2.017 DESIGN OF ELECTROMECHANICAL ROBOTIC SYSTEMS

Fall 2009 Lab 2

September 21, 2009

Dr. Harrison H. Chin

Lab Open Hours

• Tuesdays	2:30 – 5:00 (Jordan)
• Wednesdays	1:00 – 2:30 (Harrison)
• Fridays	1:00 – 5:00 (Franz)

Formal Labs

1. Microcontrollers
   - Introduction to microcontrollers
   - Arduino microcontroller kit
2. Sensors and Signals
   - Analog / Digital sensors
   - Data acquisition
   - Data processing and visualization
3. GPS and Data Logging
   - GPS receiver and shield
   - Data logging
   - Visualization of data
4. Motor Control
   - Motors
   - Encoders
   - Position control

Lab 2: Sensors and Signals

- Part 1: Do circuits 9, 10, and 11 (1:30 – 2:00)
- Part 2: Arduino mini-project: security system design (2:00 – 2:30)
- Part 3: Data acquisition (2:30 – 3:30)
- Part 4: Assemble the GPS logger shield (3:30 – 4:30)
- Part 5: Project discussion (4:30 – 5:00)

Part 1: The Arduio Kit Experiments

MITMECHE

- {CIRC01} Getting Started - (Blinking LED)
- {CIRC02} 8 LED Fun - (Multiple LEDs)
- {CIRC03} Spin Motor Spin - (Transistor and Motor)
- {CIRC04} A Single Servo - (Servos)
- {CIRC05} 8 More LEDs - (74HC595 Shift Register)
- {CIRC06} Music - (Piezo Elements)
- {CIRC07} Button Pressing - (Pushbuttons)
- {CIRC08} Twisting - (Potentiometers)
- {CIRC09} Light - (Photo Resistors)
- {CIRC10} Temperature - (TMP36 Temperature Sensor)
- {CIRC11} Larger Loads - (Relays)

Lab 2

Part 2: Arduino Mini-Project

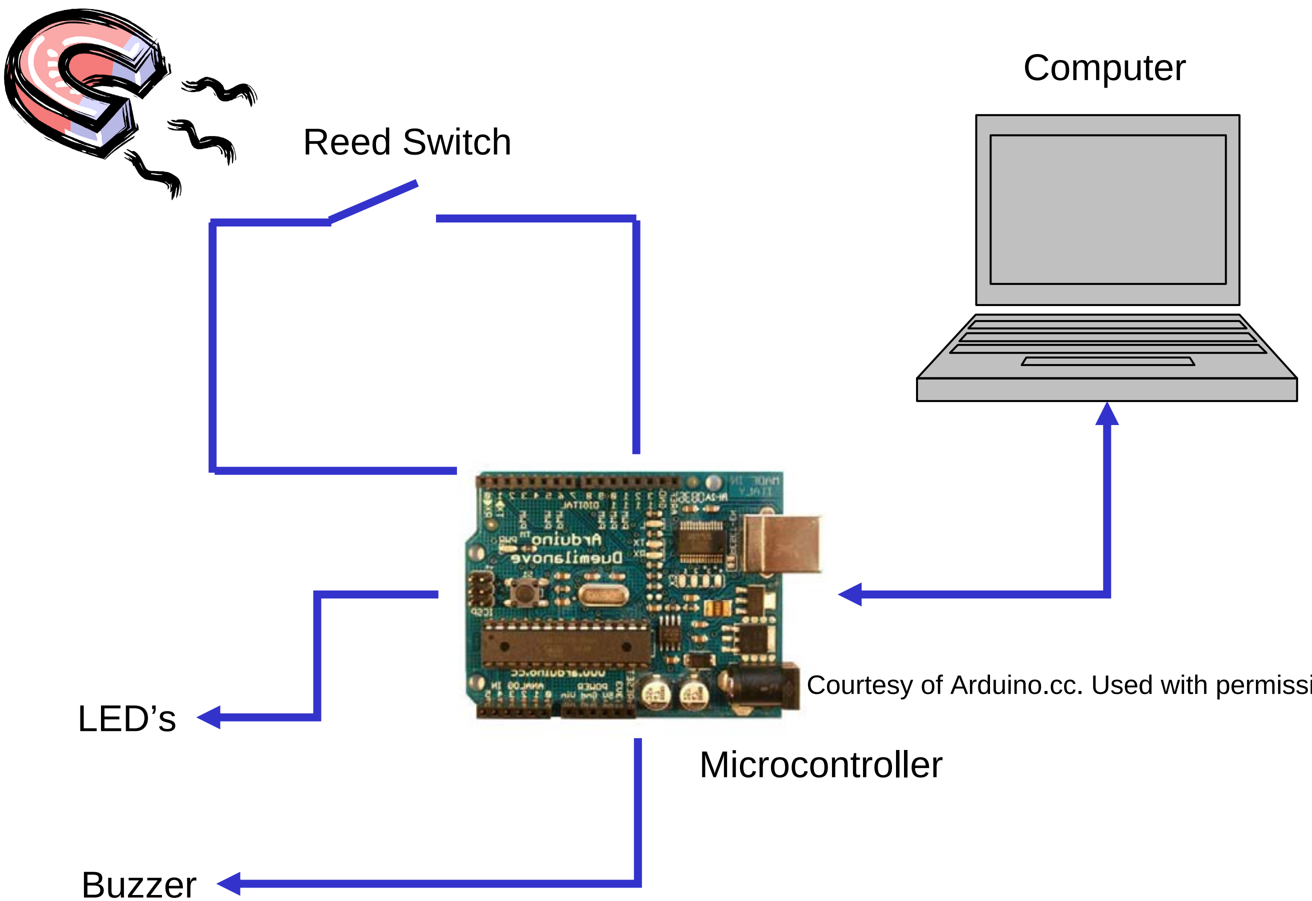
- Design a security system to sound the alarm with flashing light when a magnetic switch is tripped:
  - Use a reed switch as the digital sensor
  - Use a buzzer to sound the alarm
  - Use one or more LED's

