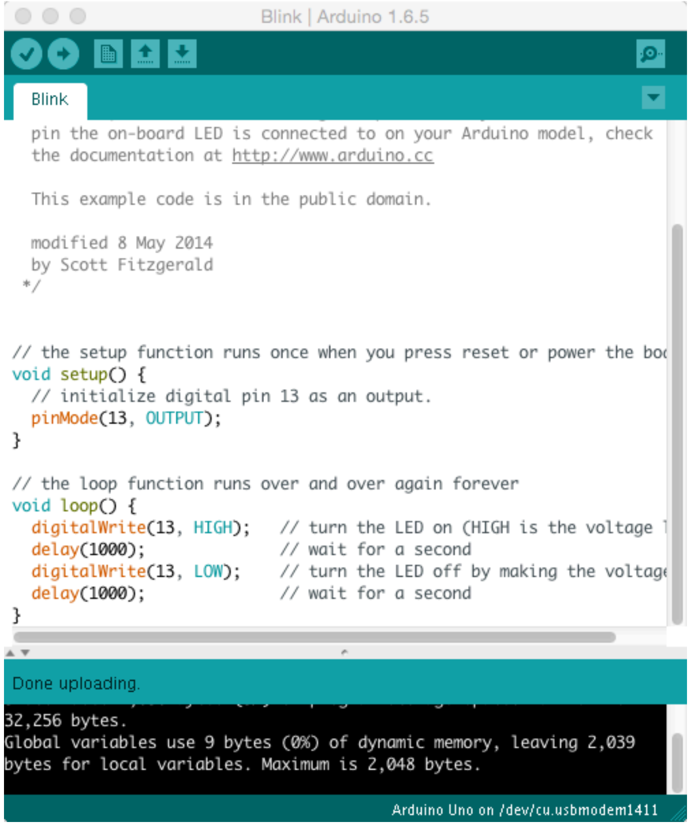
https://www.arduino.cc/en/Guide/HomePage

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Pin Overview: http://pighixxx.com/unov3pdf.pdf

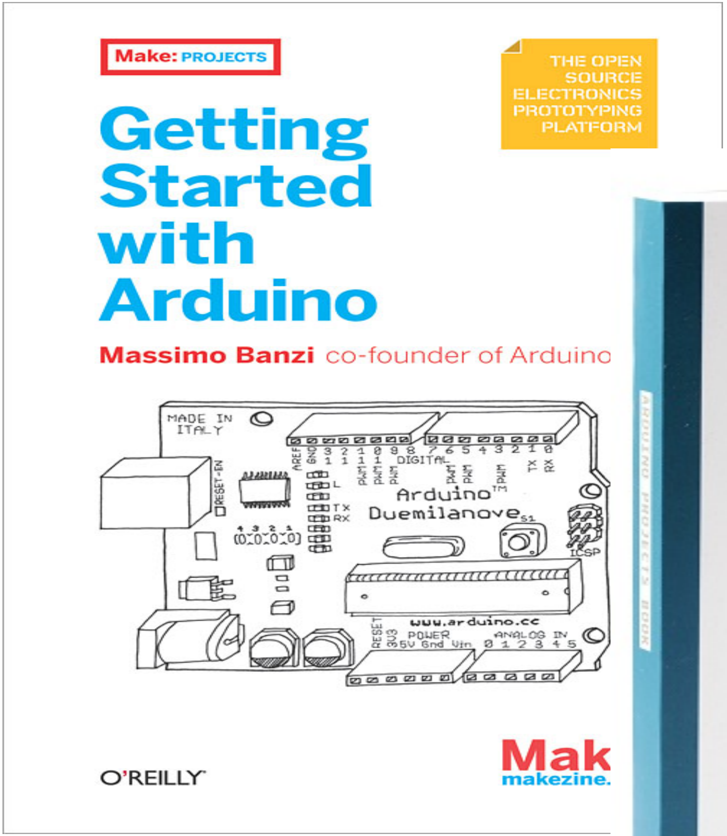
The Arduino Programming Environment (IDE)

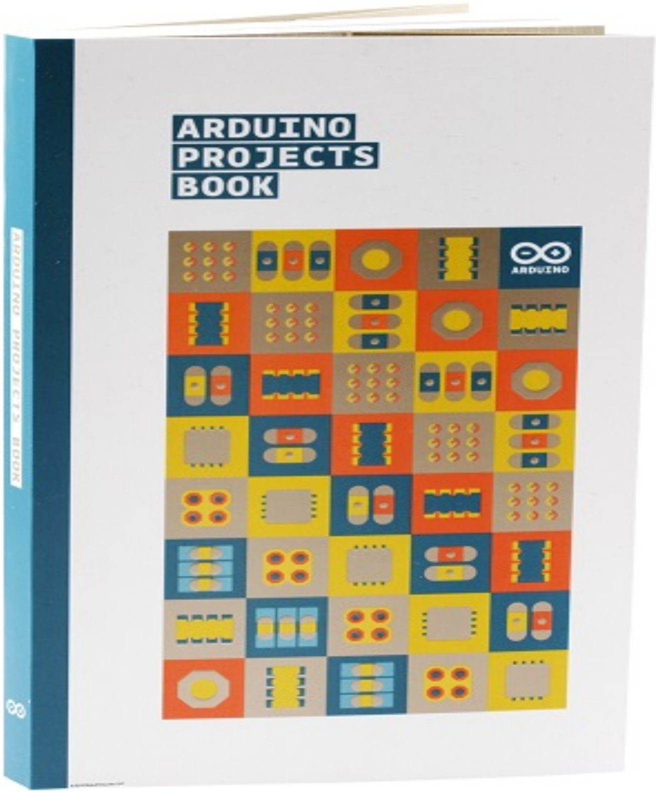
The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.



https://www.arduino.cc/en/Guide/HomePage

Books



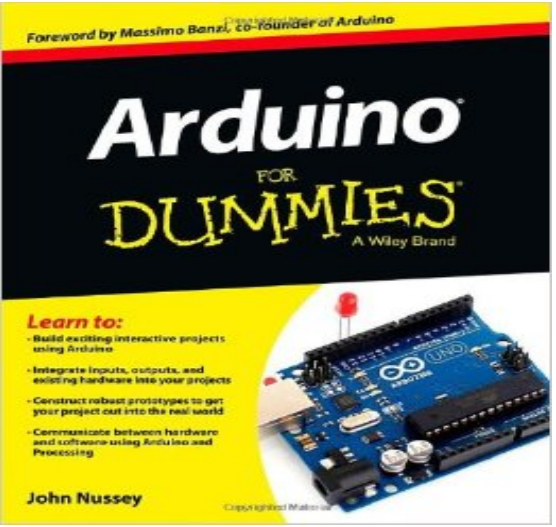


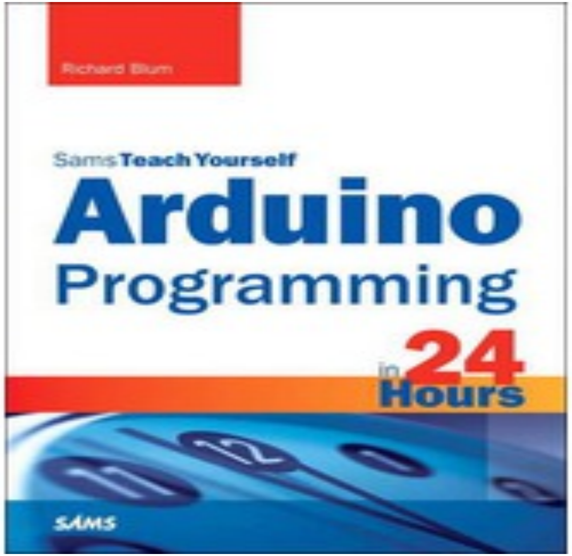
Free download from Internet

These books gives you an introduction to Arduino. These books are available on the lab. Lots of Arduino books are also available on Safari Books Online

Selected **eBooks** from Safari Online available for free for Students and Teachers at TUC

