HAB 621 Instrumentation and Measurement in Biomechanics

Çizim: De Motu Animalium G.Borelli (1680)

Serdar Arıtan
serdar.aritan@hacettepe.edu.tr
Hacettepe Üniversitesi
www.hacettepe.edu.tr
Spor Bilimleri Fakültesi
www.sbt.hacettepe.edu.tr
Biyomekanik Araştırma Grubu
www.biomech.hacettepe.edu.tr

Data Acquisition Concepts

- Analog to Digital Conversion (ADC)
- Digital to Analog Conversion (DAC)
- Digital Input/Output (DIO)

Data Acquisition Concepts

ADC : Once data has been converted from analog to digital, the digital information can then be processed by the computer, or transferred to memory.

DAC : converts stored data back to a continuous signal (analog voltage) for display or control purposes. Output from the D/A converter can be used to drive external devices which require an analog input

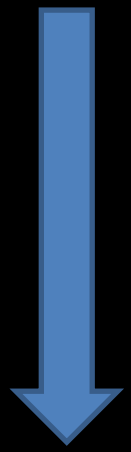
Data Acquisition Concepts

Analog signals are "continuous" signals Represented by continuously changing physical quantities Level of signal can be increased or decreased indefinitely Ex. temperature, pressure, light, strain, voltage.

Digital signals are "discrete" signals represented by separate, individual units Units are represented by "bits" (binary digit).

Data Acquisition Concepts

- Converts analog data into digital data which can be processed
- Typical components:
  - Multiplexers
  - Amplifiers
  - Sample and hold circuits
  - A/D converters



Data Acquisition Concepts

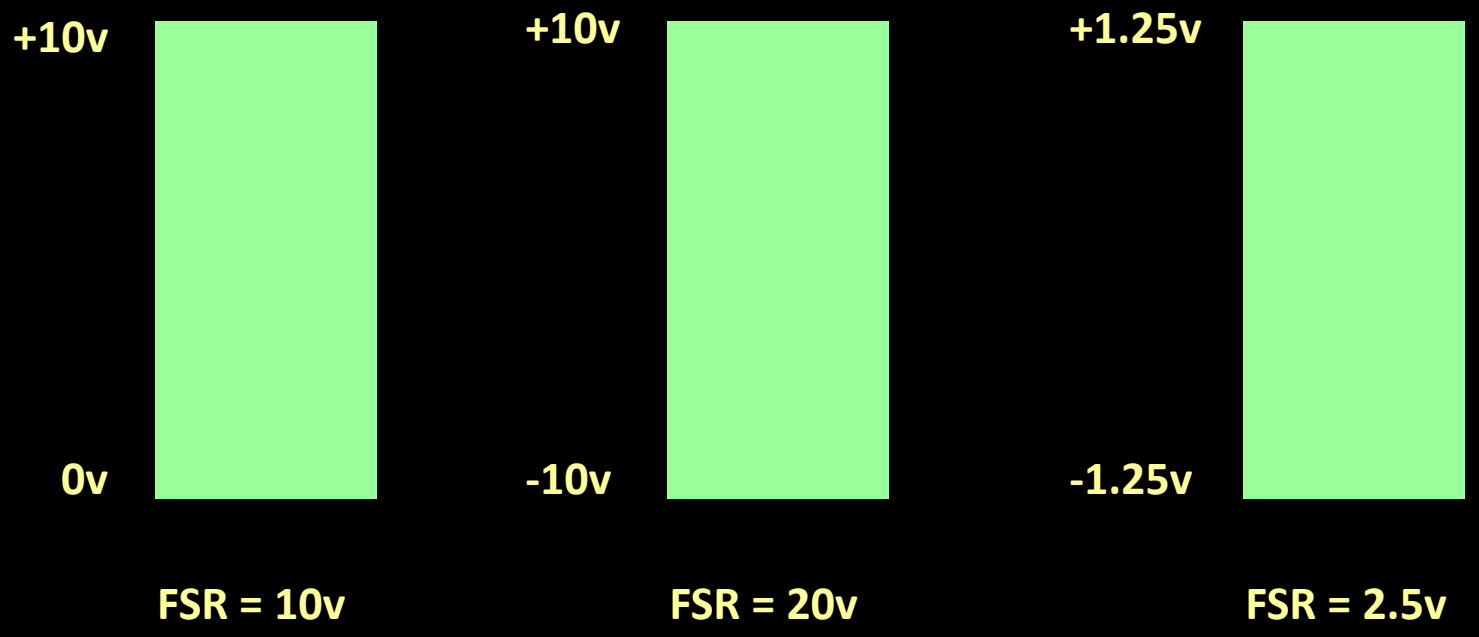
- Input and output (I/O)
  - Analog input is converted into a digital number, by comparing the voltage with its position within the Full Scale Range
  - With an n-bit A/D converter the number of output levels equals 2^n – e.g. 12-bit converter = 2^{12} = 4096
  - Full scale range refers to the largest voltage range which can be input into the A/D converter