The glass envelope is fragile, use extra care when bending the leads

The glass envelope is fragile, use extra care when bending the leads

Reed Switch (MDSR-7 by Hamlin Electronics)

Schematic of the System



Courtesy of Arduino.cc. Used with permission.

Part 3: Data Acquisition

- Use the Arduino board as a low cost data acquisition device
- Use an oscilloscope to monitor the signal
- Send a known sine wave from a function generator to one of the analog input pins on the Arduino board
- Stream the digitized data to the host PC via USB serial link for visualization and analysis

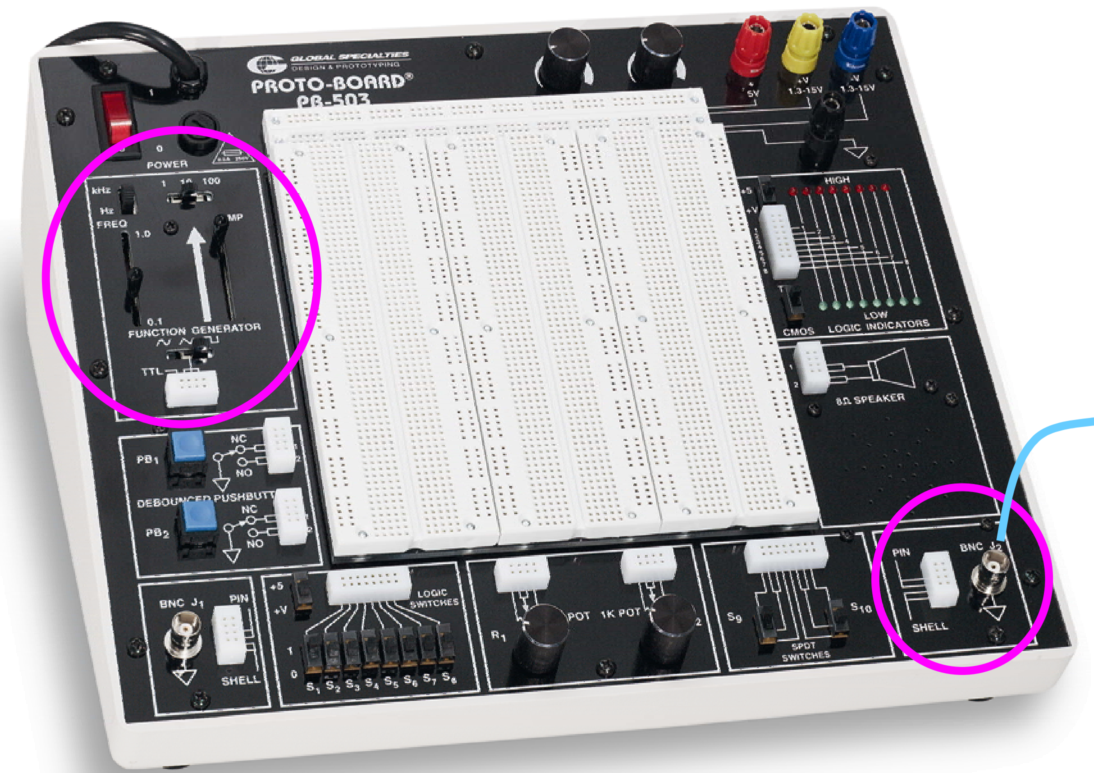
Procedures

- Grab a PB-503 protoboard, an oscilloscope, and a BNC cable from the cabinets
- Use the scope to monitor the function generator output signal
- Use the on-board function generator to generate a 10 Hz sine wave with a peak-to-peak of 4v centered at 2.5v
- Send the signal to one of Arduino's analog input pins
- Write code to read the data and send it to the PC via the USB serial port at 15 Hz, 20 Hz, 100 Hz, and 1000 Hz
- Use "RealTerm" to capture about two second worth of data
- Load the data to MATLAB
- Plot the time history of the data and compute the RMS value and the first four statistical moments
- Plot the spectrum of the data

Function Generator

- Proto-Board® Design Workstation PB-503 by Global Specialties (http://www.globalspecialties.com/pb503_spec.html)

