Apparatus Competition

2012 Summer Meeting of the American Association of Physics Teachers Philadelphia, Pennsylvania

Sponsored by the

AAPT Committee on Apparatus

Prizes are generously provided by

PASCO

Apparatus Descriptions also available at
`groups.physics.umn.edu/demo/apparatus.html`

Table of Contents

* These apparatus can be built for less than $65.

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

Apparatus for Quantitative Measurement of Heat Flow in Two Dimensions

Eric Ayars, Daniel Lund, and Lawrence Lechuga

Department of Physics
Campus Box 202
California State University
Chico, CA 95929-0202

(530) 898-6967

ayars@mailaps.org

Abstract

With modern microcontroller-based sensors it is possible to measure temperature at many points simultaneously. Here we use an Arduino microcontroller to measure the temperature of a 10x10 array of points on an aluminum plate. The measurements are then sent to a computer, which generates contour plots of the temperature and allows visualization of two-dimensional heat flow in real time. The data can also be saved and compared with numeric solutions of the heat flow equation.

Construction of Apparatus:

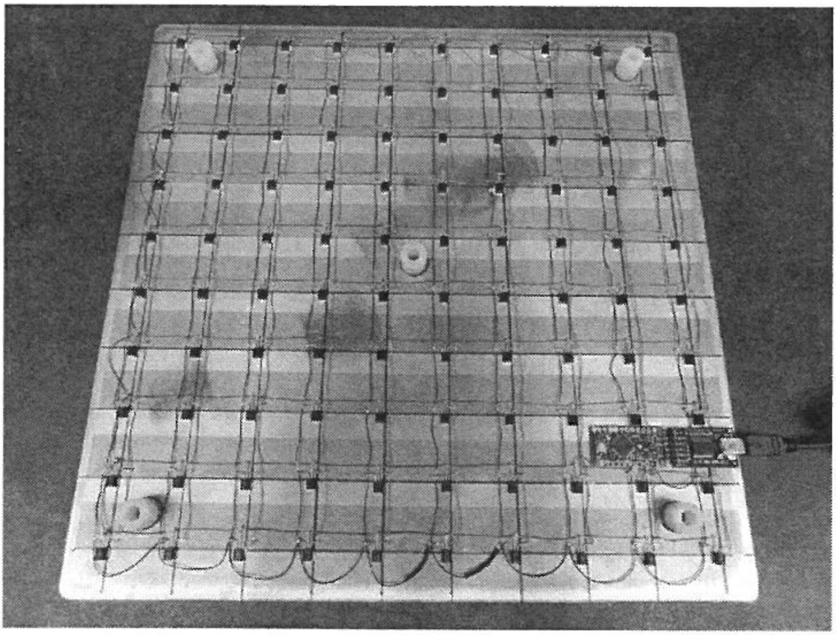
The key to the success of this apparatus is the temperature sensor: the Dallas Semiconductor DS18B20. This device measures temperature from -55°C to +125°C with 0.5°C accuracy, and with either 9-bit or 12-bit precision. It is contained in a 3-pin TO-92 package, and requires only power, ground, and data connections. The data is transferred via the 'One-Wire' protocol to a microcontroller. Each DS18B20 has a unique address, hard-coded into ROM at the factory, and can respond to either "broadcast" or "individually-addressed" commands from the microcontroller. There is no intrinsic limitation to the number of devices that can be connected to a One-Wire bus; since each device has a unique address it is possible to connect *hundreds* of temperature sensors to one microcontroller pin and still obtain data from each sensor individually.

We began by laying out a grid of sensor locations on a 30cm-square plate of 5mm-thick aluminum. We laid Kapton tape in strips under where the sensor leads would be located, to prevent the leads from shorting against the aluminum plate; then we glued each sensor to the plate using thermal epoxy.

Next, we temporarily attached each sensor to an Arduino microcontroller running FindAddress.ino. This program first determines how many devices are on One-Wire bus; then fo each device, it sends the address of that device to the Arduino serial port. A terminal emulator o the attached computer displays those addresses. By running this program with the sensors attached individually, we could then determine the hard-coded addresses of each sensor. One hundred tests later, we had a complete list of sensor addresses, ordered by their physical positic on the grid.

Once we knew the individual addresses and corresponding locations of the sensors, we permanently wired all the sensors to the Arduino. Power and ground are provided by the USB connection to the computer, and the data pins for each sensor are all connected via a single wir to one input pin of the Arduino. The data bus is also connected to power via a 1k pull-up resistor (Note: the datasheet for the DS18B20 calls for a 4.7k pull-up resistor, but we found that with 100 sensors on a single bus a 1k pull-up resistor provided more reliable operation.) The Arduino itsel is connected to the computer via a "FTDI Friend" USB-serial converter board.

The completed hardware (bottom side) can be seen below. The plastic stand-offs allow it to b placed flat on a table with the bare top side up.



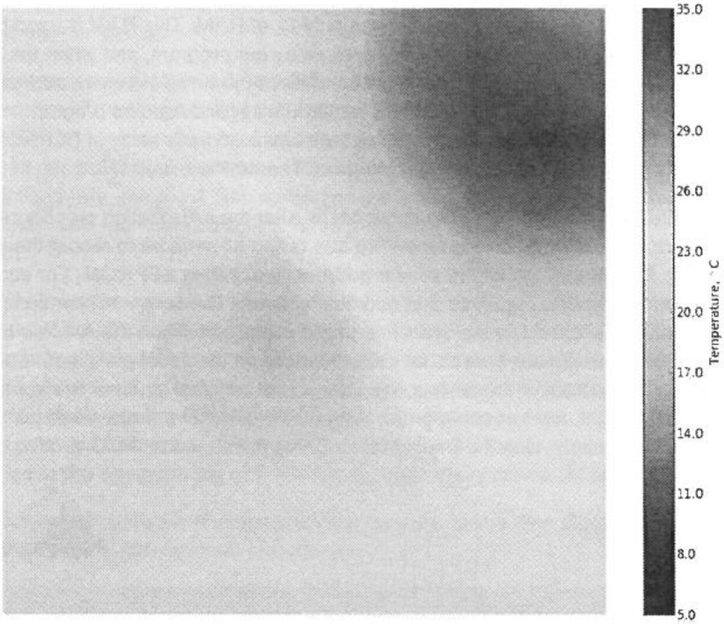
The final step in construction is to program the Arduino to measure the temperatures and send those temperatures (in order of grid position) to the computer. With this many sensors we found that we were straining the capacity of the microcontroller: although the ATmega328 chip on the Arduino board has 32k of program space, it has only 2k of RAM. This RAM is used by the serial and One-Wire communications libraries as well as by our program, and when the RAM is full the Arduino behaves erratically. Our solution was to store the array of sensor addresses in the 1k array of EEPROM on the microcontroller. This required a second Arduino program (address_storage.ino) which was run *once* to store the sensor address array in EEPROM. Once that program did its job, we uploaded our final program, ThermoPlate.ino.

The ThermoPlate.ino program operates in two steps after the initialization procedures: first, it sends out a "broadcast" message on the One-Wire bus telling all sensors to record the current temperature. Next, it goes through all the sensor addresses stored in EEPROM. For each address it sends a temperature inquiry to that address, converts the sensor response to °C, and sends the temperature as text down the serial line to the computer. When the Arduino has gotten the temperature from each sensor it sends an EOL character to the serial port, then repeats the process. That's the final output of the device: line after line of serial data, each line containing 100 temperatures in left-to-right, top-to-bottom order. With our 16-MHz Arduino, it takes just under 2 seconds to measure all points, with the limiting factor being the speed of the One-Wire bus itself.