Data Acquisition Concepts

- Range is an input span for an A/D and D/A system
- Typical ranges are based on available sensors

Uni-polar (positive)	Bipolar
0 to 5 volts	-5 to +5 volts
0 to 10 volts	-10 to +10 volts

Data Acquisition Concepts



Data Acquisition Concepts

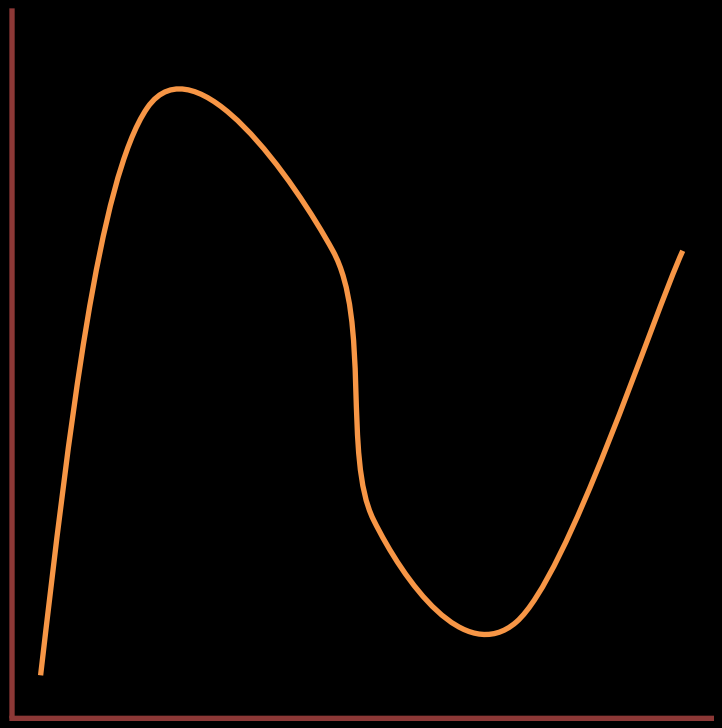
Data Acquisition Concepts

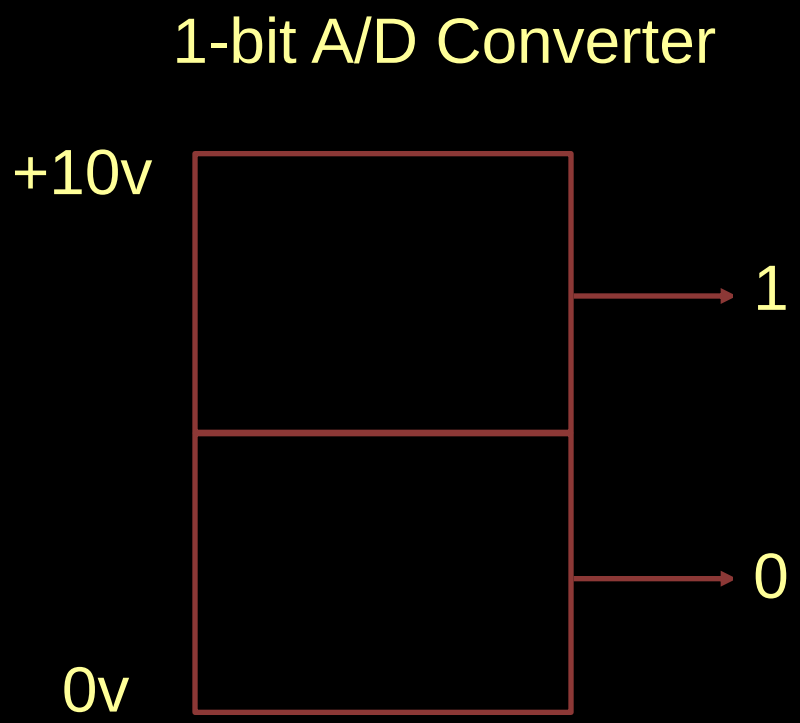
- Resolution determines the smallest change that can be detected
- Specified in bits. Determines number of output levels, or steps
  - **8 bits = 256 steps**
  - **10 bits = 1024 steps**
  - **12 bits = 4096 steps**
  - **16 bits = 65,536 steps**