Function generator



BNC output

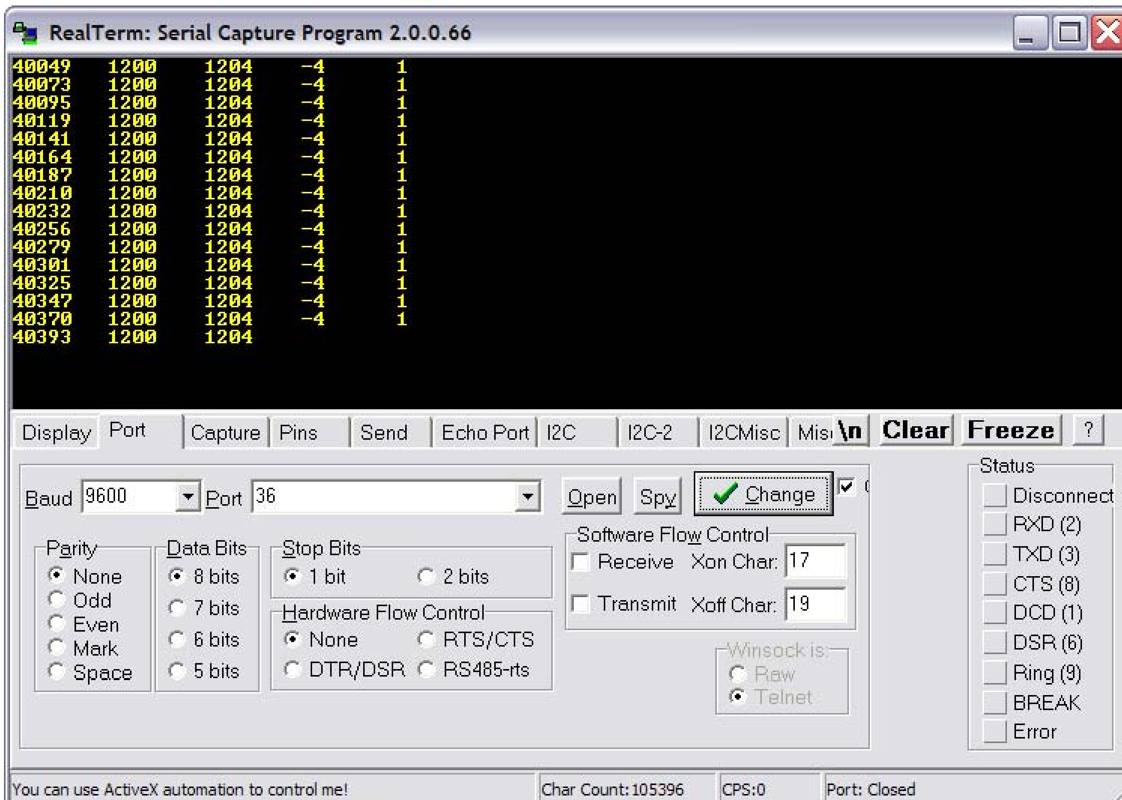
Courtesy of Global Specialties. Used with permission.

Example Code

```
/*
 * Analog Data Acquisition Test
 */

int adcPin = 0;     //
int val = 0;        // variable to store the value coming from the sensor
unsigned long time_1, time_2, delayTime, dt;

void setup() {
  pinMode(adcPin, INPUT);      // declare the adcPin as an OUTPUT
  Serial.begin(115200);        // initialize serial communication with computer
  time_1 = millis();           // read the initial time stamp
  delayTime = 10;              // delay time in ms
}

void loop() {
  time_2 = millis();            // read the current time stamp
  dt = (time_2 - time_1);       // compute delta time in seconds
  val = analogRead(adcPin);     // read the value from the sensor
  Serial.print(dt, DEC);
  Serial.print("\t");
  Serial.println(val, DEC);     // send it to the computer (as ASCII digits)
  delay(delayTime);             // define sample rate in ms
}
```

Serial Communication

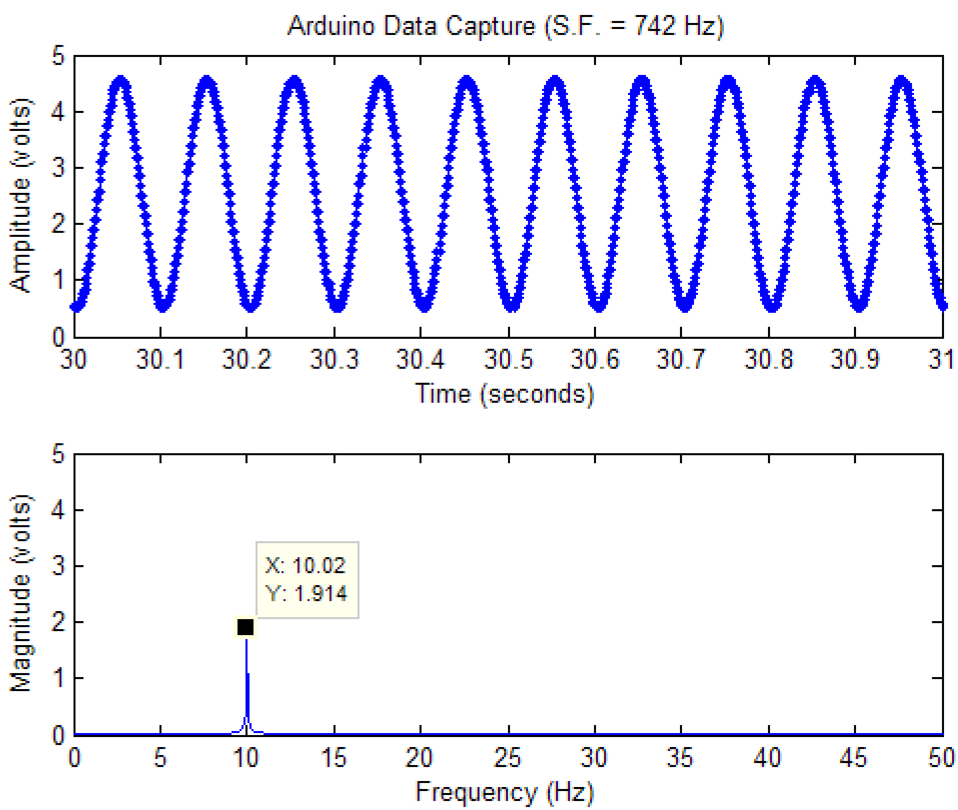
- Use "RealTerm" Serial Capture Program (http://realterm.sourceforge.net/) to monitor and capture serial data
- Import data to MATLAB for plotting



Courtesy of Simon Bridger. Used with permission.