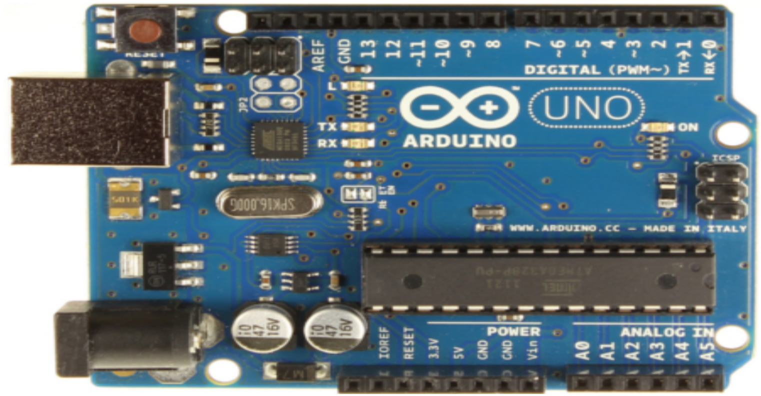
http://proquest.safaribooksonline.com/



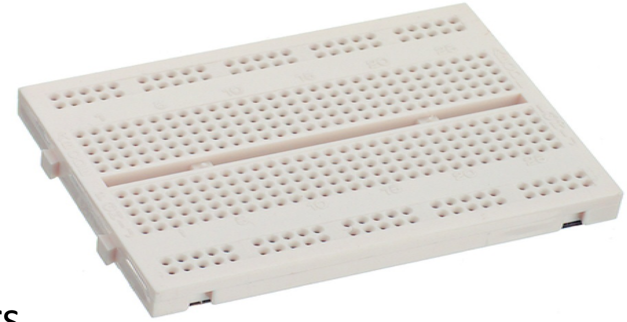


Arduino Basics

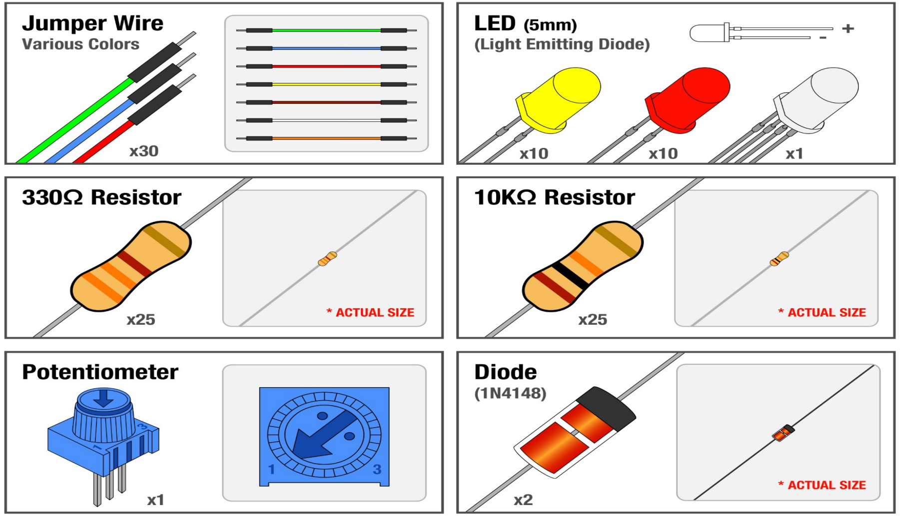
Arduino Uno Board

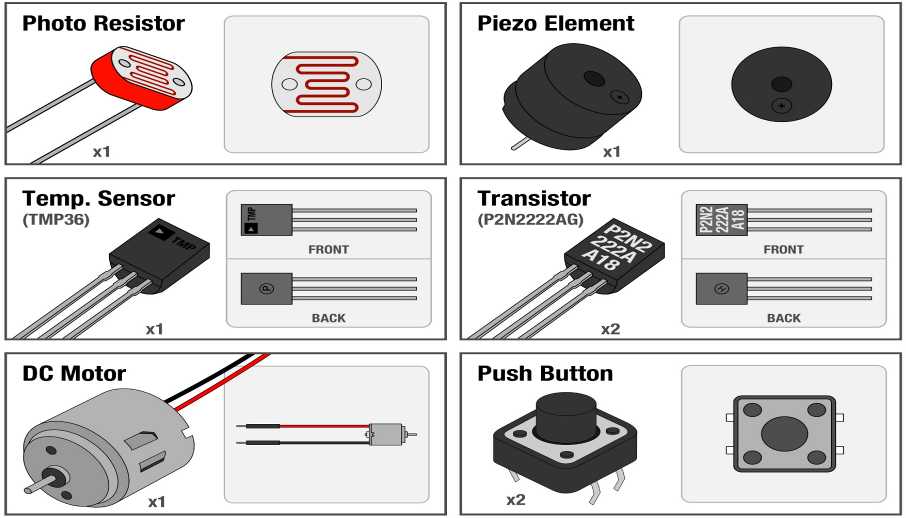


Breadboard

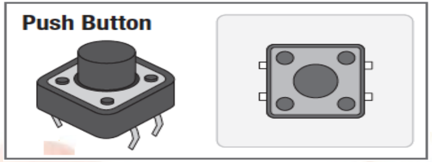


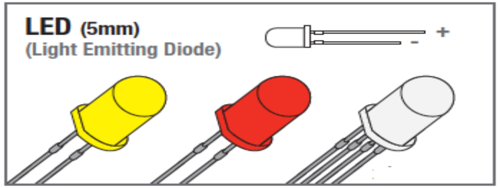
Sensors and Actuators

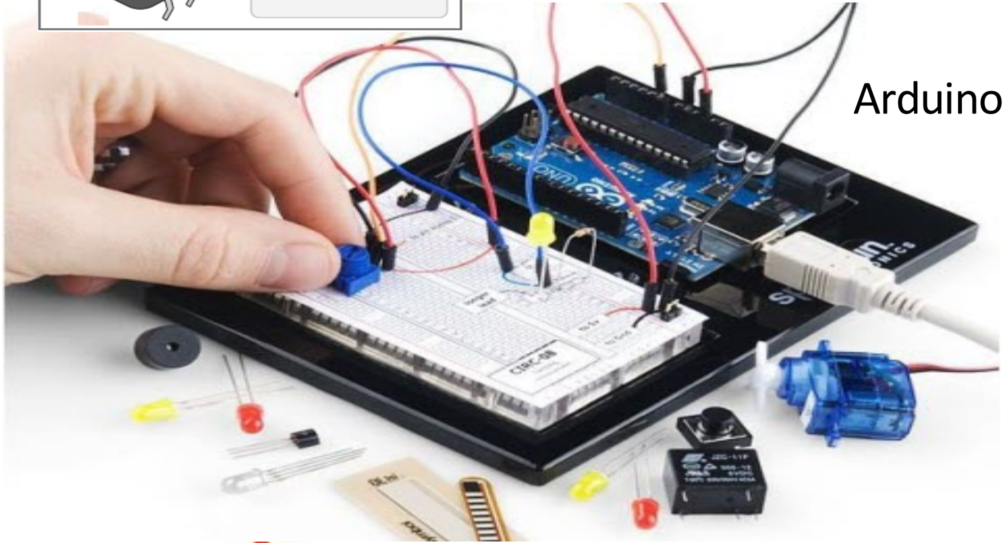




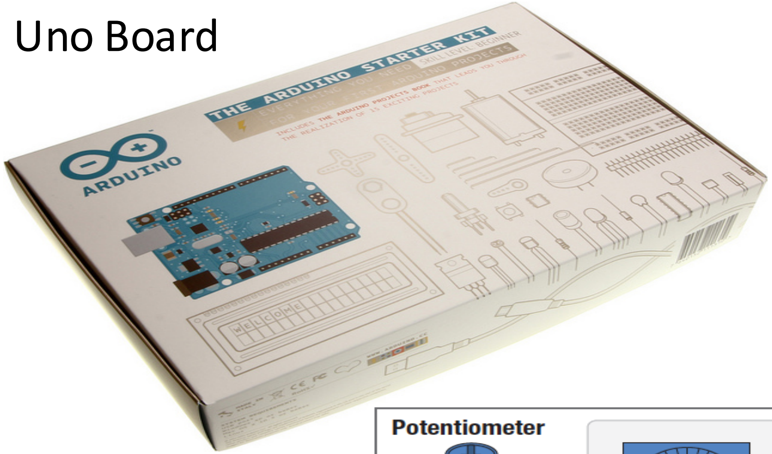
The Arduino Kit

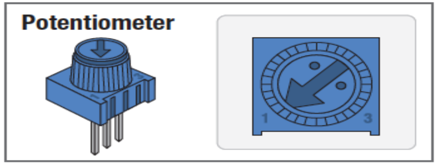




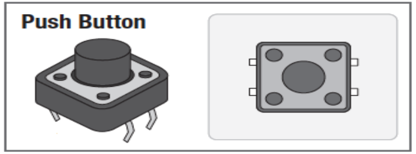


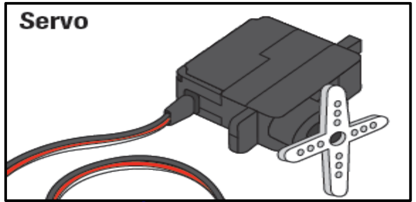
Arduino Uno Board





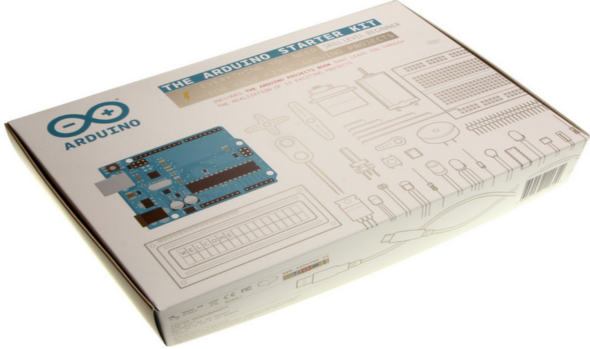
Small-size Sensors and Actuators





Getting Started with Arduino: http://arduino.cc/en/Guide/HomePage

The Arduino Kit

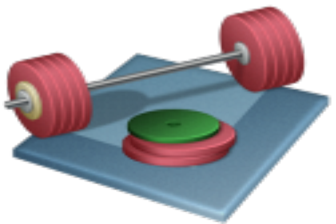


- Ardiono Home Page: http://arduino.cc
- The Arduino Starter Kit:

http://arduino.cc/en/Main/ArduinoStarterKit

- Starter Kit Videos:
  https://www.youtube.com/playlist?feature=edit_ok&list=PLT6rF_I5kknPf2qlVFlvH47qHvqvzkknd

Getting Started



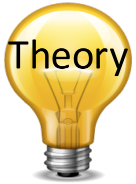
- Arduino with Breadboard Sensors/Actuators:
  - LED
  - Push Button
  - Potentiometer
  - Different Temperature Sensors
  - etc.

- Install the Arduino Sketch and Explore some of these Sensors & Actuators, i.e., make 2-3 Examples.
- Use the breadboard for creating your circuits.
- Use Arduino Sketch in order to create the Programs that interface with the Sensors & Actuators

Congratulations! - You are finished with the Task

Temperature Sensors

Hans-Petter Halvorsen, M.Sc.

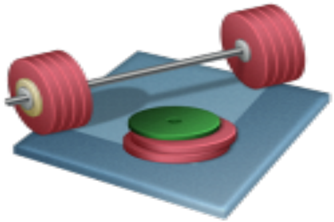


Temperature Measurements

Different methods for measuring the Tempertature:

- Thermocouples
- Thermistors
- RTD (Resistance Temperature Detector)
  - e.g. Pt100
- Infrared
- Thermometers

Temperature Sensors



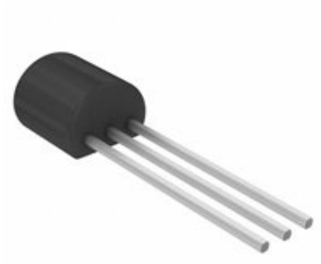
Make the following Temperature Sensors work with Arduino:

NTC Thermistor

TMP36

Small-scale Temperature Sensors