Data Acquisition Concepts

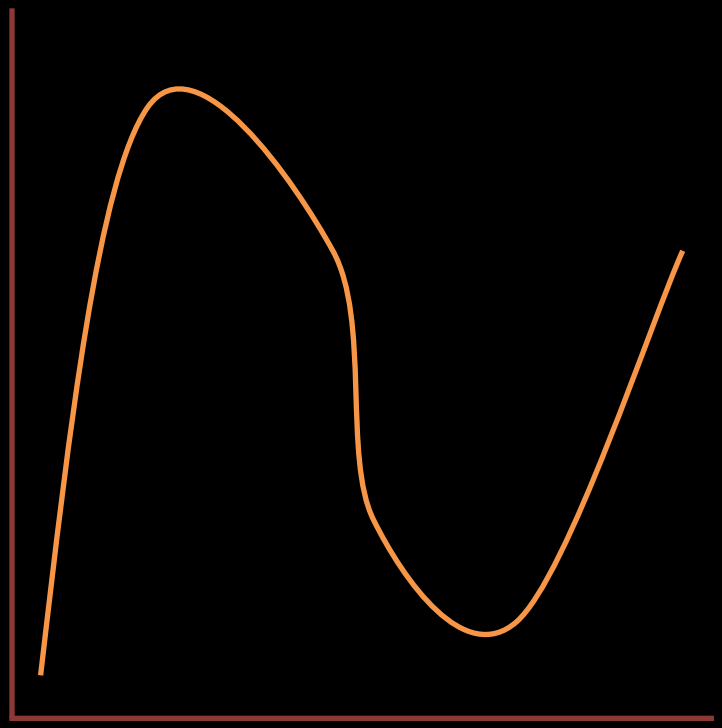
- 8-bit – common for image capture
- 10-bit – general analog acquisition
- 12-bit – general analog acquisition
- 16-bit – precision analog acquisition
- 24-bit – high-accuracy analog acquisition