Example Plots

- A delay of 1 ms is setup in the code but the actual sampling frequency is 742 Hz.
- This is due to the code execution time and the time to perform serial communication.
- On paper the Arduino ADC is capable of achieving 200 kHz sampling frequency.

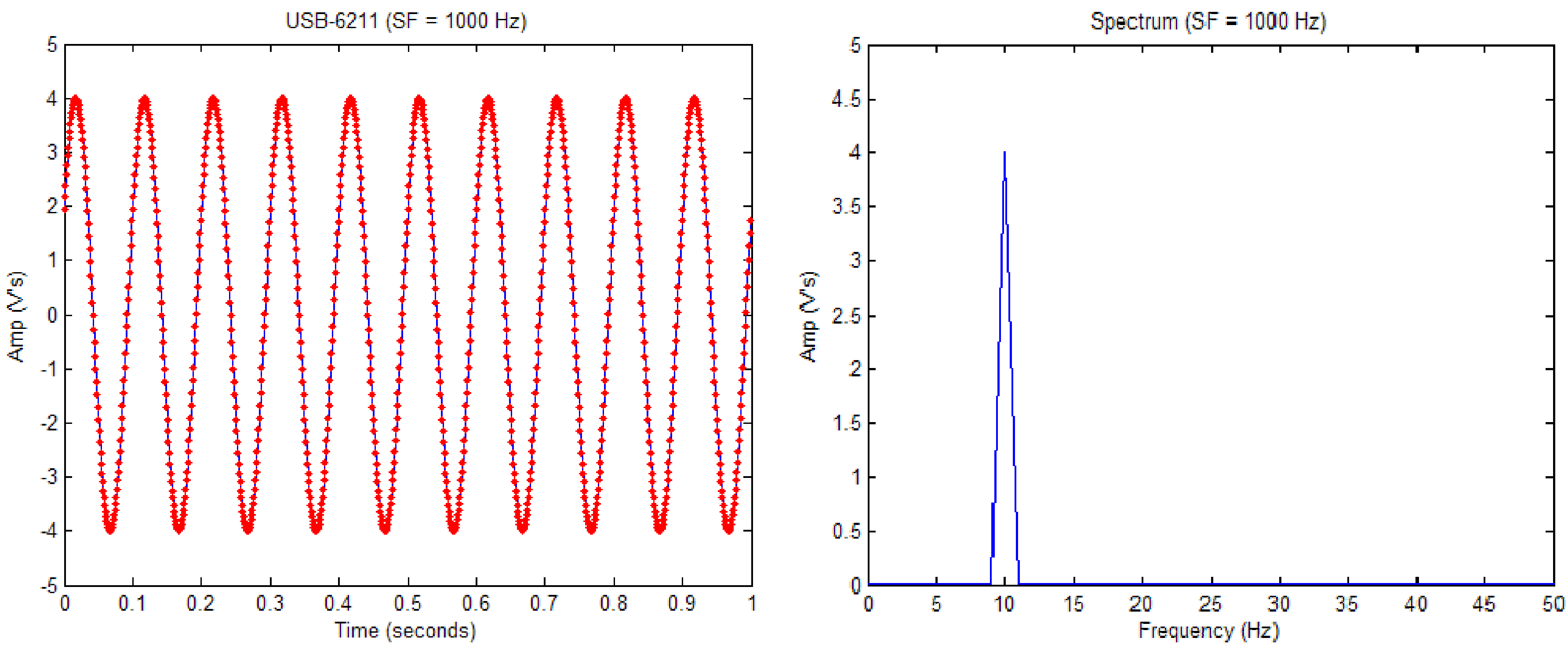



```
>> stairs(time, data);
```

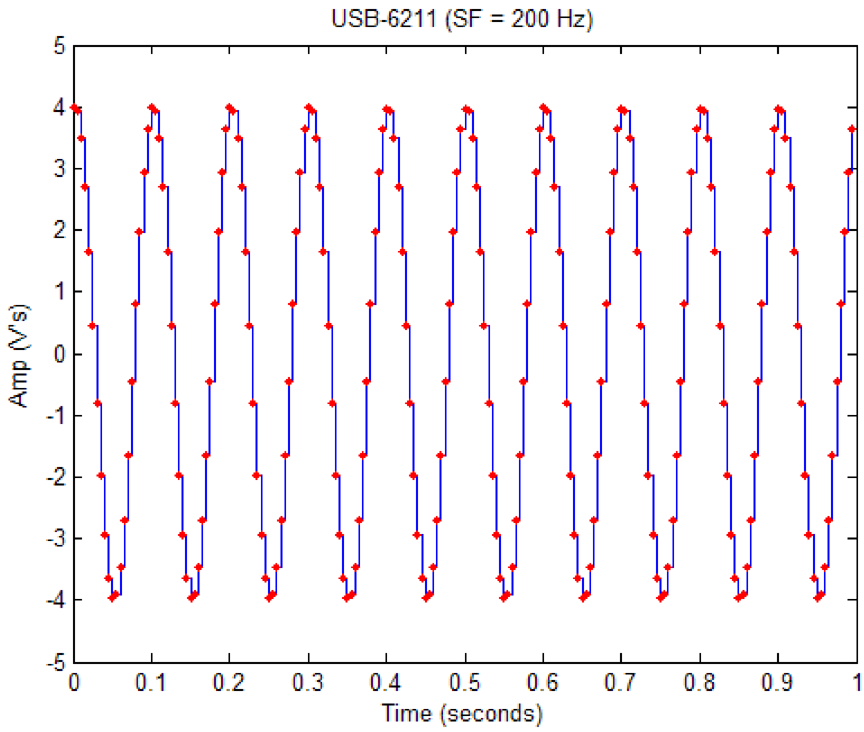