TMP36



Technical data	
Temperature measurement range	-40...+125 °C
Accuracy	±2 °C (0...70 °C)
Power supply	2.3...5.5 V
Package	TO-92
Temperature sensitivity, voltage	10 mV/°C

https://www.sparkfun.com/products/10988

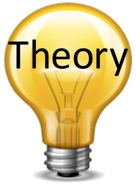
https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=73-889-29&toc=0&q=73-889-29

NTC Thermistor

Technical data	
Resistance @ 25°C	10 kΩ
Temperature range	-40...+125 °C
Power max.	500 mW
Pitch	2.54 mm
Resistance tolerance	±5 %
$W_{25/100}$ value	3977 K
B value tolerance	±0.75 %
Thermal time constant	15 s

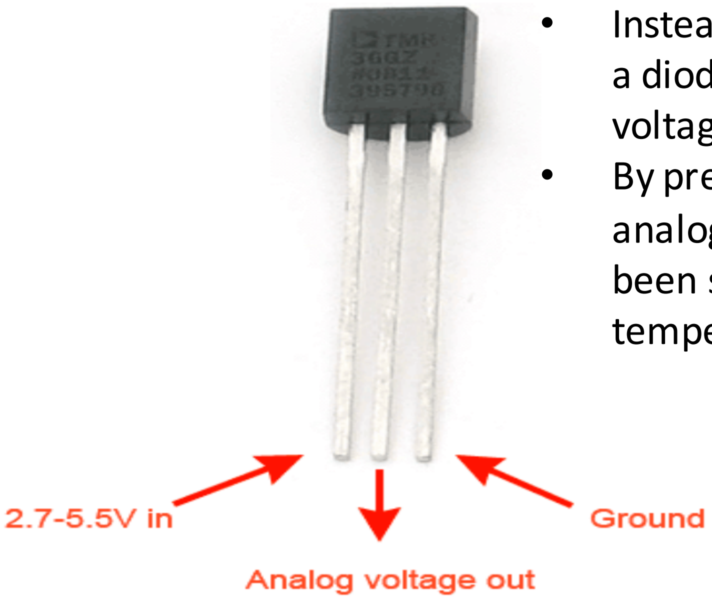
https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=60-260-41&toc=0&q=60-260-41

Tutorial: http://garagelab.com/profiles/blogs/tutorial-using-ntc-thermistors-with-arduino



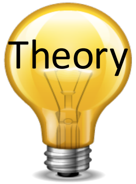
TMP36

- These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetalic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors).
- Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor.)
- By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.



Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistant between sensors and readings. Moreover they are very inexpensive and quite easy to use.

https://learn.adafruit.com/tmp36-temperature-sensor



Datasheet Calculations

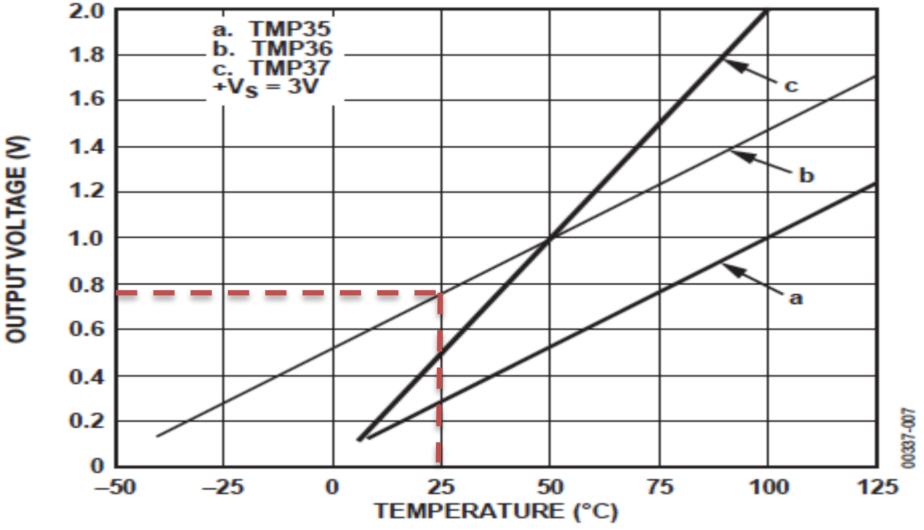


Figure 6. Output Voltage vs. Temperature

From the plot we have:

(x1, y1) = (750mV, 25°C)

(x2, y2) = (1000mV, 50°C)

Linear relationship: $y = ax + b$

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

You have to find a (slope) and b (intercept):

y-25°C = ((50°C-25°C)/(1000mV-750mV)) * (x-750mV)

This gives: y[°C] = (1/10)*x[mv]-50

Voltage-based Sensors

TMP36

According to the TMP36 datasheet, the relation of the output voltage to the actual temperature uses this equation:

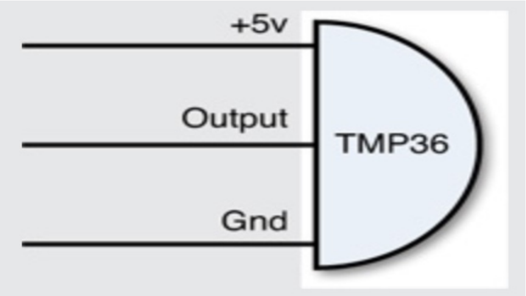
$$y[°C] = (1/10)*x[mv]-50$$

Where the voltage value is specified in millivolts.