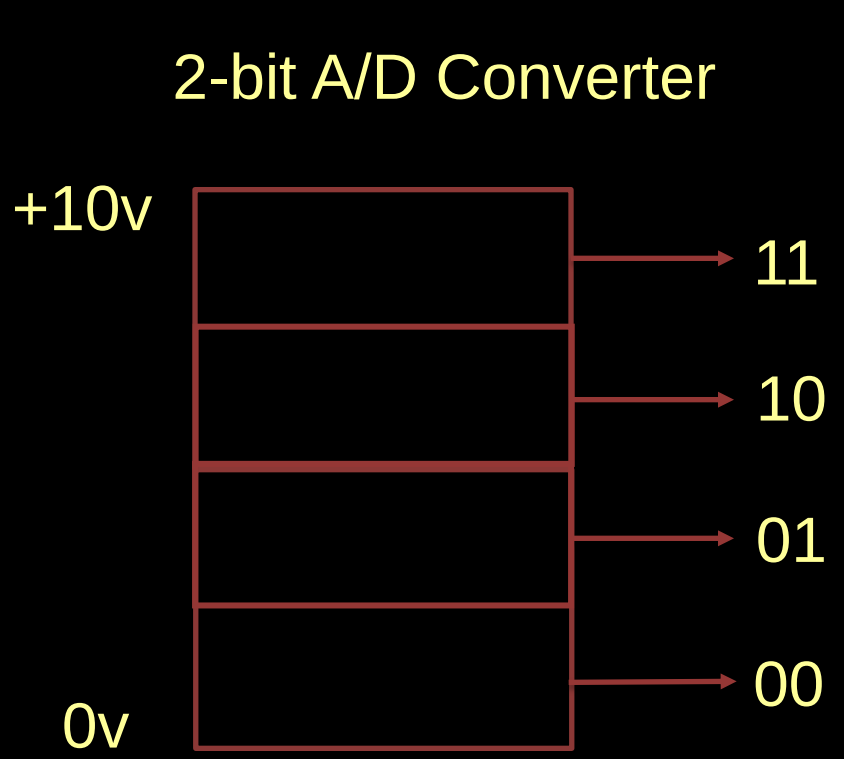
Data Acquisition Concepts



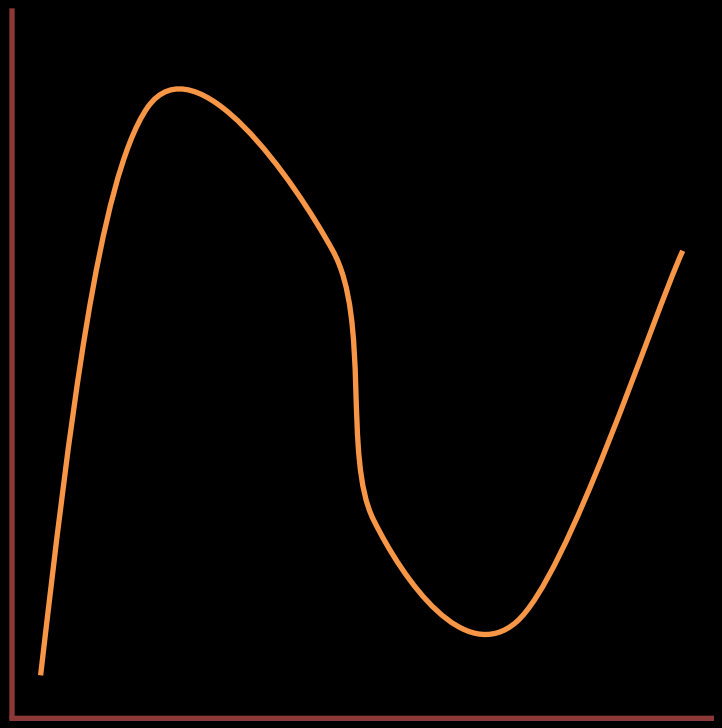


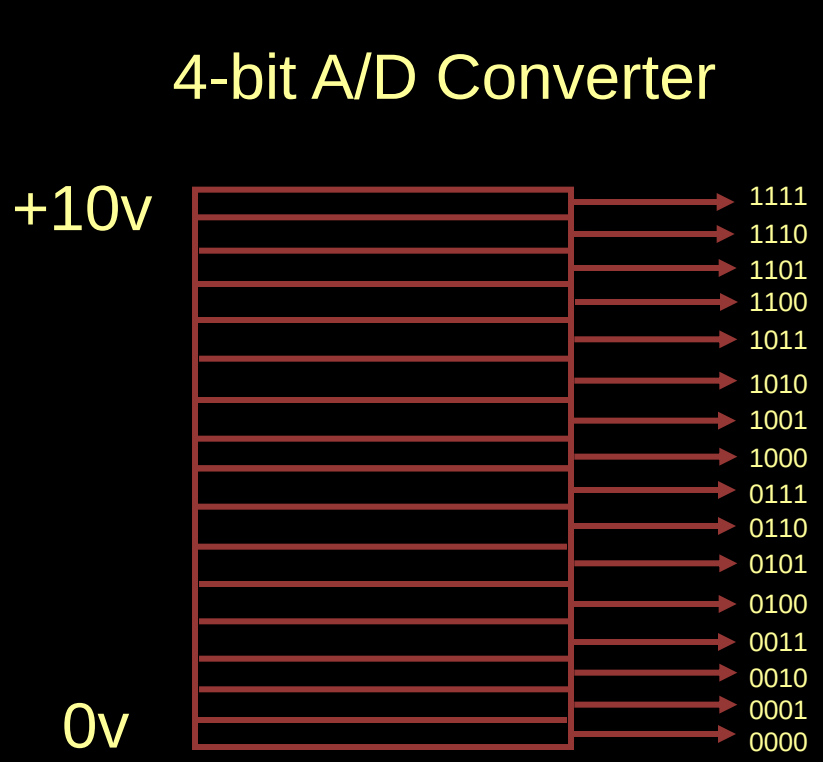
Data Acquisition Concepts



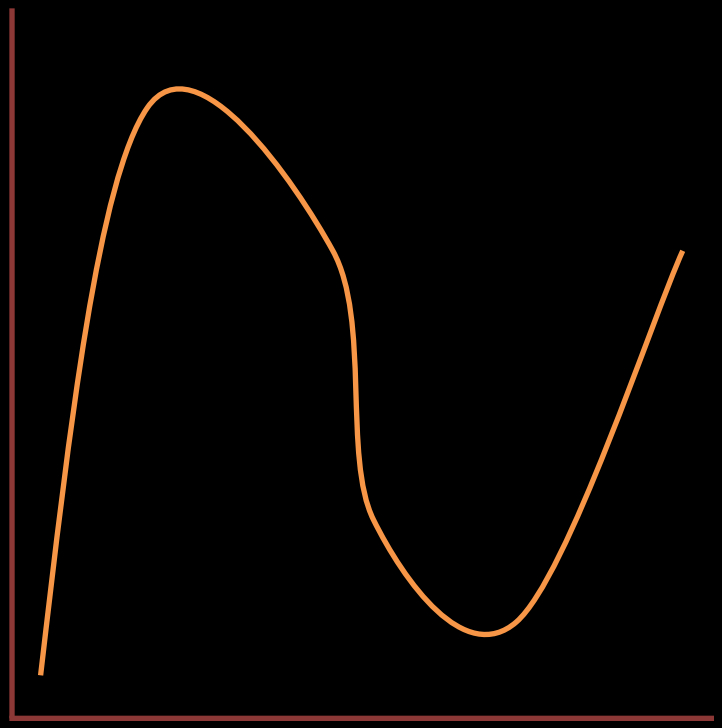


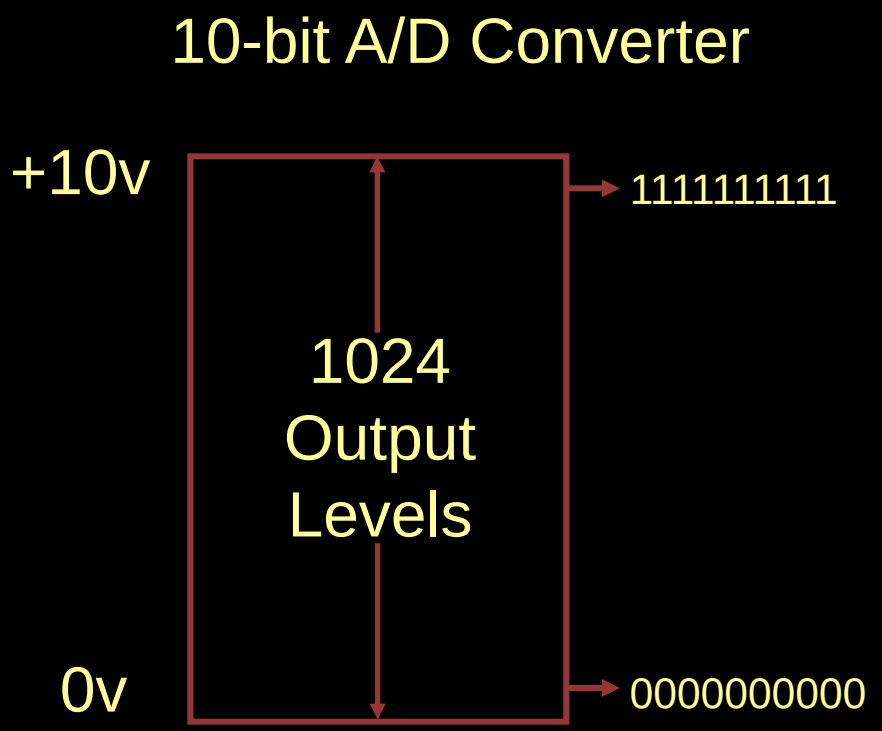
Data Acquisition Concepts





Data Acquisition Concepts





Data Acquisition Concepts

- LSB stands for “least significant bit”
- An LSB represents the smallest change that can be resolved by the A/D converter
- An LSB carries the smallest value or weight