Use of Apparatus:

The computer has to be able to listen to the apparatus. If the Arduino IDE is installed on the computer —and it will have to be, if you've been using that particular computer to program the Arduino— then this is already taken care of. But if that is not the case, you must at the very least install the "FTDI virtual com port" software on your computer so that it can communicate with the USB-to-serial converter chip located between the USB cable and the Arduino. Once this is accomplished, you must determine the virtual port to which the device is attached. This varies by operating system and computer hardware: on Mac it will be something like "/dev/tty.usbserial-01872J", on Linux "/dev/ttyUSB0" or similar, and on Windows "COMx" where x is an integer.

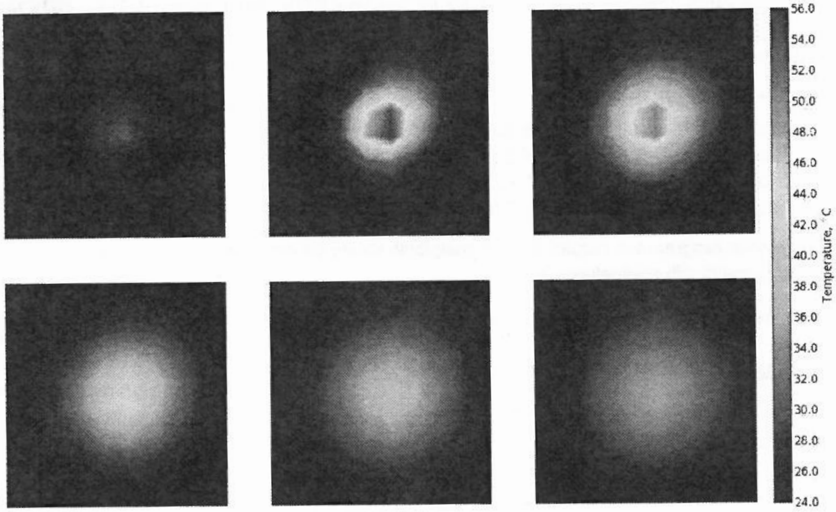
Once communication is established, it is necessary for the computer to make sense of the incoming data. Since the incoming data is simply text, there is a wide range of possible tools to use for this purpose, ranging from LabVIEW to IDL to C++. For simplicity, we chose to write our program, TempMovie.py, in Python. This program captures each line of data coming in the serial port, splits the line into individual temperatures, rearranges those temperatures into a grid, and then draws a filled contour plot based on that grid data. One can set the temperature scale in the program to cover the expected range of values, and if desired the program will save the raw data for later analysis. The figure below shows a single frame of output of TempMovie.py when a small bag of ice was placed at the upper right and a hot soldering iron was placed near the center.



The image changes as the plate moves towards its new equilibrium state, and one can observe the heat flow in real time. For the sequence of images shown below, the plate was hit with a flame from a propane torch, which was removed after the second image in the sequence. Time-dependent thermal data such as this can be compared to numeric solutions of the Heat Equation,

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0$$

(Aside: These temperature sensors can detect changes as small as 0.1°C. If you look closely at all of these images, you can see that an area near the lower left corner of the plate is about 0.3°C warmer than the rest of the plate, due to the power dissipation of the microcontroller.)



This technique of using multiple DS18B20 sensors controlled by a single Arduino can be used for one-dimensional experiments as well. Twenty-five sensors, evenly spaced along a meter of copper pipe, would make a wonderful replacement for the typical half-dozen thermocouples we've all used and hated as undergrads!

Summary of software and hardware sources:

The simplest Arduino to use for this apparatus is the latest hardware version, the "Arduino Uno", which is available from many sources including the superbly helpful people at Adafruit.com:

https://www.adafruit.com/products/50

If you go this route, you will not need a separate USB-to-Serial adaptor. For our apparatus, we used an Arduino Pro Mini from Sparkfun.com in conjunction with an "FTDI Friend" USB-to-Serial adaptor, just because we had extras available in the lab.

http://www.sparkfun.com/products/11113
https://www.adafruit.com/products/284

The DS18B20 sensors are available from most electronic suppliers; Jameco currently has them for just over $2 each in bulk:

http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_865936_-1

The thermal epoxy we used was "Arctic Silver Alumina", available from Newegg.com and other computer parts suppliers.

The FTDI virtual com port driver is available for free from FTDI's website at

http://www.ftdichip.com/Drivers/VCP.htm

Note that if you are using the latest Arduino Uno, you will *not* need this driver.

Our software is available here:

http://physics.csuchico.edu/~eayars/code/FindAddress.ino.html
http://physics.csuchico.edu/~eayars/code/address_storage.ino.html
http://physics.csuchico.edu/~eayars/code/ThermoPlate.ino.html
http://physics.csuchico.edu/~eayars/code/TempMovie.py.html

The python program will require that the matplotlib library be installed:

http://matplotlib.sourceforge.net/

A full movie of the propane torch on the aluminum plate (sped up by a factor of 4) can be viewed here:

http://physics.csuchico.edu/web/ayars/torch_movie.mov

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

Ammeter Protection Circuit

Dan Beeker

Physics Department, Indiana University
727 E 3rd Street
Bloomington, IN 47405

(812) 855-5903

debeeker@indiana.edu

Abstract

Low current analog ammeters are valuable pedagogical tools. But they must be protected from student abuse. Rather than use fuses one can use a mosfet with low internal on resistance to protect the meter. Because of the high impedance characteristics of the circuit, battery life is very long. Additional features that can be incorporated are beepers and leds to announce when a meter is connected incorrectly.

Construction of Apparatus:

The circuit can either be hand wired or built on a circuit board. Printed circuit board files are available on request. Two identical circuits protect the meter from over current and reverse current. Depending on the meter range and internal impedance, current trip level is selected by resistors.

Use of Apparatus:

The circuit has been tested on ammeters with full scale readings of 1 amp or less. By using 9V batteries, the circuit is protected against overvoltage to greater than 15V. The circuit prevents users from applying excessive currents to panel ammeters. The apparatus consists of two essentially identical circuits. Each circuit monitors and controls current flow through the meter.

A mosfet transistor is biased in the on state by 9 V batteries. Because of the high impedance between the gate and source/drain of the mosfet very little current flows to maintain the mosfet in the ON (conducting) state. In their on state, each transistor has a resistance of about 65 milliohm. This is considerably less than the impedance of most ammeters. An opamp with hysterises monitors the current flow across the meter and the mosfets. When the current exceeds the trip current (set by R3 and R5) the opamp turns the mosfet off preventing current flow through the meter.

Breaking the circuit or otherwise bringing the current to near zero will allow the circuit to reset.

Additional information can be found at http://www.indiana.edu/~hisci/meterprot/Meter_Protector.html

[Hard copies of bom, photos and schematics, theory of operation and selection of components will be made available at the summer meeting. They will also be posted on the above web site.]

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

Open Source Physics Laboratory Data Acquisition System

Zengqiang Liu

720 4th. Ave. S. WSB311

St. Cloud MN 56301

320-308-3154

zliu@stcloudstate.edu

Abstract

Open source physics laboratory data acquisition system (OSPL DAQ) is an electronics hardware platform supplement/alternative to PASCO or Vernier interfaces that can be used with sensors/actuators in and outside lab rooms to perform experiments, process data, and even make your own creative designs. Both circuit board and firmware are open source so anyone may build it at a cost less than $65, and make any changes to suit their own project needs.