However, before you use that equation, you must convert the integer value that the analogRead function returns into a millivolt value.

10-bit analog to digital converter

You know that for a 5000mV (5V) value span the analogRead function will return 1024 possible values:



voltage = (5000 / 1024) * output

mV

Where

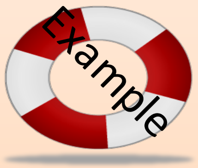
output = analogRead(aichannel)

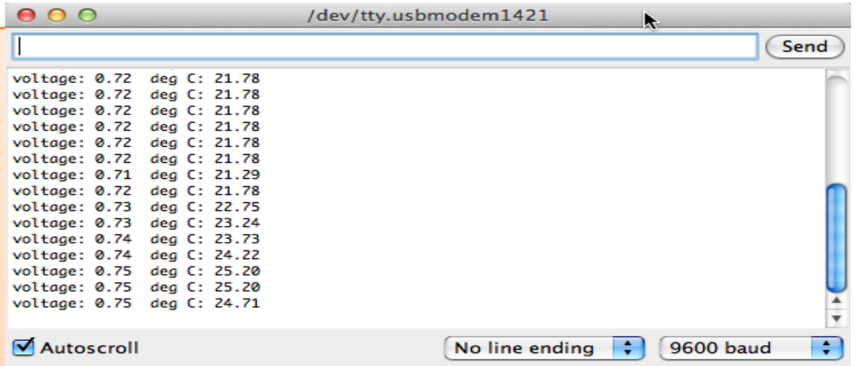
0-1023

A0-A5

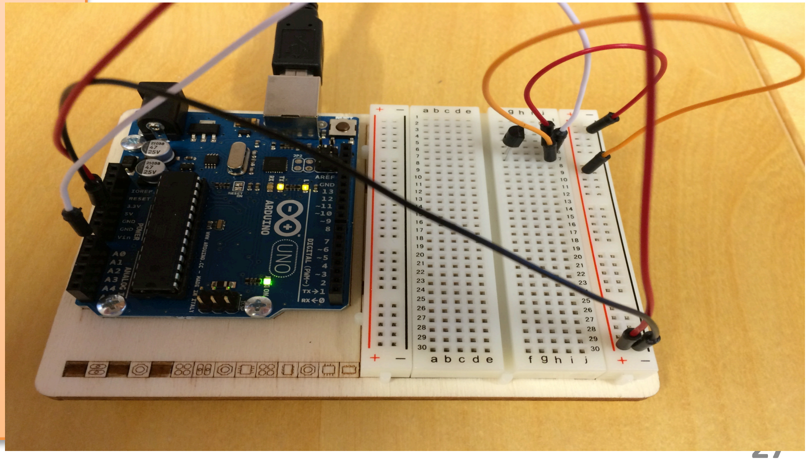
TMP36 Temperature Sensor Example

```
// We'll use analog input 0 to read Temperature Data

const int temperaturePin = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  float voltage, degreesC, degreesF;

  voltage = getVoltage(temperaturePin);

  // Now we'll convert the voltage to degrees Celsius.
  // This formula comes from the temperature sensor datasheet:

  degreesC = (voltage - 0.5) * 100.0;

  // Send data from the Arduino to the serial monitor window
  Serial.print("voltage:  ");
  Serial.print(voltage);
  Serial.print("   deg C: ");
  Serial.println(degreesC);

  delay(1000); // repeat once per second (change as you wish!)
}


float getVoltage(int pin)
{
  return (analogRead(pin) * 0.004882814);

  // This equation converts the 0 to 1023 value that analogRead()
  // returns, into a 0.0 to 5.0 value that is the true voltage
  // being read at that pin.
}
```

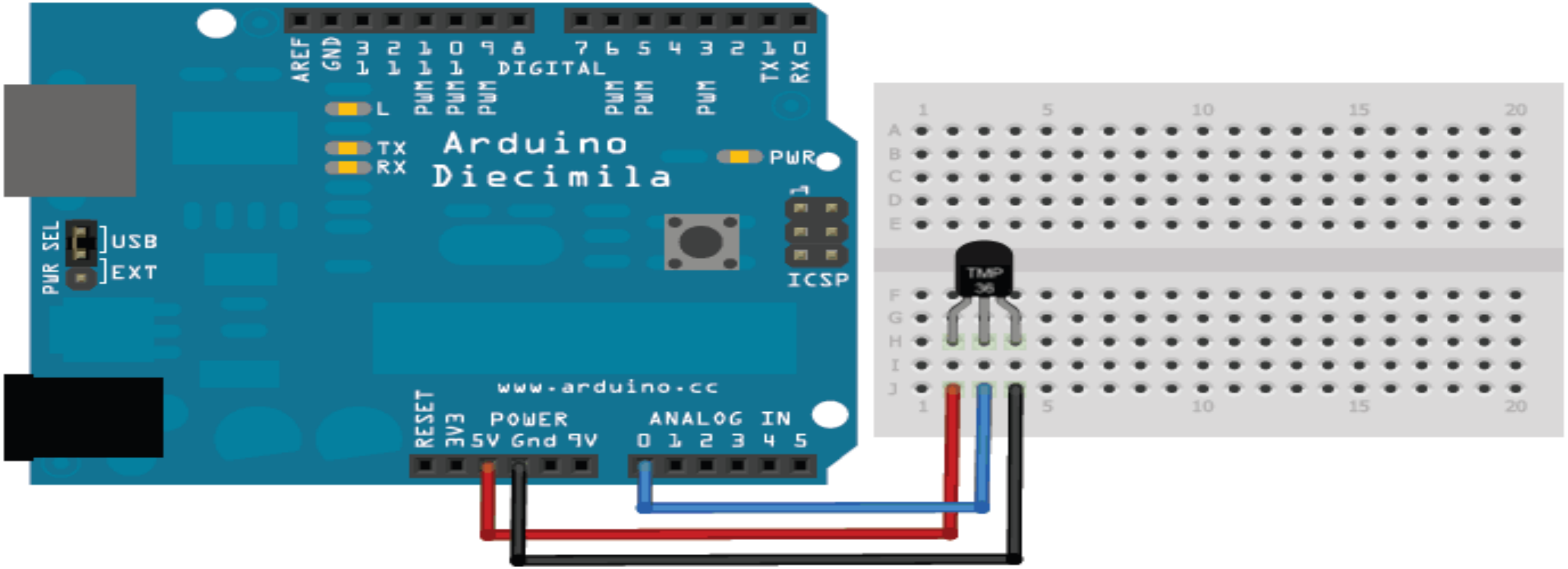




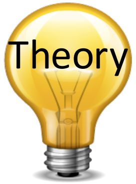
Serial Monitor



TMP36 Temperature Wiring

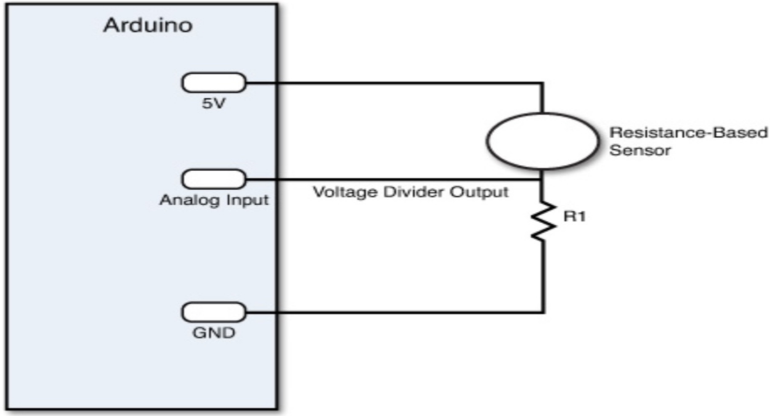


Resistance-based Sensors



The problem with resistance sensors is that the Arduino analog interfaces can't directly detect resistance changes.
This will require some extra electronic components. The easiest way to detect a change in resistance is to convert that change to a voltage change. You do that using a **voltage divider**, as shown below.