- An LSB is the rightmost bit
- LSB = Full Scale Range (FSR) ÷ 2^n

Data Acquisition Concepts

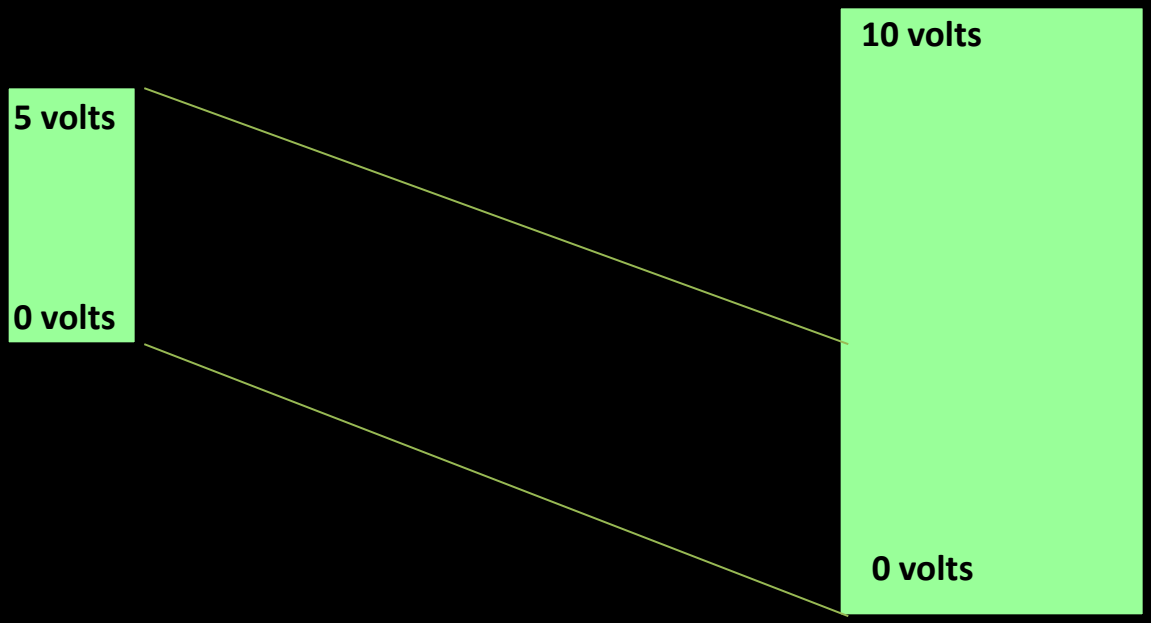
The time required to perform a complete conversion from the analog signal to digital

- The time required after receipt of "start digitizing" command until the A/D converter has finished digitizing
- Time it takes to switch to a new channel
- Each time a user switches between channels, there is a delay, referred to as *settling time*

Data Acquisition Concepts

User Input

Selected Range (uni-polar)