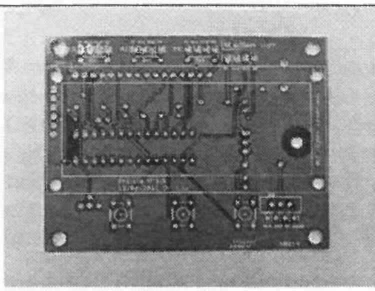
Construction of Apparatus:

The complete DAQ inside a project box with DIN-5 sockets is shown in fig.1F. Purchase a kit from inmojo.com so you can build the DAQ according to the video instructions. You will also need a project box and connectors that will work with your sensors. We chose DIN-5 sockets since we have several types of older Vernier sensors that have DIN-5 connectors. Also a power barrel and AC adapter or 9V battery is needed to power the system. An USB adapter is needed to program it but any number of DAQ can be programmed by the same USB adapter so only one is needed.

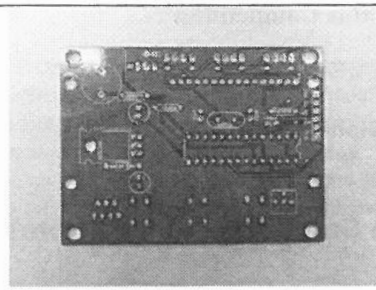
Item	Qty	Unit price	Subtotal
INMOJO Open source physics laboratory DAQ	1	$50.00	$50
Moderndevice USB bub V2	1	$14.00	$14

Table 1. Parts list of the apparatus.

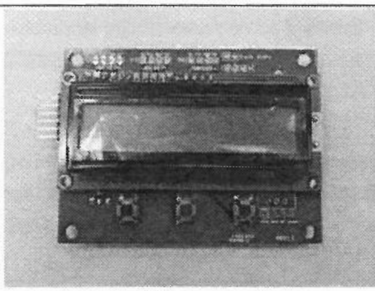
A)



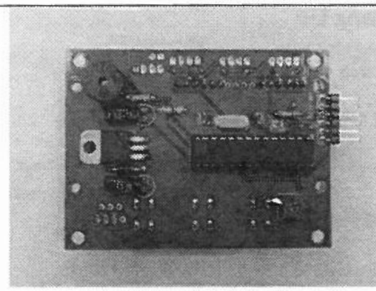
B)



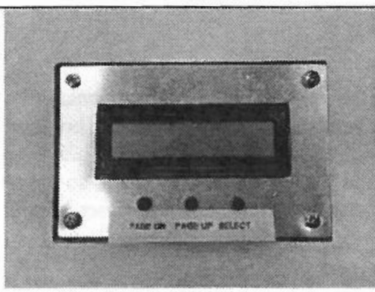
C)



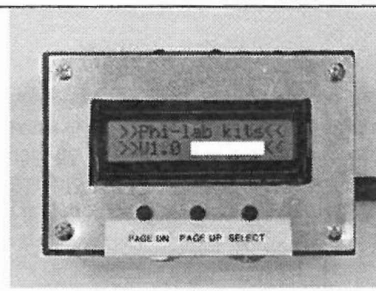
D)



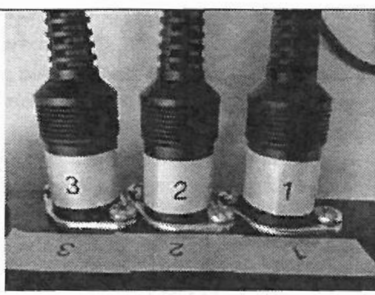
E)


PAGE DN PAGE UP SELECT

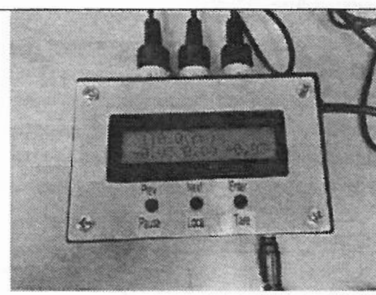
F)


>>Phi-lab kits<<
>>V1.0
PAGE DN PAGE UP SELECT

G)


3
2
1

H)



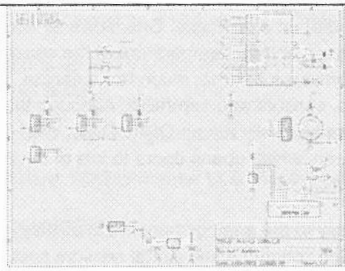
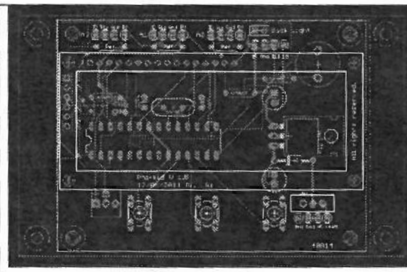
I)		J)	

Figure 1. Assembling the OSPL and its features.

1) Obtain all parts listed in table 1.

2) Assemble the OSPL DAQ according to online instructions and the video tutorial. Any AC adapter with 7-12V 200mA DC output will be sufficient to power the DAQ. The front and back sides of the circuit board are in fig.1-A-B. An assembled kit is depicted in fig.1-C-D.

3) Add power barrel, proper sockets and project box to the OSPL DAQ that will be connected to the sensors you desire. The 5-pin DIN sockets were used in the unit since we have older Vernier sensors that use 5-pin DIN plugs. Newer Vernier gauges use BTD connectors. We also added two ¼" stereo jacks for future photogates and PASCO sonic rangers.

4) Load a sample program to the DAQ box with the USB Bub II via Arduino IDE (code in appendix II).

5) Make your measurement either with the on-board display or with a PC.

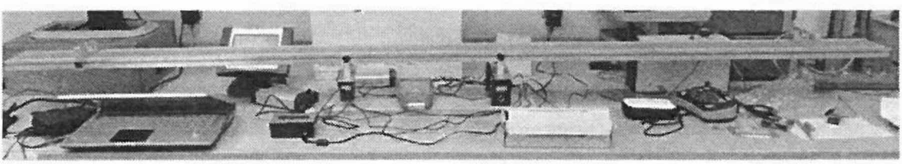


Figure 2. An apparatus designed with OSPL DAQ, the smart track.

Use of Apparatus:

This apparatus has endless usages in and outside a lab room. You may use this DAQ similarly to a PASCO or Vernier interface. You may also design your own apparatus with it, such as a smart track (fig.2) that reads force gauges and displays the location of a cart on a track with the solution to a 1-D equilibrium problem. Commercial interfaces often only provide data acquisition but lack any programmable logics that can drastically enhance student experience in labs and interest in experimental physics. One simple example of programmable logic shown at

AAPT 2011 summer meeting Vernier parlor was a "candle" on a cup cake. One blows air across the "candle", made with an LED and a temperature sensor, and the temperature at the sensor drops and the LED is programmed to turn off at such temperature drop, much like a candle. Such simple logic was made possible with Vernier's interface, sensors and LabVIEW. Although this is a solution, it needs a whole PC to run this logic, which can be easily run on OSPL DAQ. With OSPL DAQ, one can make this setup portable, small, and cheap, which opens doors to lots of fun and educational projects that can't be anchored down by a PC.

One may connect any Vernier analog sensors directly to the analog channels to display, record, process data or relay data to a PC. One may also connect some digital sensors such as photogates, sonic rangers with the DAQ, or even turn the DAQ into an emulated Vernier digital sensor such as a sonic ranger in the smart track project. This device enhances the existing lab data acquisition system and sometimes replaces the existing one. Programming this apparatus is also extremely easy. The following are five lines of code demonstrating the five steps to use a Vernier analog sensor:

Line 1: acquire data:

reading=analogRead(channel);

Line 2: scale data according to user's manual:

result=a*reading+b;

Line 3: clear on-board LCD:

lcd.clear();

Line 4: display result:

lcd.print(result);

Line 5: pause momentarily for user to read the result:

delay(200);

If one replaces line 4 with Serial.print(result), then the result will show up on a PC when the USB adapter is connected. Programming other DAQs are not as simple. The schematic design and printed circuit board design are depicted in fig.1I-J while original files can be requested if interested. Appendix I lists the specifications of the DAQ while Appendix II is a sample program.