Thermistor



By keeping the power source output constant, as the resistance of the sensor changes, the voltage divider circuit changes, and the output voltage changes. The size of resistor you need for the R1 resistor depends on the resistance range generated by the sensor and how sensitive you want the output voltage to change.

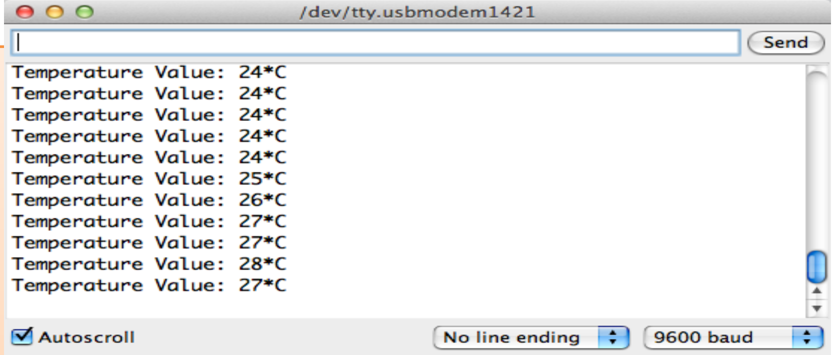
Generally, a value between 1K and 10K ohms works just fine to create a meaningful output voltage that you can detect in your Arduino analog input interface.

E.g., the Steinhart-Hart Equation can be used to find the Temperature:

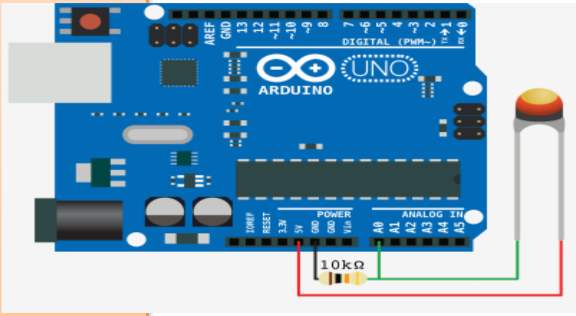
$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

NTC Thermistor Example

```
// Read Temerature Values from NTC Thermistor
const int temperaturePin = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int temperature = getTemp();
  Serial.print("Temperature Value: ");
  Serial.print(temperature);
  Serial.println("*C");
  delay(1000);
}

double getTemp()
{
  // Inputs ADC Value from Thermistor and outputs Temperature in Celsius

  int RawADC = analogRead(temperaturePin);
  long Resistance;
  double Temp;

  // Assuming a 10k Thermistor. Calculation is actually: Resistance = (1024/ADC)
  Resistance=((10240000/RawADC) - 10000);

  // Utilizes the Steinhart-Hart Thermistor Equation:

  // Temperature in Kelvin = 1 / {A + B[ln(R)] + C[ln(R)]^3}
  // where A = 0.001129148, B = 0.000234125 and C = 8.76741E-08

  Temp = log(Resistance);
  Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
  Temp = Temp - 273.15;  // Convert Kelvin to Celsius
  return Temp;  // Return the Temperature
}
```



Serial Monitor



Steinhart-Hart Equation:

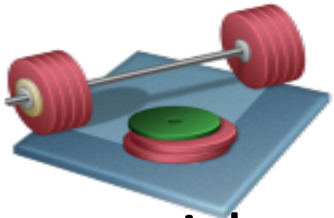
$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

Congratulations! - You are finished with the Task

Pt-100 Measurements

Hans-Petter Halvorsen, M.Sc.

Temperature Measurements

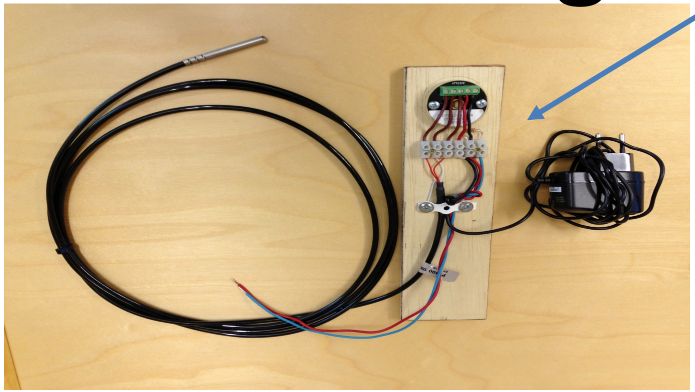


Pt100: Create your own temporary Pt-100 sensor with Transmitter (create the circuit on a breadboard) and then Read Temperature values using Arduino.

Suggested Tasks:

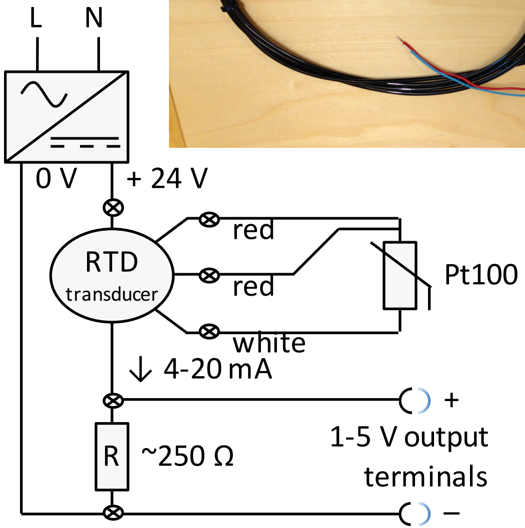
- Compare with Temperature Sensors available within the Arduino Kit (TMP36, Thermistor,...)
- Compare and Discuss the following Temperature Measurements; Pt-100, Thermocouple, Thermistor (Measurement principles, etc.)

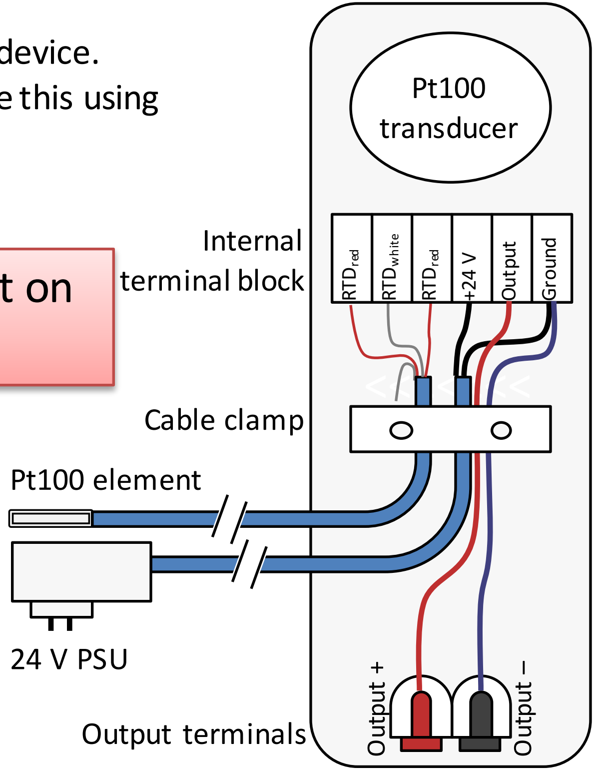
Pt-100 Wiring



This is a pre-made device.
Your job is to create this using a Breadboard

Create this circuit on your breadboard

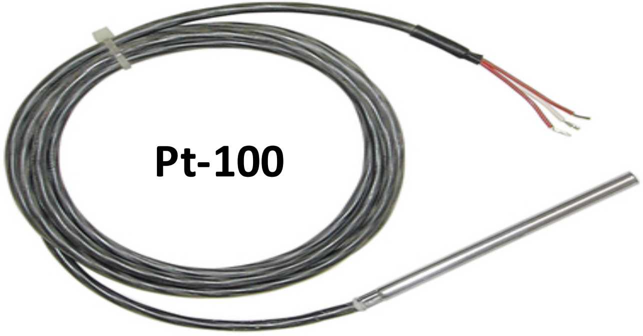




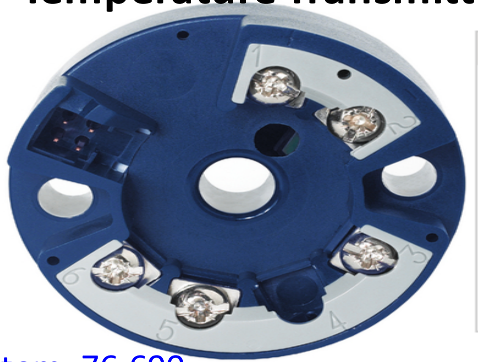
Important! Test the output of your circuit BEFORE connecting it to the Arduino Analog Input with a Multimeter to make sure the voltage is not higher than 5V, else the Arduino will be damaged!