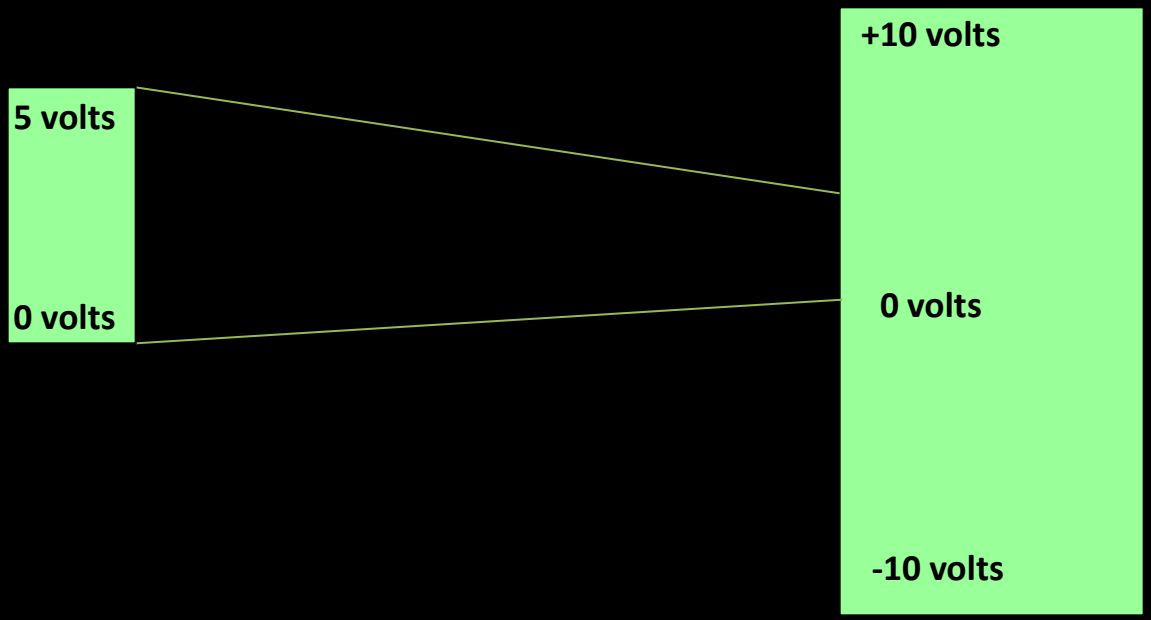


Only 50% of the available A/D range is used.

Data Acquisition Concepts

User Input

Selected Range (bipolar)

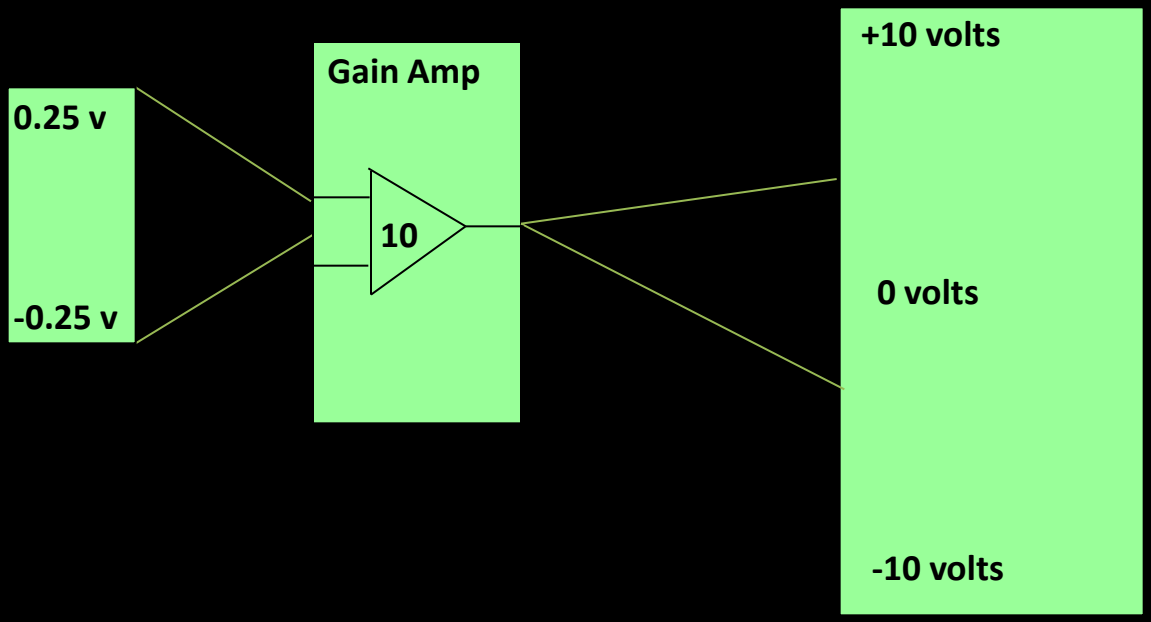


Only 25% of the available A/D range is used.

Data Acquisition Concepts

User Input

Selected Range (bipolar)



Only 25% of the available A/D range is used.