Appendix I: specifications.

Microcontroller: ATMEGA328P-PU with Arduino Uno optiboot bootloader

Program memory: 32KB, roughly stores up to 3,000 lines of C/C++ code.

Variable memory: 2KB, roughly stores up to 1,000 data points.

EEPROM: 1KB, roughly stores up to 500 data points.

Available pins: 6 analog pins and 2 digital pins. Analog pins can also be used as digital pins.

Supports up to 3 Vernier analog sensors with autoID (resistor-based) or up to 6 analog sensors without autoID

Supports up to 8 digital inputs or outputs for photo gates or 4 sonic rangers.

On-board 16X2 character LCD monitor with back light

On-board speaker

LCD back light jumper to disable back light to preserve battery power

Programmer port for program upload and data link with a PC/Linux/Mac

3 Buttons for interactions with the user

Regulated 5V power source

Fits inside a project box as small as 4"*3"*2"

Massive amount of library and sample codes as templates to start a project or load pre-written code for specific tasks.

Appendix II: One sample program to be loaded via Arduino IDE version 1.0 or higher

```
#include <LiquidCrystal.h>     ///< Include the liquid crystal library

//These describe how the LCD is connected to the development kit and need not be changed.
#define LCD_RS 2           ///< Arduino pin connected to LCD RS pin
#define LCD_EN 3           ///< Arduino pin connected to LCD EN pin
#define LCD_D4 4           ///< Arduino pin connected to LCD D4 pin
#define LCD_D5 5           ///< Arduino pin connected to LCD D5 pin
#define LCD_D6 6           ///< Arduino pin connected to LCD D6 pin
#define LCD_D7 7           ///< Arduino pin connected to LCD D7 pin

#define lcd_rows 2          ///< Specify the height of the LCD.
#define lcd_columns 16        ///< Specify the width of the LCD.

LiquidCrystal lcd(LCD_RS,LCD_EN,LCD_D4,LCD_D5,LCD_D6,LCD_D7);  ///< Create the lcd object

void setup()
{
   lcd.begin(lcd_columns, lcd_rows);    ///< Initialize the lcd object.
}

void loop()
{
  int reading=analogRead(2); // Read channel 2
  float result=(reading*5.0/1024)*(-4.9)+12.25; // Convert reading to physical quantity
  lcd.setCursor(0,0); // Set where to display result
  lcd.print(result); // Display result
  delay(200); // Delay for human eyes to respond.
```

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

Smart Track

Zengqiang Liu

720 4th. Ave. S. WSB311

St. Cloud MN 56301

320-308-3154

zliu@stcloudstate.edu

Abstract

We used open source physics laboratory hardware to construct a "smart track" based on the equilibrium in order to demonstrate equilibrium and measure the location, velocity, and acceleration of a cart on the track. This apparatus can be used to explore a lot of concepts in mechanics. It works as a standalone unit and can also be plugged into a Vernier interface to emulate a sonic ranger, with accuracy comparable to a sonic ranger.

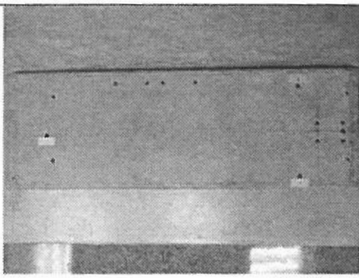
Construction of Apparatus:

The complete apparatus (fig.2) consists of: PASCO track, collision cart, three Vernier dual-range force sensors, one open source physics laboratory (OSPL) data acquisition system (DAQ), and optionally a Vernier LabQuest handheld DAQ and necessary cable if one intends to hook the setup directly to an interface and use it as a sonic ranger.

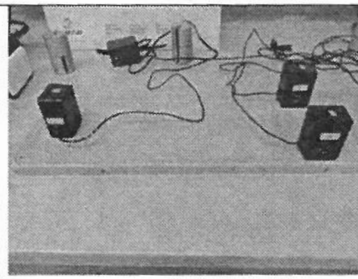
Item	Qty	Unit price	Subtotal
PASCO 1.2 m PAScar Dynamics System	1	$289.00	$289
Vernier force gauge	3	$109.00	$327
Vernier LabPro/CBL2 digital cable	1	$5.00	$5
INMOJO Open source physics laboratory DAQ	1	$50.00	$50
Moderndevice USB bub V2	1	$14.00	$14

Table 1. Parts list of the apparatus.

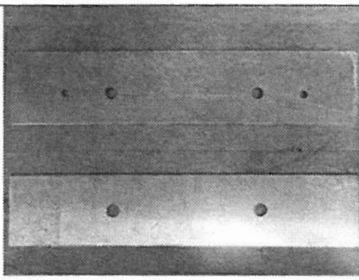
A)



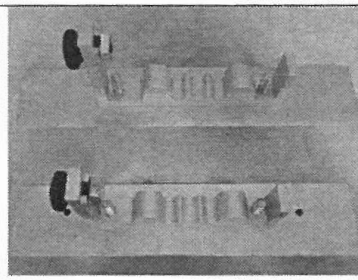
B)



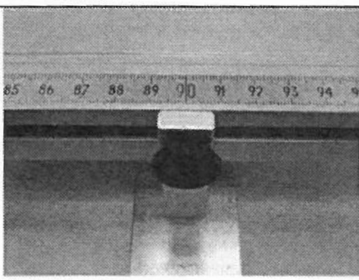
C)



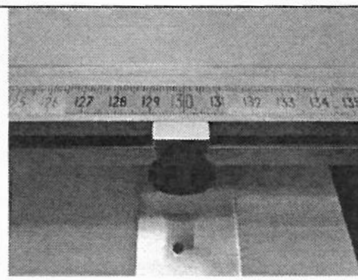
D)



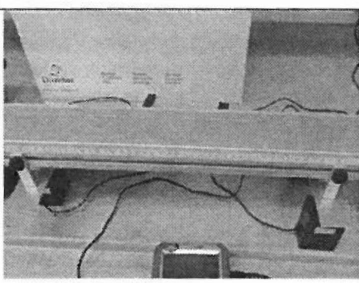
E)



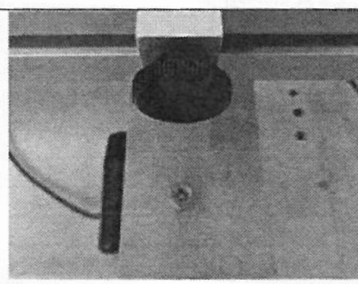
F)



G)



H)