Simple MATLAB Code to Compute DFT

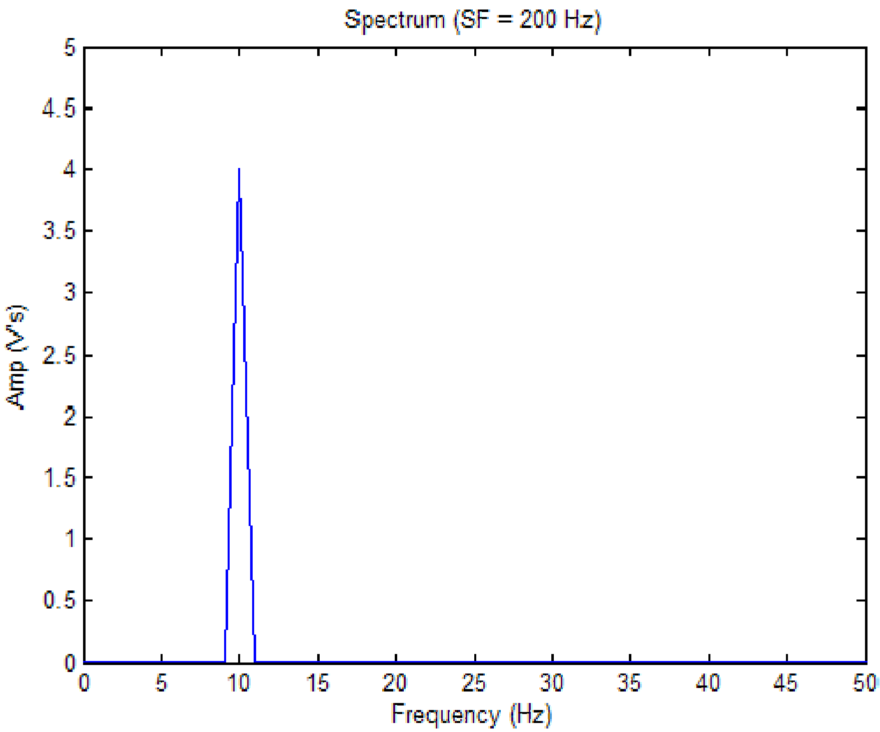
```
N = length(x);                % length of the data x
fx = fft(x(1:N))/N;           % perform FFT
Px = fx.*conj(fx);            % power density
Px((N/2+1):N)=[];             % remove the negative spectrum
fx = 2*sqrt(Px);              % one-sided spectrum
f = SF/N*(0:(N/2-1));         % form the frequency axis
```