Data Acquisition Concepts

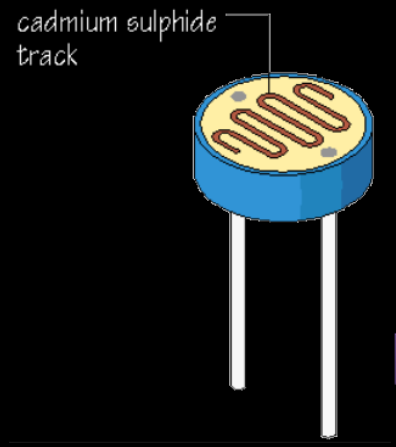
- To get the highest degree of accuracy possible out of an I/O board – try to utilize as much of the available range as possible
- Use internal or external gain selection
- Determine the maximum range that the input signal will use
- Determine if the signal is uni-polar (above zero) or bipolar (above and below zero)
- Evaluate the available gain and range combinations to select the most appropriate product

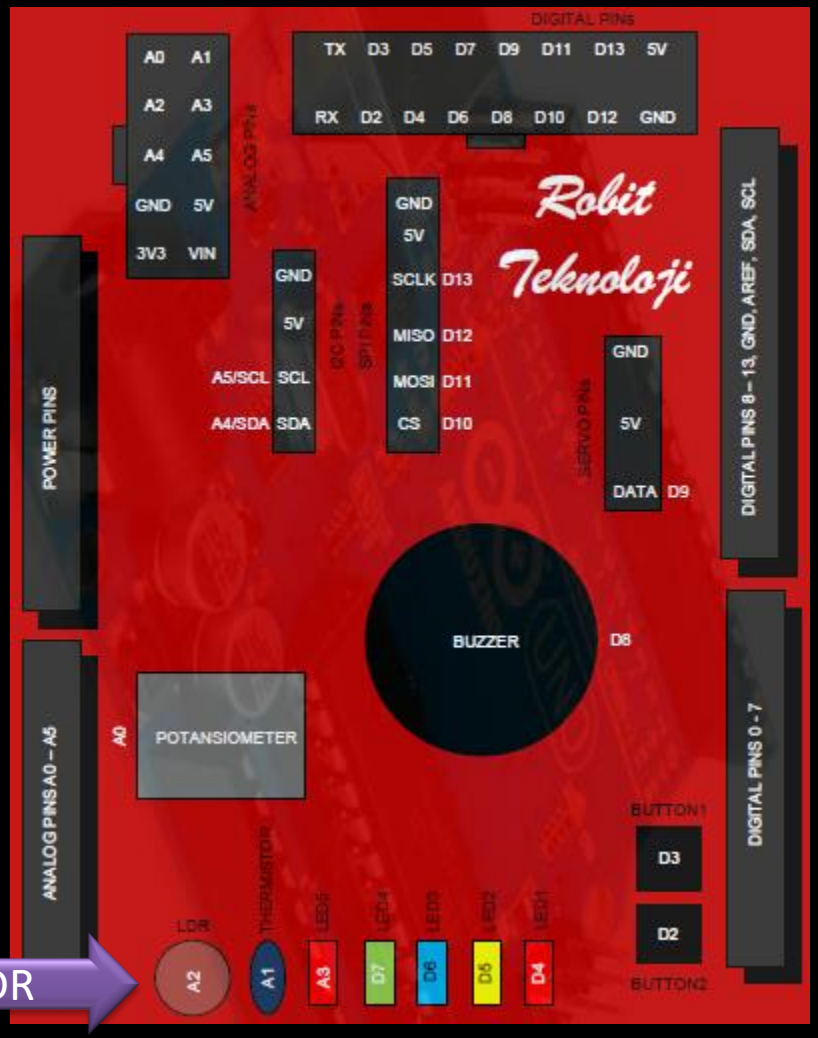
Data Acquisition Concepts

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 µs Conversion Time
- Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{cc} ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

Arduino & PYTHON

Light Dependent Resistors (LDRs) are very useful especially in light/dark sensor circuits. The resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically.





LDR

Arduino & PYTHON

```
void setup() {
// initialize serial communication at 9600 bits per sec
   Serial.begin(9600);
 }

// the loop routine runs over and over again forever:
 void loop() {
   // read the input on analog pin:
   int sensorValue = analogRead(A2);
   // print out the value you read:
   sensorValue = map(sensorValue, 0, 1023, 0, 255);
   Serial.println(sensorValue);
   delay(100);
 }
```

Arduino & PYTHON

`map(value, fromLow, fromHigh, toLow, toHigh)`

Re-maps a number from one range to another. That is, a value of fromLow would get mapped to toLow, a value of from Highto toHigh, values in-between to values in-between, etc. Note that the "lower bounds" of either range may be larger or smaller than the "upper bounds" so the map() function may be used to reverse a range of numbers, for example

```
y = map(x, 1, 50, 50, 1);
```