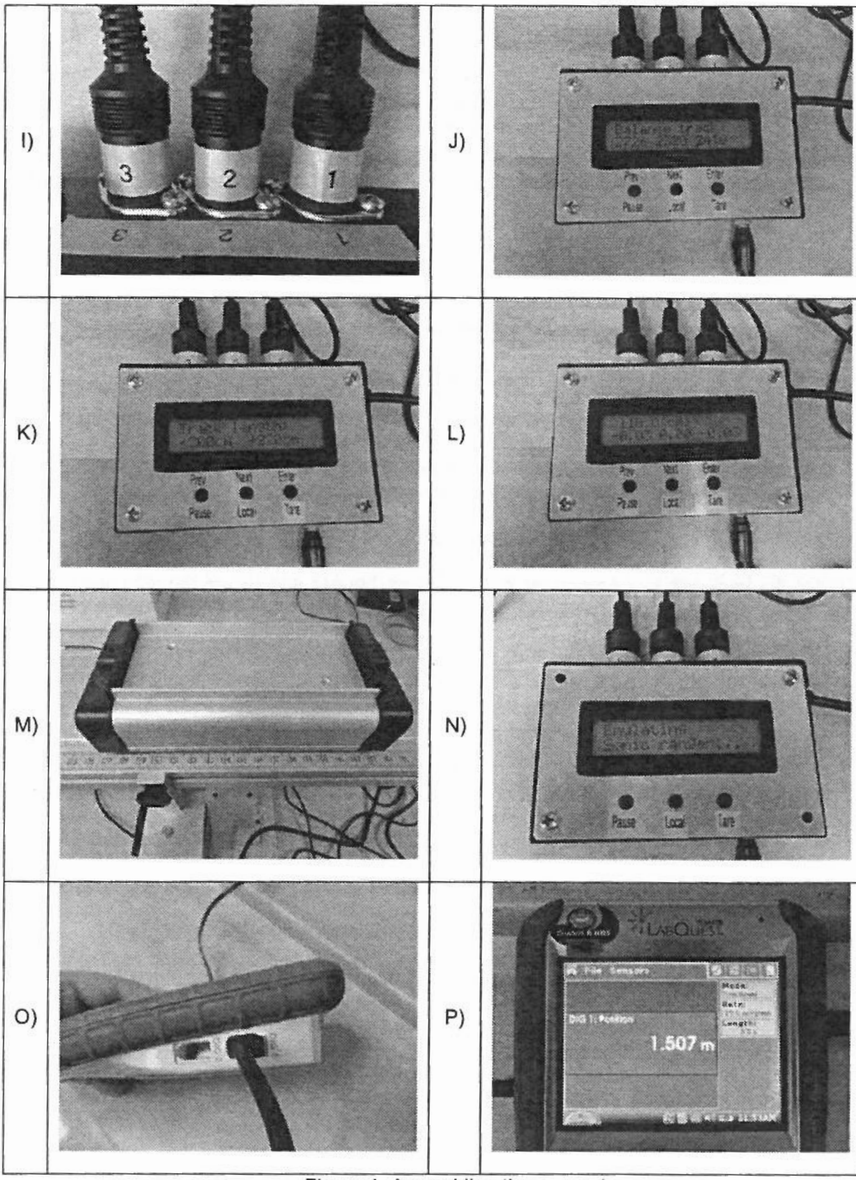


Figure 1. Assembling the apparatus

6) Obtain all parts listed in table 1.

7) Assemble the OSPL DAQ according to instructions and the video tutorial. Any AC adapter with 7-12V 200mA DC output will be sufficient to power the DAQ. add proper sockets to the OSPL DAQ that will be connected to the force gauges. The 5-pin DIN sockets were used in the demonstration unit since we have older force gauges that use 5-pin DIN plugs. Newer Vernier gauges use BTD connectors. Connect a Vernier BTD wire to the OSPL if one wants to emulate a sonic ranger.

8) Load the program to the DAQ box with the USB Bub II via Arduino IDE (code in appendix II).

9) Make a bottom plate (fig.1A-B) to secure the force gauges to it. Simply placing the force gauges on a flat surface will result in slightly less accuracy.

10) Make two adapters to secure the force gauges to the PASCO track feet (fig.1C).

11) Replace the original feet from PASCO track foot supports with the adapter plates (fig.1C).

12) Secure the foot supports at two symmetrical locations on the track 40cm apart, 30cm and 70cm for a 1.2m track, or 90cm and 130cm for a 2.2m track (fig.1D-F).

13) Notice the right plate has two additional smaller holes (fig.1F).

14) Place the track with foot support on top of the force gauges and secure the right plate with the smaller screws(fig.1G-H).

15) Connect the force gauges to the DAQ (fig.1I) and power up the box (fig.1J).

16) Press the "Tare" to set the balance, without a cart on top of the track.

17) Select the length of the track with Prev/Next and Enter (fig.1K).

18) The DAQ will display half the length of the track when there is no cart on the track (fig.1L).

19) Place a cart on the track, make sure its wheels sit inside the grooves on the track (fig.1M).

20) Now you can read the position of the cart on the DAQ screen. Press pause to pause.

21) If you want to emulate a sonic ranger, press Local to enter emulation mode (fig.1N).

22) Connect the optional Vernier cable to Vernier LabQuest channel "DIG 1" (fig.1O).

23) Read distance from the LabQuest (fig.1P). Notice the distance is offset by 1m since a sonic ranger can't report 0m distance.

24) To return to local mode, press Local to exit emulation mode and disconnect from LabQuest.

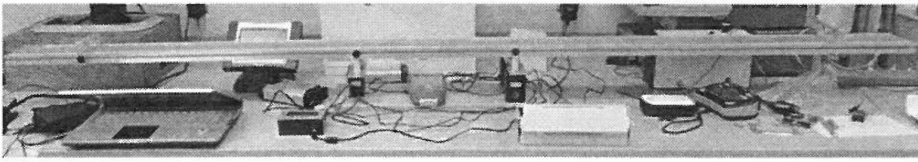


Figure 2. Complete apparatus

Use of Apparatus:

This apparatus not only demonstrates the concept and application of translational and rotational equilibrium, it also acts as a device to report position of a cart, much like a sonic ranger. The concept of equilibrium in one dimension involves two equations, the balance of forces, and the balance of torques. By using two force gauges (three in practice for stability), one can calculate two unknowns, i.e. the weight of the cart, and its center of mass, which is displayed as the location of the cart. This simple principle has been exploited countless times in recent technology gadgetry and yet is so simple that an introductory physics student can readily understand and solve the equations. Application of equilibrium with force gauges has been used to Segway to control speed and forward/backward motion, Nintendo Wii Balance Board to detect two-dimensional center of mass of a player standing on top of it as a means of game control, highway weighing station that weights a vehicle without having to stop the vehicle, and countless other applications.

To use this apparatus as a smart track, simply put a cart on the track and perform an experiment with steps detailed by construction steps 13 thru 19. If one wants to pull the cart with a weight and a pulley, one needs to clamp the pulley on the edge of a lab bench instead of the end of the track to avoid the support force of pulley being detected and calculated as a part of the cart's force. One may also add springs on both ends of the track to make the cart oscillate or use a driver on one end of the track. Again, the driver may need to be clamped on a separate object to minimize its effect on the force gauges.

Appendix I: how the sonic ranger emulation works for Vernier system.

1. The data acquisition system pulls the INIT digital line to HIGH to initiate the sonic ranger to start measurement.

2. The ranger immediately sends out an ultrasonic.

3. The ranger receives a reflected pulse from the nearest object and pulls the ECHO digital line to HIGH to indicate it has detected an object.

4. The data acquisition system measures the delay between the start and the end of the measurement.

5. It then uses the speed of sound to calculate round trip distance and only outputs the one-way distance to the user. Steps 3-5 are emulated by the OSPL DAQ.