Sample Rate = 1,000 Hz

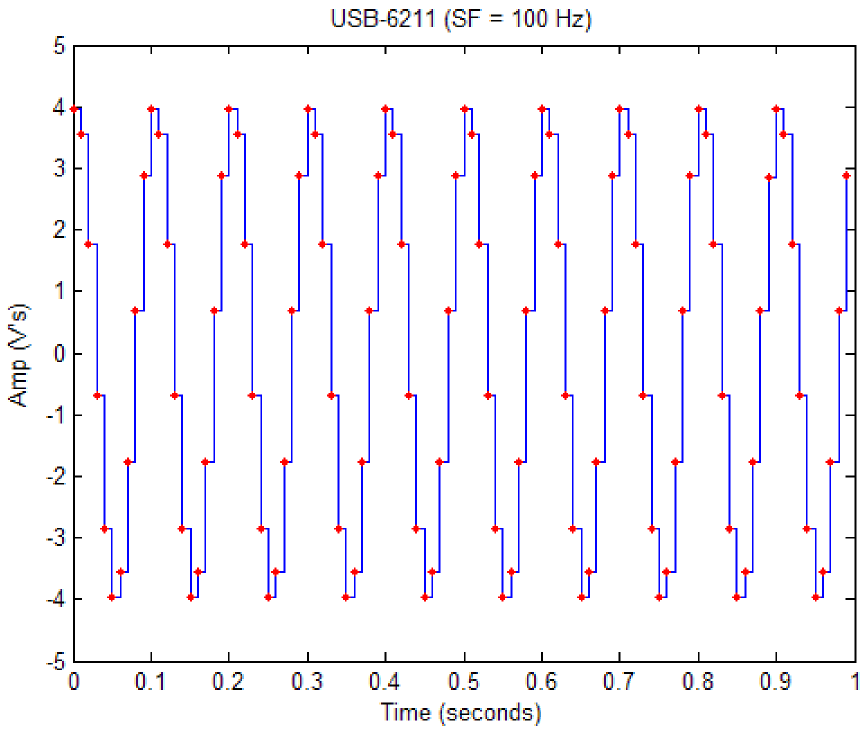


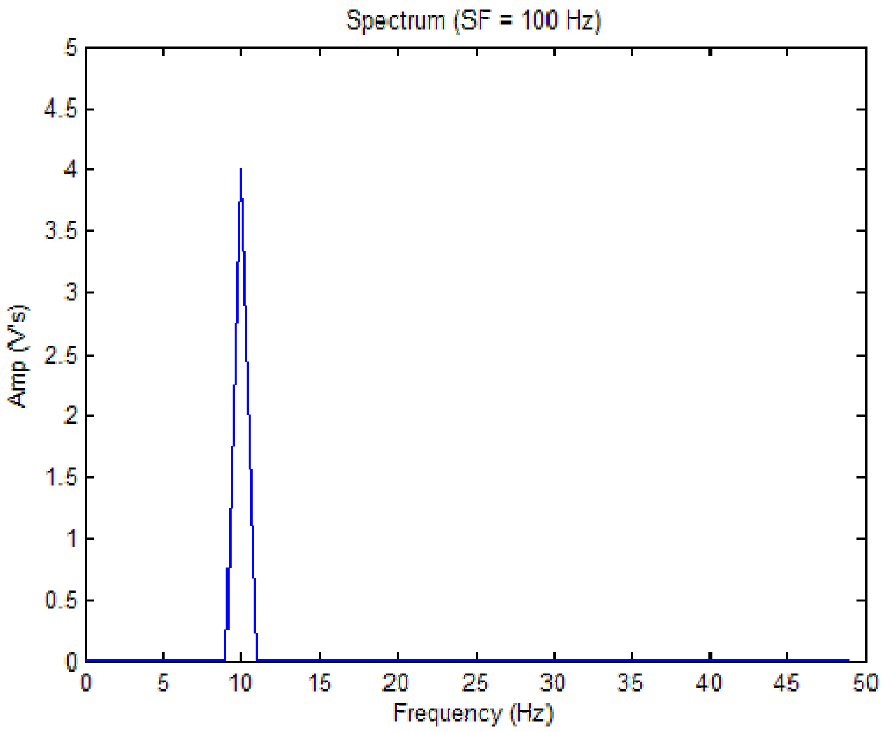
Sample Rate = 200 Hz



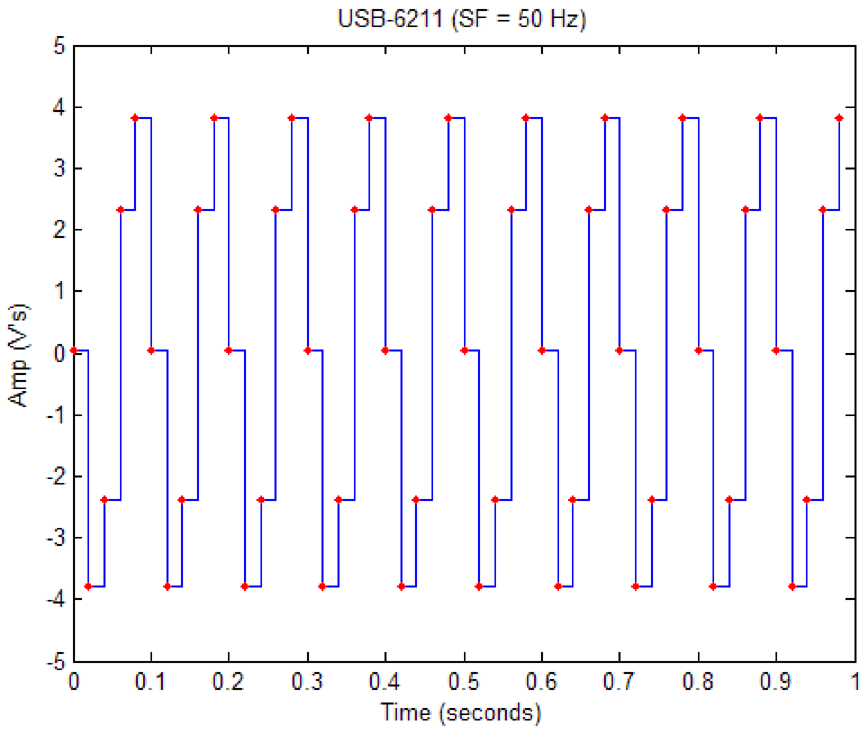


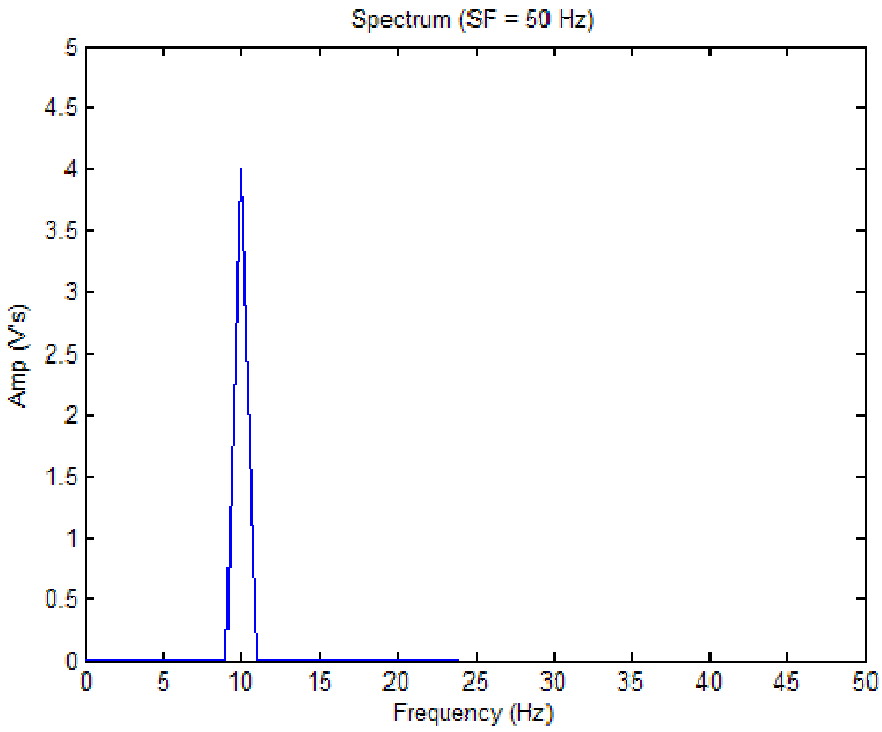
Sample Rate = 100 Hz



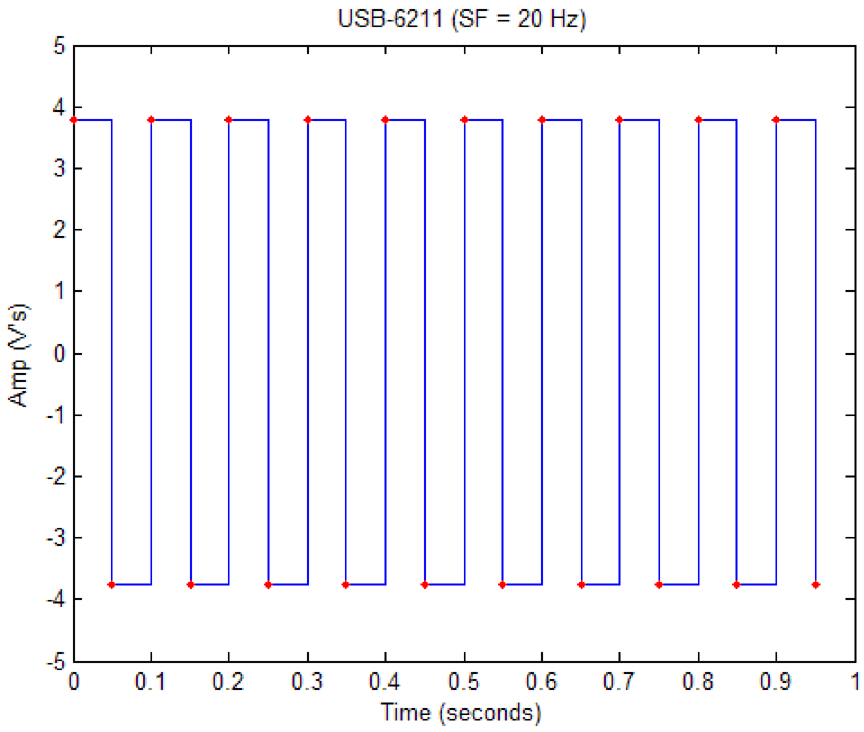


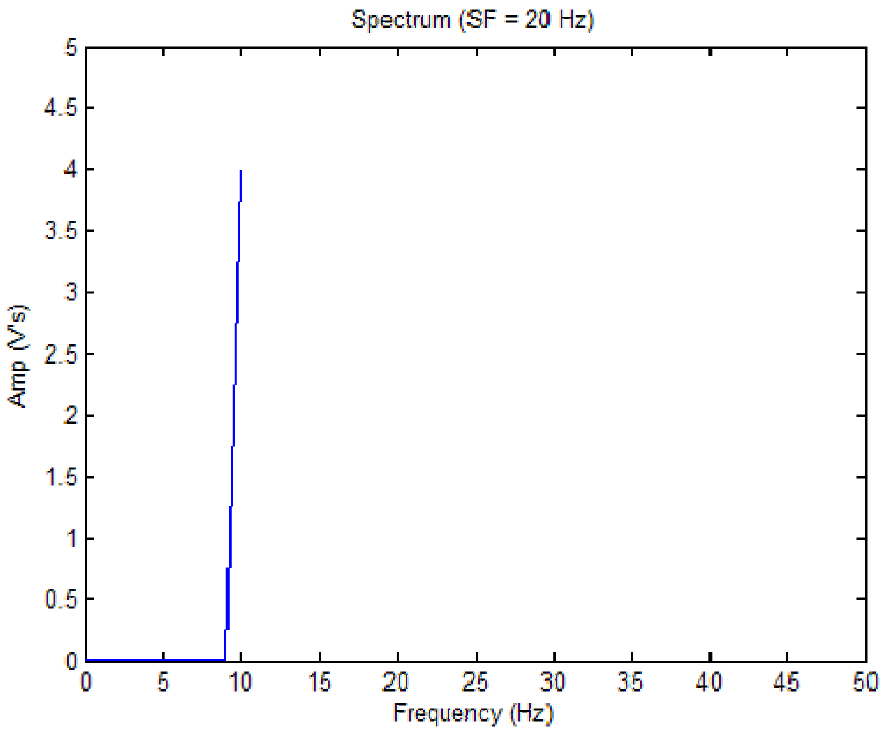
Sample Rate = 50 Hz



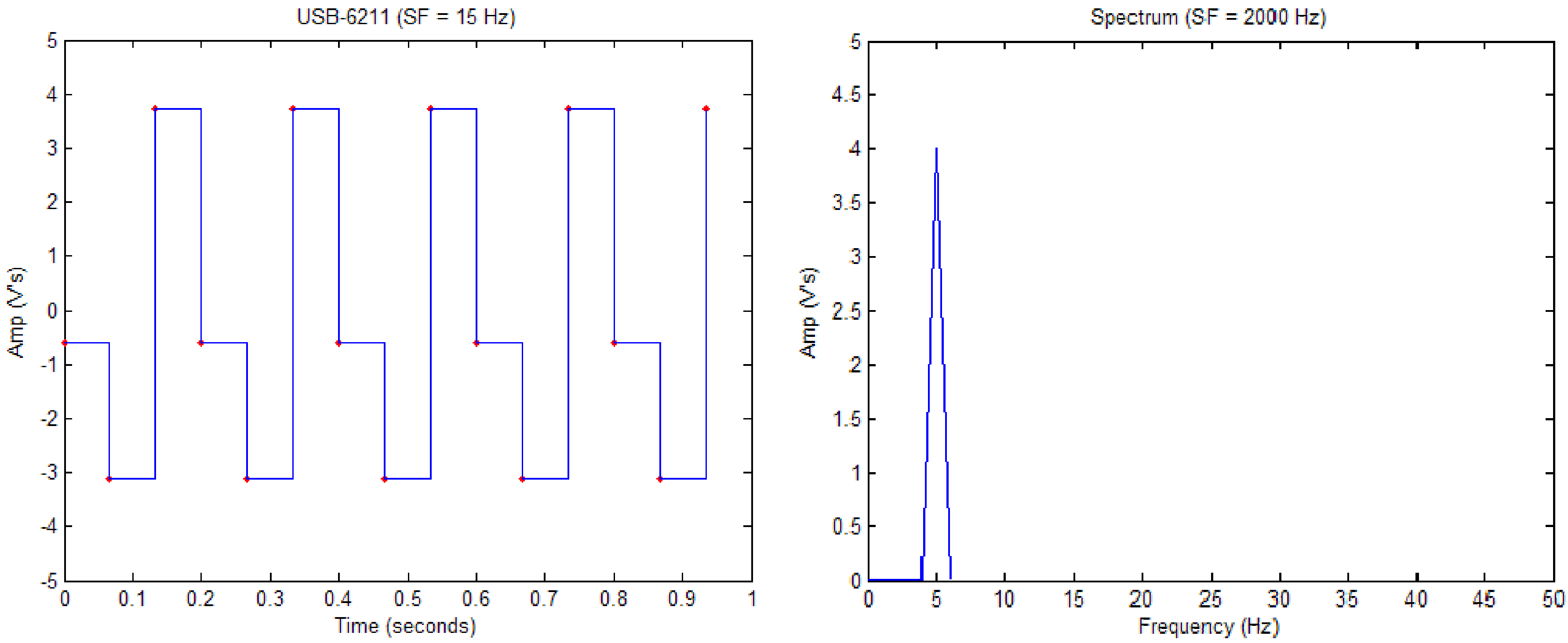


Sample Rate = 20 Hz





Sample Rate = 15 Hz



Root-Mean-Square & Statistical Moments

- The Root-Mean-Square (RMS) value of a signal represents the overall energy level of the signal. The RMS value of signal is calculated as:

$$RMS = \sqrt{E[x^2(t)]}$$

- The RMS value can also be computed from the spectrum, as the sum of the amplitudes of all frequency components in the spectrum, divided by $\sqrt{2}$.
- Statistical moments
  - 1st: Mean: location
  - 2nd: Variance (or Standard Deviation): spread
  - 3rd: Skewness: symmetry