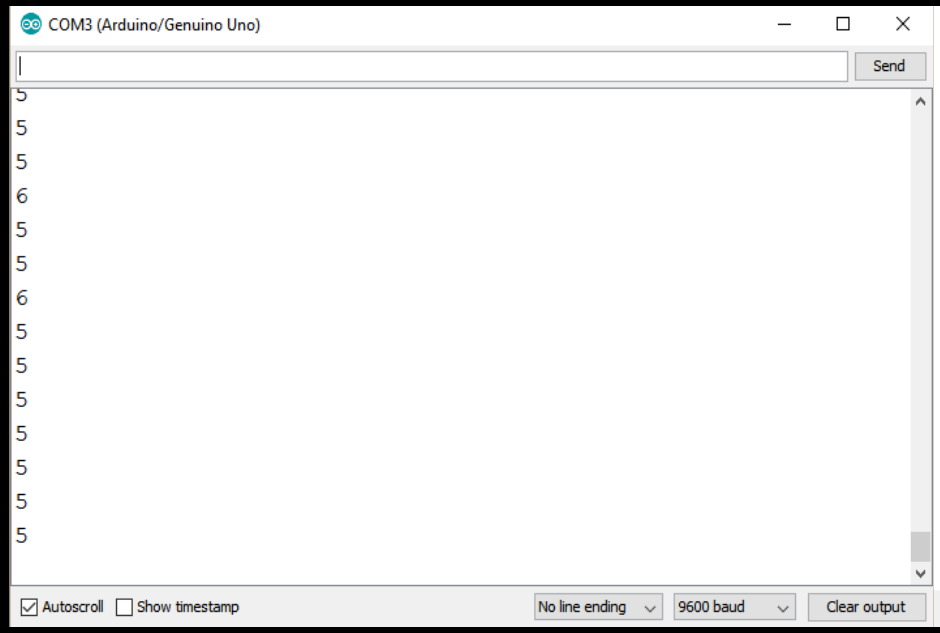
The function also handles negative numbers well, so

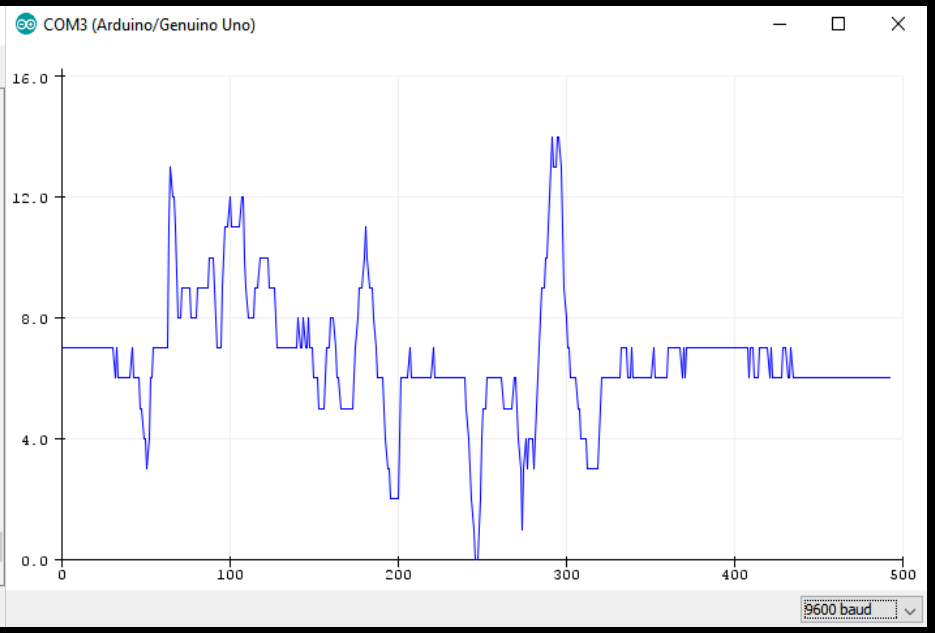
```
y = map(x, 1, 50, 50, -100);
```

is also valid and works well.

The map() function uses integer math so will not generate fractions,

Arduino & PYTHON





Arduino & PYTHON

```
import serial
from time import sleep

connected = False
# open the serial port that the Arduino is connected to
locations=['COM3','COM4','COM10','COM11','COM12','COM13']

for device in locations:
    try:
        print ("Trying...", device)
        arduino = serial.Serial(device, 9600, timeout=1)
        print ('Connected to ', arduino.portstr)
        sleep(2) # This is the minumum time for communication
        connected = True
        break

    except:
        print ("Failed to connect on ", device)
```

Arduino & PYTHON

```
while connected:
    try:
        # read the string and convert to integer
        message = int(arduino.readline())
        print(message)
    except:
        arduino.close()
        connected = False
print('Connection Closed')
```

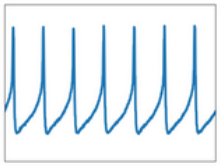
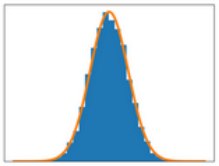
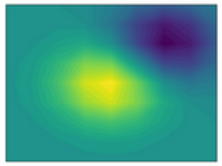
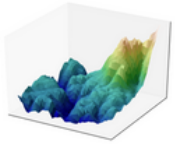
Arduino & PYTHON

https://matplotlib.org/

matplotlib
Version 3.0.3

home | examples | tutorials | API | docs »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.