Appendix II: Program to be loaded via Arduino IDE version 1.0 or higher

```
#include <LiquidCrystal.h>    ///< Include the liquid crystal library
#include <phi_interfaces.h> ///< Include the phi_interfaces input devices library
#include <phi_prompt.h>     ///< Include the phi_prompt user interface library
#include "stat.h"

#define lcd_rows 2      ///< Specify the height of the LCD.
#define lcd_columns 16  ///< Specify the width of the LCD.

#define total_buttons 3  ///< The total number of push buttons in a buttons group object. This is needed to instantiate a
phi_button_groups object
#define btn_u 13        ///< I/O pin for a button
#define btn_d 12        ///< I/O pin for a button
#define btn_b 11        ///< I/O pin for a button

// LCD pin setting
#define LCD_RS 2 ///< Arduino pin connected to LCD RS pin
#define LCD_EN 3 ///< Arduino pin connected to LCD EN pin
#define LCD_D4 4 ///< Arduino pin connected to LCD D4 pin
#define LCD_D5 5 ///< Arduino pin connected to LCD D5 pin
#define LCD_D6 6 ///< Arduino pin connected to LCD D6 pin
#define LCD_D7 7 ///< Arduino pin connected to LCD D7 pin

LiquidCrystal lcd(LCD_RS,LCD_EN,LCD_D4,LCD_D5,LCD_D6,LCD_D7); // Create the lcd object

// The following lines instantiates a button group to control 6 buttons
byte pins[]={btn_u,btn_d,btn_b}; // The digital pins connected to the 6 buttons.
char mapping[]={'U','D','B'}; // This is a list of names for each button.
phi_button_groups my_btns(mapping, pins, total_buttons);

// This serial keypad is for debugging.
phi_serial_keypads debug_keypad(&Serial,115200);

// The following sets up function keys for phi_prompt library
char up_keys[]={"U"};
char down_keys[]={"D"};
char left_keys[]={"L"};
char right_keys[]={"R"};
char enter_keys[]={"B"};
char escape_keys[]={"A"};
char * function_keys[]={up_keys,down_keys,left_keys,right_keys,enter_keys,escape_keys};

multiple_button_input * keypads[]={&my_btns,&debug_keypad,0};

#define local_mode 0
#define emu_mode 1

const int InitPin = 9;
const int EchoPin =  10;
const int autoID_pull_down = 15; // The autoID resistor 22K has been soldered to this pin for easy pulling down to gnd.
const int left_gauge = 0;
const int right_gauge = 2;
const int right_gauge2 = 4;

const float weight_threshold = 0.5; // The threshold to consider a cart is on the track.
const float force_per_volt = 21.0; // This is 21N/V
const float level_arm = 0.2; // This is the distance from the center to one force gauge
const float t_offset = 5797; // This offset in microseconds will shift the sonic ranger reading by one meter so there will be
enough time to make the measurement and emulate an output. All sonic ranger output are thus offset by 1 meter.

int InitState = 0;
int prevState= 0;
int mode=local_mode;
```

```
int display_lcd=HIGH;
int track_lengths[]={100,110,200,220};
float x_offset=1.1; // This offsets the origin of x from center of the track to correspond to the reading on the tra
origin at one end.
float f_l=0, f_r=0, f_r2=0; // Left and two right forces, with weight of the track already taken off, so they add up
weight of the cart.
float f_tare_l=52.5;
float f_tare_r1=20.0;
float f_tare_r2=20.0;
float x_com=0, x_out=0; // Center of mass of the cart taking middle of the track as origin, and the output x with
origin.

long readings[3];
unsigned long pulse_us=0;

stat x_out_avg(x_offset);

void setup()
{
 Serial.begin(115200);
 lcd.begin(lcd_columns, lcd_rows);
 init_phi_prompt(&lcd,keypads,function_keys, lcd_columns, lcd_rows, '~'); // Supply the liquid crystal object, in
keypads, and function key names. Also supply the column and row of the lcd, and indicator as '>'. You can als
'\x7e', which is a right arrow.
 pinMode(EchoPin, OUTPUT);
 pinMode(InitPin, INPUT);
// pinMode(autoID_pull_down,OUTPUT);
// digitalWrite(autoID_pull_down,LOW); // Pull down this pin to supply gnd to the autoID resistor.
}

void loop()
{
 char buffer[128];
 char in_key;
 char track_choice;
 tare();
 lcd.clear();
 track_choice=simple_select_list("Track length:\n100cm\n120cm\n200cm\n220cm\n");
 x_offset=track_lengths[track_choice]*0.005;
 lcd.clear();

 while(1) // Measurement loop
 {
  switch (mode)
  {
   case emu_mode:
   InitState = digitalRead(InitPin);

   if ((InitState == HIGH)&&(prevState == LOW))
   {
    pulse_us=micros();
    measure_adc();
    calc_x();
    ranger_emu_output(pulse_us);
//     sprintf(buffer,"Reading left:%d right:%d\nForce(mN) left:%d
right:%d\nPosition:%d(cm)\n",readings[0],readings[1],int(f_l),int(f_r),int(x_out*100));
//     Serial.print(buffer);
   }
   prevState=InitState;
   in_key=my_btns.getKey();
   if (in_key=='D')
   {
    mode=local_mode;
    lcd.clear();
   }
```

```
    break;

    case local_mode:
    measure_adc();
    calc_x();
    in_key=wait_on_escape(50);

    if (display_lcd) print_lcd();

    if (in_key==phi_prompt_enter) return;  // Restart program

    if (in_key==phi_prompt_down)
    {
      mode=emu_mode;
      lcd.clear();
      lcd.print("Emulating");
      lcd.setCursor(0,1);
      lcd.print("Sonic ranger...");
      display_lcd=HIGH;
    }

    if (in_key==phi_prompt_up)
    {
      display_lcd=!display_lcd;
    }
    break;
   }
  }
}

void tare()
{
 lcd.clear();
 lcd.print("Balance track");
 while(1)
 {
  measure_adc();
  lcd.setCursor(0,1);
  lcd.print(readings[0]);
  lcd.print(" ");
  lcd.print(readings[1]);
  lcd.print(" ");
  lcd.print(readings[2]);
  lcd.print(" ");
  if (wait_on_escape(100)==phi_prompt_enter)
  {
   f_tare_l=readings[0]*force_per_volt/4/1024*5;
   f_tare_r1=readings[1]*force_per_volt/4/1024*5;
   f_tare_r2=readings[2]*force_per_volt/4/1024*5;
   Serial.println(f_tare_l);
   Serial.println(f_tare_r1);
   Serial.println(f_tare_r2);
   return;
  }
  delay(100);
 }
}

void measure_adc()
{
 readings[0]=0;
 readings[1]=0;
 readings[2]=0;
 for (byte i=0;i<16;i++)
 {
```

```
    readings[0]+=analogRead(left_gauge);
    readings[1]+=analogRead(right_gauge);
    readings[2]+=analogRead(right_gauge2);
  }
  readings[0]/=4;
  readings[1]/=4;
  readings[2]/=4;
}

void calc_x()
{
  f_l=readings[0]*force_per_volt*5/4096-f_tare_l;
  f_r=readings[1]*force_per_volt*5/4096-f_tare_r1;
  f_r2=readings[2]*force_per_volt*5/4096-f_tare_r2;
  if (abs(f_l)<weight_threshold/2&&abs(f_r+f_r2)<weight_threshold/2) x_com=0; // No cart is on the track, just output
x_com=0
  else x_com=(f_r+f_r2-f_l)*level_arm/(f_r+f_r2+f_l);
  x_out=x_com+x_offset;
  x_out_avg.add(x_out);
}

void ranger_emu_output(unsigned long init_t_us)
{
  float t_us=x_out*2/343*1e6+t_offset; // Calculate time delay needed for the echo to turn HIGH. 2 indicates round trip.
  while(micros()-init_t_us<t_us){}
  digitalWrite(EchoPin, HIGH);
  delayMicroseconds(200);
  digitalWrite(EchoPin, LOW);
}

void print_lcd()
{
  char buffer[128];
  lcd.setCursor(0,0);
  sprintf(buffer,"%4d.%01d(cm)",int(x_out_avg.get_avg()*100),int(abs(x_out_avg.get_avg())*1000+0.5)%10);
  lcd.print(buffer);
  lcd.setCursor(0,1);
  lcd.print(f_l);
  lcd.print(" ");
  lcd.print(f_r);
  lcd.print(" ");
  lcd.print(f_r2);
  lcd.print(" ");
}
```

stat.h

```
#include <Arduino.h>
#ifndef stat_h
#define stat_h
class stat
{
  public:
  float buffer[5]; // This stores floating point numbers for running average.
  int buffer_pointer; // This is the pointer for analog level for running average.
  void add(float number);
  float get_avg(); // Calculate for running average.
  float get_stdev(); // Calculate for running average and then standard deviation.
  boolean within_eb(float center, float dev); // Calculate whether the running average is within center +- dev.
  stat(float number=0.0);
};
#endif
```