Create you own Pt-100 Sensor with Transmitter

Test the Device using Arduino



Temperature Transmitter (0-100deg. C)



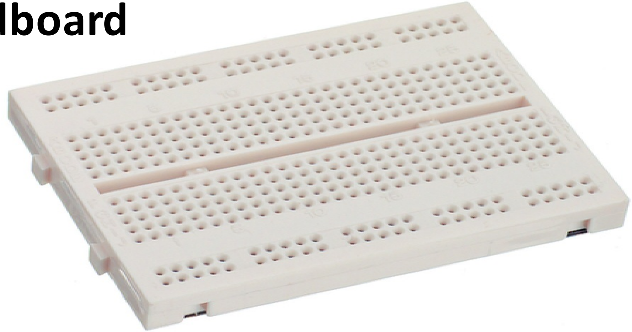
Technical data	
Operating voltage	7...30 VDC
Temperature measurement range	0...100 °C
Input, resistance thermometer	Pt100
Output, analogue	4...20 mA
Operating temperature	-40...+85 °C
Protection rating	IP 20, free mounting
Dimensions ø x H	44 x 21 mm
Accuracy	±0.1 %

https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=76-690-51&toc=0&q=76-690-51

https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=76-895-74&toc=0&q=76-895-74

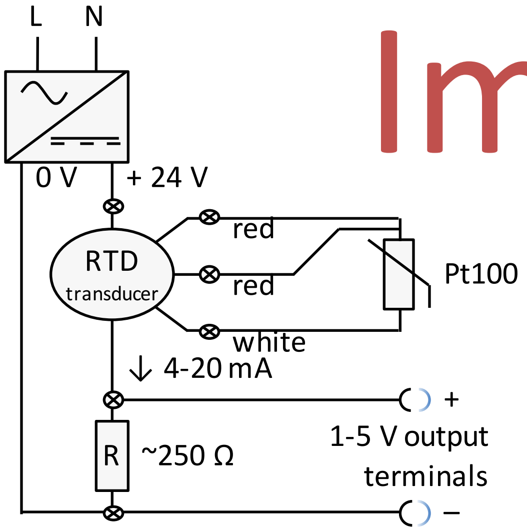
24VDC/0.25A Power Supply

Breadboard



https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=69-061-79&toc=0&q=69-061-79

Important!



- Test the output of your circuit BEFORE connecting it to the Arduino Analog Input
- Use a Multimeter to make sure the voltage is not higher than 5V
- The Arduino will be damaged if the voltage is higher than 5V!

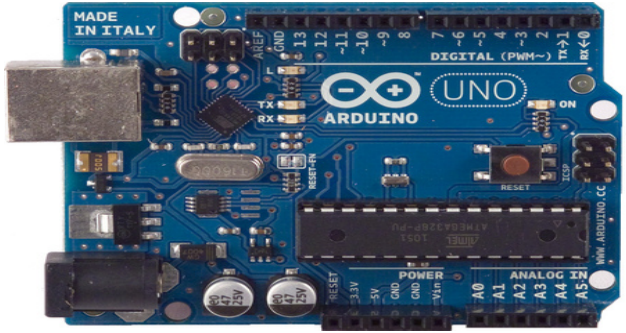
Congratulations! - You are finished with the Task

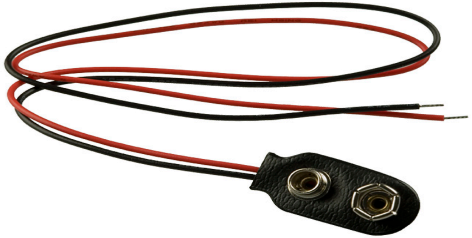
Temperature Data Logger/ Embedded DAQ System

Hans-Petter Halvorsen, M.Sc.

Temperature Data Logger/Embedded DAQ System

You use the PC when creating the software, then you download the software to the Arduino and disconnect the USB cable. Use e.g., a 9V battery or an external Power Supply.

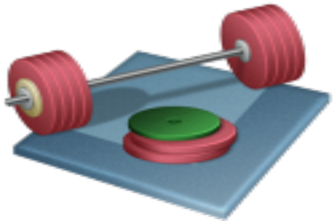




NTC Thermistor

Use different Temperature sensors for comparison, i.e log data from 2 different sensors at the same time.

Temperature Data Logger/ Embedded DAQ System

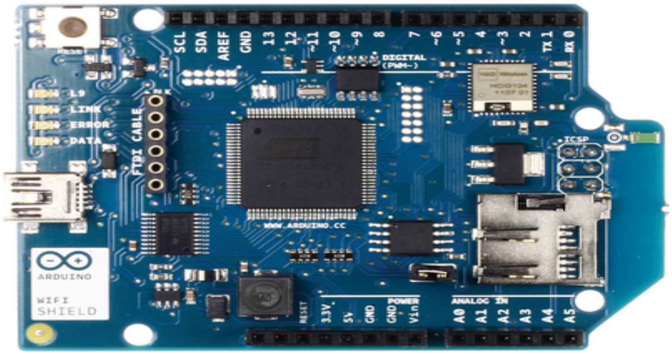


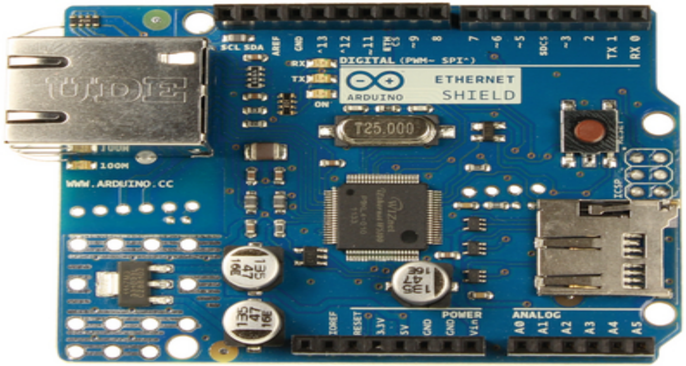
Create a **Temperature Logger**/Embedded DAQ System. Suggested Tasks:

- Create and use a **Lowpass Filter/Average Filter**
- **Alarm** functionality: Use LEDs with different colors when Temperature is above/below the Limits
- Use e.g., Arduino **Wi- Fi/Ethernet Shield** for Communication over a network - or use the microSD card on these Shields
- Save the data to a microSD card located on the Wi- Fi/Ethernet Shield - or connect e.g., to **xively.com** or **temboo.com** - which are free datalogging sites.
- Log Temperature Data for e.g., 24 hours and import Data into Excel, LabVIEW or MATLAB for Analysis and Visualization
- Use e.g. a 9V battery or an external power source to make it portable and small

Arduino Wi-Fi/Ethernet Shield

http://arduino.cc/en/Reference/WiFi





With the Arduino Wi-Fi/Ethernet Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones.