  - 4th: Kurtosis: peakedness / flatness

Part 4: Assemble The GPS Logger Shield

- Grab a soldering iron and solder
- Power the soldering iron and set the temperature to 4
- Follow the on-line instructions on http://www.ladyada.net/make/gpsshield/solder.html to assemble the board
- Also solder the 9v battery holder
- We will test and use the board next week

Soldering Guidelines

- Wear safety glasses when soldering
- Do not touch a hot iron
- Never leave your iron turned on while unattended
- Never set the soldering iron down on anything other than an iron stand
- Use needle nose pliers, heat resistant gloves, or a third hand tool to hold small pieces
- Practice a few times if you have not done soldering recently
- Do not use excess amount of solder
- Double check the part you want to solder before you actually do it
- When done soldering, tinning the iron is required to protect the tip from oxidation thereby dramatically increasing its life

Part 5: Project Discussion

- Try out the vehicle
- Generate a rough plan (or task list)
- Identify project needs

Deliverables

- Show and explain each completed circuit to the teaching staff
- Print out a graph showing the relationship between the input voltages and the ADC values, and write down the scaling factor (V/ADC) you found
- Show your security system to the teaching staff once you are done. Hand in your code (with comments) and a simple description of your system
- Plots of time history and spectrum of the captured sine wave data at each sample rate. Write down the computed RMS value and the values of the first four statistical moments (i.e., mean, variance, skewness, and kurtosis)
- Assembled GPS logger shield and battery holder
- Show the teaching staff your lab notebook

2.017J Design of Electromechanical Robotic Systems
Fall 2009