stat.cpp

```
#include "stat.h"
#include "math.h"
```

```
stat::stat(float number)
{
 for (short i=0;i<5;i++)
 {
  buffer[i]=number;
 }
 buffer_pointer=0;
}

void stat::add(float number)
{
 if (buffer_pointer==5) buffer_pointer=0;
 buffer[buffer_pointer++]=number;
}

float stat::get_avg()
{
 float avg=0;
 for (short i=0;i<5;i++)
 {
  avg+=buffer[i];
 }
 avg/=5.0;
 return avg;
}

float stat::get_stdev()
{
 float avg=0;
 float stdev=0;
 for (short i=0;i<5;i++)
 {
  avg+=buffer[i];
 }
 avg/=5.0;

 for (short i=0;i<5;i++)
 {
  stdev+=(buffer[i]-avg)*(buffer[i]-avg);
 }

 stdev/=5.0;
 stdev=sqrt(stdev);
 return stdev;
}

boolean stat::within_eb(float center, float dev)
{
 float avg=0;
 float stdev=0;
 for (short i=0;i<5;i++)
 {
  avg+=buffer[i];
 }
 avg/=5.0;
 if (abs(avg-center)<dev) return true;
 else return false;
}
```

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

RC Circuits and Bioelectrical Impedance Analysis

Elliot Mylott, Sabrina Hoffman and Ralf Widenhorn

SRTC, 1719 SW 10th Ave., Room 134
Portland, OR 97201

503-997-7148

emylott@pdx.edu

Abstract

A single frequency bioelectrical impedance analysis (SF-BIA) device is presented, which is capable of measuring the impedance of known RC circuits as well as the impedance of a person's body. BIA is a common method of studying body composition that models the human body as an RC circuit. This device enables students to explore concepts in RC circuits such as phasor diagrams and impedance, as well as electrical characteristics of cells.

Construction of Apparatus:

The design of our apparatus (Fig. 1) was based on the device originally proposed by Yang, *et al.* (2006) and is similar to those found in commercial devices. The primary components are a XR-2206 function generator (FGEN) kept at a constant 50 kHz sine wave and the AD8302 Gain/Phase Detector. Differential amplifiers, input buffers and a voltage controlled current source (VCCS) were all constructed using LM324 op amps. The VCCS outputs a constant 800 μA, which is safe for human use and similar to the value used in commercial BIA devices. Four electrodes are needed to measure the impedance of deep body tissue and to avoid the effect of skin impedance. The hand held electrodes are constructed out of aluminum foil and plastic tubing. They are connected to the current injecting (*Ia* and Ib) and voltage sensing (*Va* and Vb) inputs using banana cables.

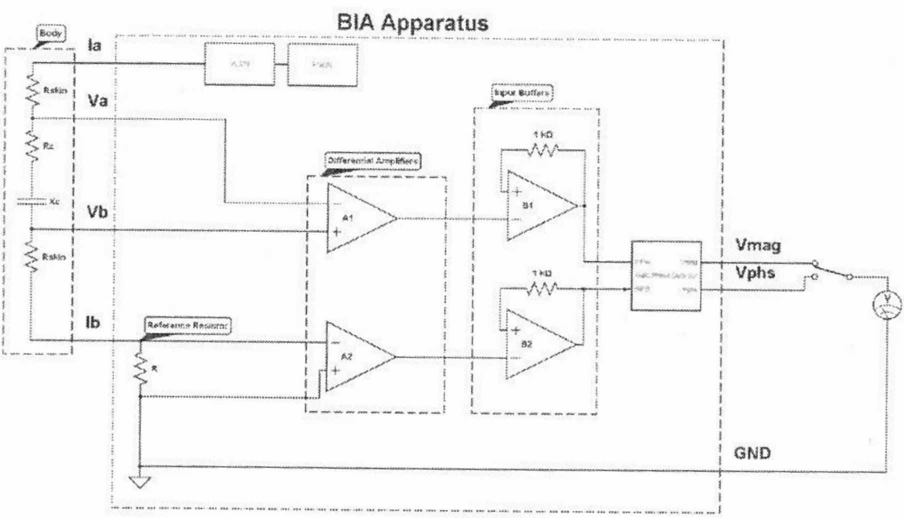


Figure 1 – Block diagram of single frequency bioelectrical impedance analysis (SF-BIA) device.

Component	Cost	Source
AD8302 Gain/Phase Detector	11.80	http://en.module-china.com/index.php?main_page=product_info&products_id=111552
DIP Adapter for AD8302	7.00	http://www.logicalsys.com/painfo-vpasp-vb.asp?adapter=pa-ssd3sm18-14
XR-2206 Function Generator	6.49	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_34972_-1
3x LM324 Op Amps	1.44	http://www.digikey.com/product-detail/en/LM324N/497-1581-5-ND/591683
IC Board	0.99	http://www.mcmelectronics.com/product/21-4590&scode=GS401&CAWELAID=220241326
11x Banana Jacks	6.86	http://www.digikey.com/product-detail/en/108-0902-001/J151-ND/5929
6x Banana Patch Cords	9.42	http://www.artistsupplysource.com/product.php?productid=61818
50 Piece Assorted Resistors	6.29	http://www.radioshack.com/product/index.jsp?productId=2062305&filterName=Type&filterValue=Resistor+assortments
7x 1 µF Capacitors	1.40	http://www.digikey.com/product-detail/en/ECA-1HM010/P5174-ND/245033
8x 10 nF Capacitors	2.32	http://www.digikey.com/product-detail/en/FK18X7R1H103K/445-5297-ND/2256777
9V Battery	1.40	http://www.amazon.com/UltraLast-Alkaline-General-Purpose-Battery/dp/B002KT7URA
Multimeter	7.86	http://www.parts-express.com/pe/showdetl.cfm?Partnumber=390-500
4x 16 pin Dual In Line Sockets	0.52	http://www.mouser.com/ProductDetail/3M-Electronic-Solutions-Division/4816-3000-CP/?qs=lGgAdOvCTsQZ8wkl0%2fMDj1VOKBEgxFRy6PyN6jpO87c%3d
2 Way Switch	0.29	http://www.digikey.com/product-detail/en/EVQ-11U09K/P8085SCT-ND/259570
Wire	negligible	
Solder	negligible	
Project Box or Tin	negligible	
Aluminum Foil	negligible	
2x Plastic tubes	negligible	
Electrical Tape	negligible	

Total Cost 64.08

Use of Apparatus:

One of the simplest electrical models of the body is as a RC series circuit in which the resistor represents both intra and extracellular water and the capacitor represents the ability of cell membranes to retain charge. Compared to muscle tissue, fat cells have a very high impedance and so conduct electricity very poorly. Therefore the affect a body has in changing the currents through it will be predominately due to fat free mass (FFM). The FFM is an indicator of a person's body composition and used in devices such as bath room scales, but can also be found in a clinical setting to monitor a patient's health.While the better known Body Mass Index (BMI) is a decent indicator for the general population, it does not take into account muscle mass. BMI compares height to weight. When one person is heavier than another of the same height, they may have gained that weight from fat weight or muscle weight. Of course, this results in very different conclusion about a person's health. The FFM takes into account the amount of muscle mass a person is carrying and relates that to their height. The FFM index is not necessarily a superior index, but it does factor in different parameters than simply total weight.

This laboratory exercise can be designed such that a person's FFM is revealed or remain private if students choose to do so. We recommend that data is taken from student volunteers. Our experience has been that you always find students that are quite curious about their FFM index. It should be pointed out that this experiment is not intended to measure a person's FFM precisely, but illustrate the physics behind this very intriguing device.