Arduino Wi-Fi Library: http://arduino.cc/en/Reference/WiFi
Arduino Ethernet Library: http://arduino.cc/en/Reference/Ethernet
SD Library: http://arduino.cc/en/Reference/SD

Discrete Lowpass Filter

Note! Implement the Lowpass Filter as a separate Function

Lowpass Filter Transfer function:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{T_f s + 1}$$

Inverse Laplace gives the differential Equation:

$$T_f \dot{y} + y = u$$

We use the Euler Backward method:

$$\dot{x} = \frac{x_k - x_{k-1}}{T_s}$$

This gives:

$$T_f \frac{y_k - y_{k-1}}{T_s} + y_k = u_k$$

$$\downarrow$$

$$y_k = \frac{T_f}{T_f + T_s} y_{k-1} + \frac{T_s}{T_f + T_s} u_k$$

We define:

$$\frac{T_s}{T_f + T_s} \equiv a$$

This gives:

$$y_k = (1 - a) y_{k-1} + a u_k$$

Filter output (y_k) — Noisy input signal (u_k)

This algorithm can be easly implemented in a Programming language

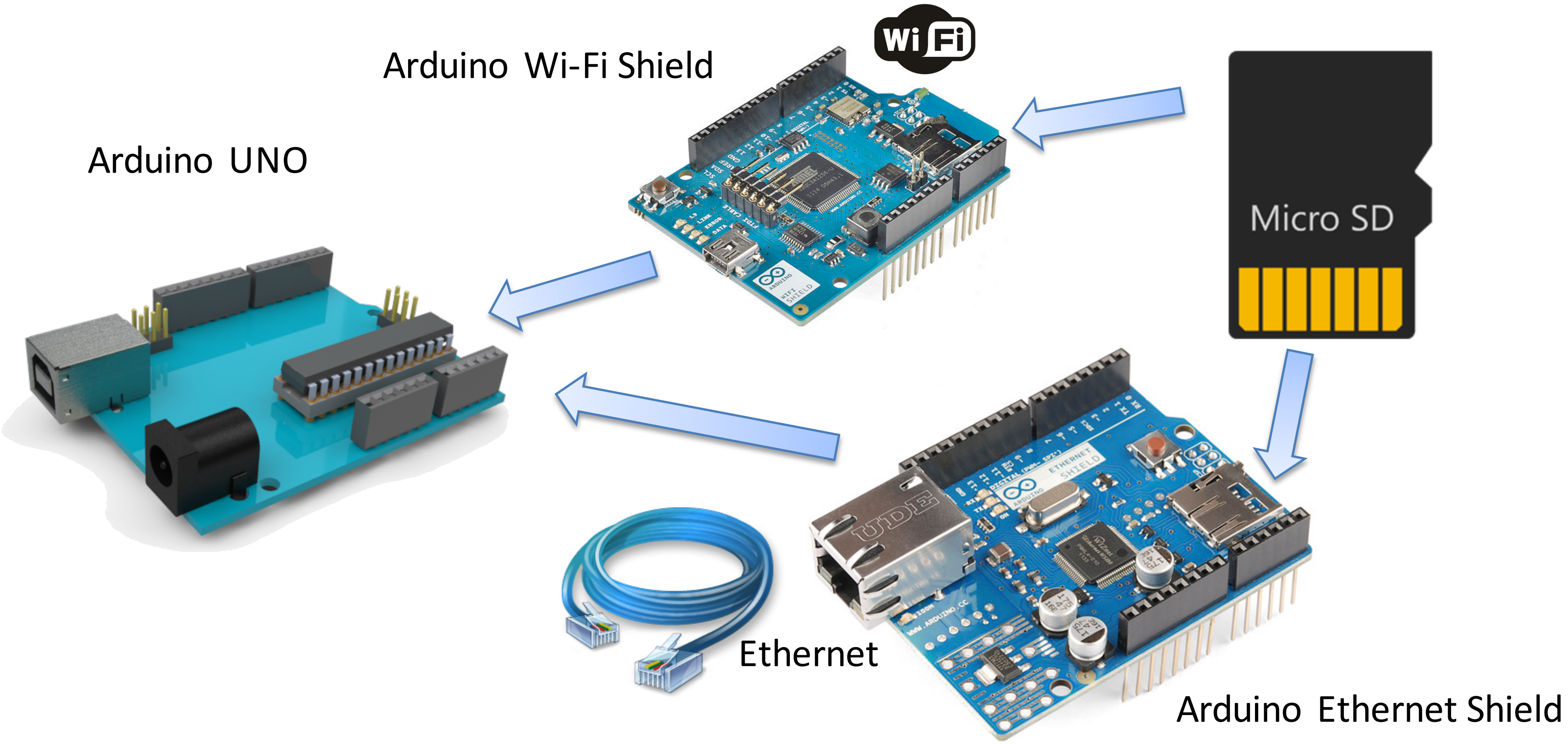
$$T_s \leq \frac{T_f}{5}$$

Arduino Libraries

- The Arduino environment can be extended through the use of libraries, just like most programming platforms.
- Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. A number of libraries come installed with the IDE, but you can also download or create your own.
- You could say an Arduino Library is like a Class with Methods/Functions. It is a good way to structure your code
- Examples:
  - Ethernet Library for the Arduino Ethernet Shield
  - SD Library: The SD library allows for reading from and writing to SD cards, e.g. on the Arduino Ethernet Shield or Arduino Wi-Fi Shield
  - Wi-Fi library for the Arduino Wi-Fi Shield
- Writing your own libraries: https://www.arduino.cc/en/Hacking/LibraryTutorial

https://www.arduino.cc/en/Reference/Libraries

Arduino Ethernet/Wi-Fi Shield with SD Card



Web-based Logging Service

You may want to connect e.g., to xively.com, a free datalogging site

https://xively.com

Arduino xively.com: https://xively.com/dev/tutorials/arduino_wi-fi

http://www.twilio.com/

Use the "Xively for Arduino" library in order to connect and store measurement data from your Arduino device into the Xively cloud

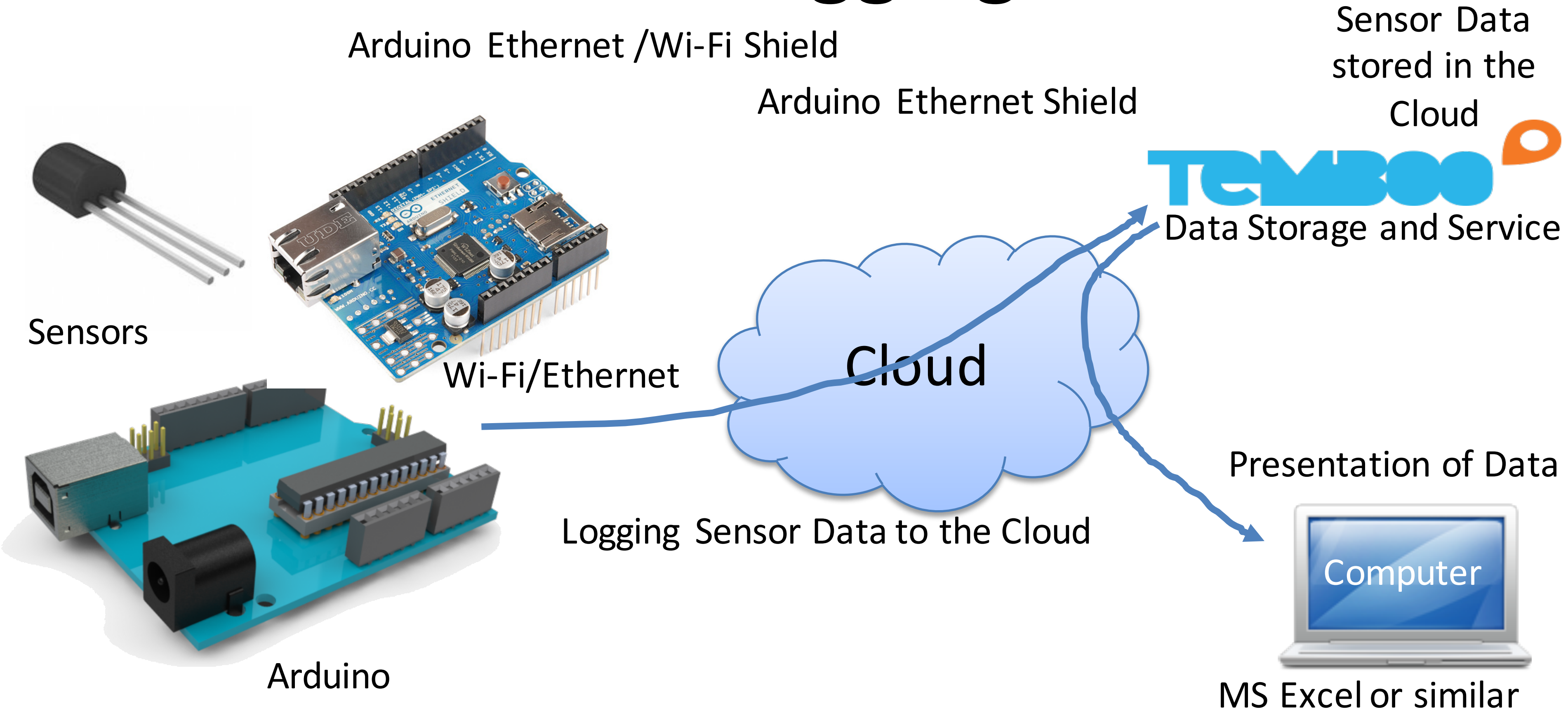
Another alternative is:

www.temboo.com/arduino

www.temboo.com

SMS: http://www.twilio.com

Web-based Logging Service

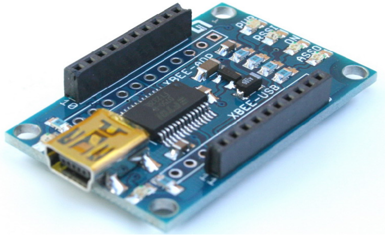


www.digi.com/xbee/

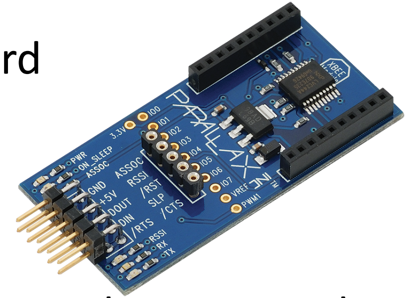
XBee - Wireless Communication

https://en.wikipedia.org/wiki/XBee

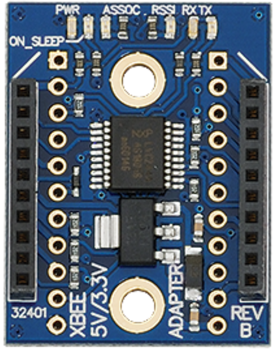
XBee Module



XBee USB Adapter Board



XBee SIP Adapter Board



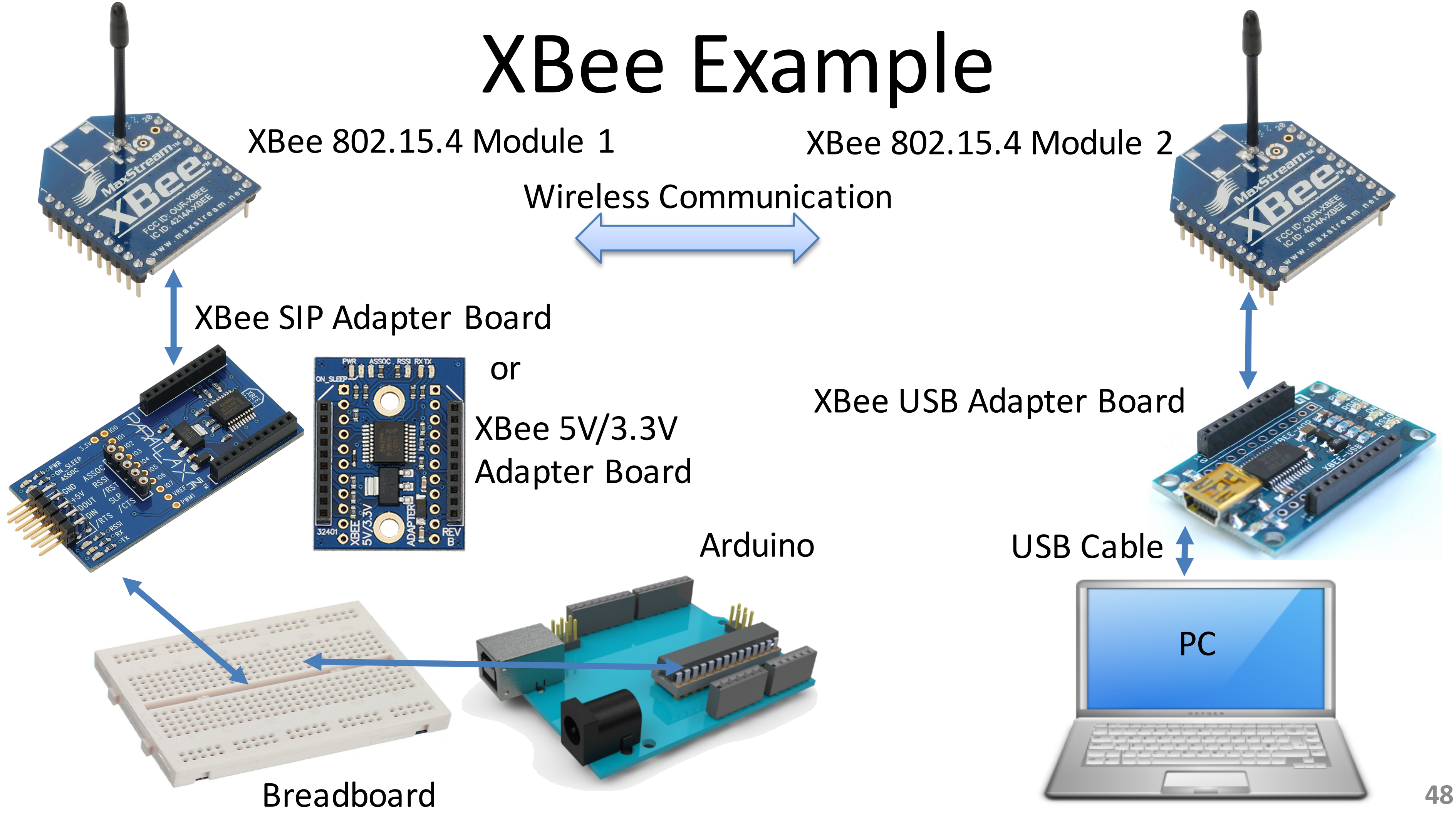
XBee 5V/3.3V Adapter Board

Arduino Tutorial: Let’s make XBee talk!:
http://www.norwegiancreations.com/2013/10/arduino-tutorial-1-lets-make-xbee-talk/

Making Things Talk, 2nd Edition (eBook available at Safari Books Online):
http://proquest.safaribooksonline.com/book/hardware-and-gadgets/9781449314668

Exploring XBees and XCTU:
https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu?_ga=1.116328385.696451024.1434708629

XBee Example



Congratulations! - You are finished with the Task

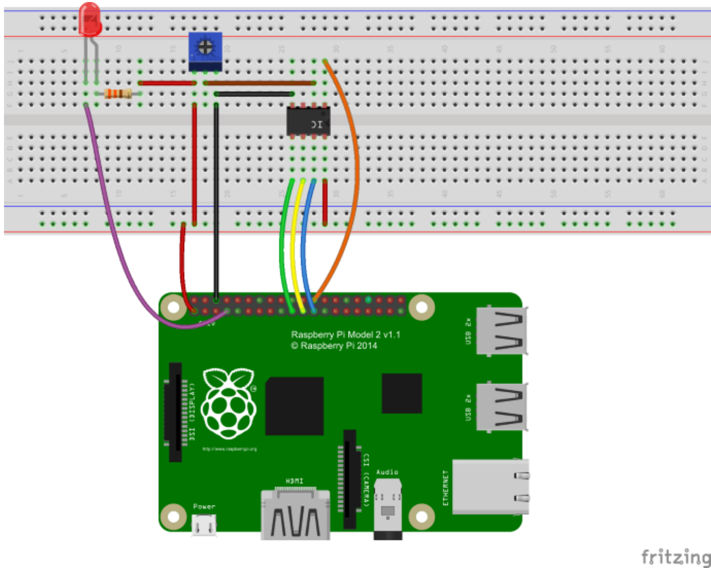
Fritzing

A open source tool for making simple wiring diagram for your hardware wiring

https://en.wikipedia.org/wiki/Fritzing

http://fritzing.org

Wiring made with Fritzing



Congratulations! - You are finished with <u>all</u> the Tasks in the Assignment!

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: http://home.hit.no/~hansha/