By comparing the voltage across a person and across a known resistor connected in series (Fig. 2), the value of R_Z and X_c can be calculated. And much in the same way that the mass of a resistor can be calculated using its dimensions, resistance and resistivity, a person's FFM can be estimated using equations of the form

$$FFM = a + b \cdot \frac{Height^2}{R_Z} + c \cdot X_C + d \cdot Weight + e \cdot Gender, \qquad (1)$$

where *a-e* are empirically determined coefficients and *Gender*=1 for men and 0 for women.

If an AC voltage is applied across the circuit in Fig. 2, the dashed section will have a complex impedance $Z\angle\varphi$ given by

$$Z = \sqrt{R_z^{\,2} + X_c^{\,2}} \qquad \tan\varphi = \frac{X_c}{R_z} \qquad (2)$$

Eq. (2) can be solved for R_Z and X_c as functions of impedance Z and phase angle φ.

$$R_z = \sqrt{\frac{Z^2}{1+\tan^2\varphi}} \qquad X_C = R_z \tan\varphi \qquad (3)$$

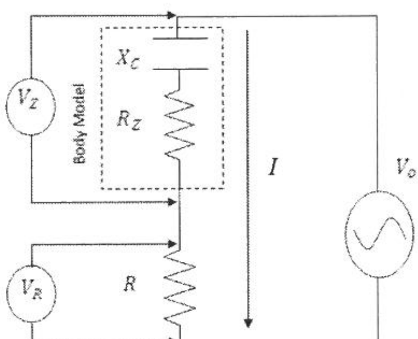


Figure 2 – Circuit to determine a body's resistance and reactance (dashed lines) Using a known resistor *R*.

Because all of the components are in series they have the same current passing through them, so the gain of the voltages across Z and R is given by

$$K = \frac{V_Z}{V_R} = \frac{ZI}{RI} = \frac{Z}{R} \tag{4}$$

The AD8302 Gain/Phase Detector outputs DC voltages (V_{mag} and V_{phs}) proportional to the gain K and phase φ between the two inputs given by

$$V_{mag} = V_{slp} \cdot \log_{10} K + V_{offset}$$

(5)

$$V_{phs} = V_{\varphi} \cdot \varphi + V_{offset}$$

The values of the slopes and offsets of each equation can be calculated using known components and plotting V_{mag} and V_{phs} versus the expected *log K* and φ respectively. Using these measurements, the value of the known resistor R and Eq. (3, 4) the components R_Z and X_c can be estimated.

Accurate measurements are heavily dependent on concentrations of water in the body, so a strict procedure lasting 24 hours controlling fluid intake, restroom use and even the position of the patient while testing need to be followed. Because students are not asked to follow these guidelines prior to using the device, there will likely be an error in an individual's final calculated values.

Procedure

Measuring known components

- Use four banana cables to connect *Ia* and *Va* to one end of an RC circuit and *Ib* and *Vb* to the other end (Fig. 1).
- Measure *Vmag* by connecting the High and Low of the multimeter to *Vmag* and *GND* respectively.
- Measure *Vphs* by connecting the High and Low of the multimeter to *Vphs* and *GND* respectively.
- Use Eq. (3 - 5) to calculate R_Z and X_c.

Measuring FFM

- Use banana cables to connect *Va* and *Vb* the voltage sensing electrodes and *Ia* and *Ib* to the current injecting electrodes.
- Measure Vmag by connecting the High and Low of the multimeter to *Vmag* and *GND* respectively.
- Measure Vphs by connecting the High and Low of the multimeter to *Vphs* and *GND* respectively.
- Use Eq. (3 - 5) to calculate R_Z and X_c.
- Use the calculated R_Z and X_c and Eq. (1) to calculate the FFM.

Using this setup, students can quantitatively explore many topics commonly taught in undergraduate electronics classes including: AC voltage, impedance (resistive and reactive), phasor diagrams and phase shifts. And by highlighting the electrical characteristics of biological tissue it also reinforces the relevance of general physics to pre-health education.

References

Yang, Yuxiang *et al* 2006 Design and preliminary evaluation of a portable device for the measurei
bioimpedance spectroscopy *Physiol. Meas.* 27 1293

Apparatus Competition

2012 AAPT Summer Meeting

Philadelphia, Pennsylvania

Faraday Rotation Apparatus for Lecture Demonstration

Dale Stille and Prof. Michael Flatté

Rm 58 Van Allen Hall, Dept. of Physics &Astronomy, Univ. of Iowa, Iowa City, IA 52242

319-335-1833

dale-stille@uiowa.edu

Abstract

Faraday rotation experiments, plane-polarized light through a sample being rotated by some angle when a magnetic field is applied, have long been a staple of most advanced physics laboratory curriculums but have been plagued by high costs, size, fragility, or operating difficulties of the components used. Technological advances in the areas of diode laser pointers, high strength permanent magnets, and readily available metal doped glass samples, combined with dramatic price decreases for these components now make this experiment easy and suitable for not only any advanced laboratory but also as a lecture demonstration.

Construction of Apparatus:

Our Faraday rotation apparatus is shown in Fig. 1 and is similar to standard laboratory apparatus for measuring optical rotation but uses permanent magnets that can be slid over the sample material instead of a fixed electromagnet or permanent magnets. The unit was made from an old homemade polarimeter apparatus that was modified to use large diameter 360 degree protractor scales for easy classroom viewing. A large dial pointer was added so that angular rotation of the analyzer could also be quickly pinpointed. The sample of terbium doped glass rod[2] is approximately 4 mm dia. X 10 mm long and is held in a 10 cm. length of impact resistant polycarbonate tube[3] mounted between the polarizer and the analyzer with the sample located approximately 4 cm. from one end of the tube. The neodymium ring magnets[4] then slide onto the polycarbonate tube so that they may be positioned away from or over the sample when doing the demonstration.

Use of Apparatus:

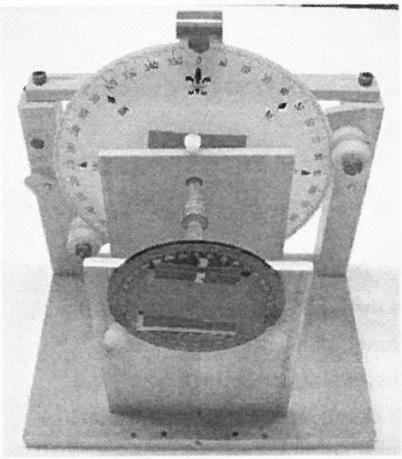


Fig. 1. The basic apparatus with the magnets positioned over the sample.

To date we have demonstrated 3 separate effects in class with this apparatus and are working on a 4th. The three effects we will report here are the dependence of wavelength on rotation angle, the effect of temperature on rotation angle, and time reversal.

The Dependence of Wavelength on Rotation Angle

We have six laser pointers[5] of different wavelengths mounted in an array that we use for this demonstration which is shown in Fig. 2. The array allows us to start with all the lasers at the correct height. We then just slide the apparatus from one laser to another during class which allows us to show the full range of the effect with a minimum of wasted time. The wavelengths and colors of our pointers are 650 nm red, 632.8 nm red, 593.5 nm yellow, 532 nm green, 473 nm blue, and a 405 nm purple. Our procedure is to shine the desired laser through the sample without the magnetic field being applied, set the analyzer to zero degrees, and rotate the polarizer until minimum light transmission or extinction is achieved on the screen. The magnetic field is then applied by moving the ring magnets over the sample at which point a bright spot of light will appear on the screen. Rotating the analyzer until that bright spot is again at a minimum or extinguished will give you the rotation angle for that wavelength. As seen in Table 1, our samples rotation angle is less than ten degrees at 650 nm but increases to 30 degrees at the 405 nm wavelength.

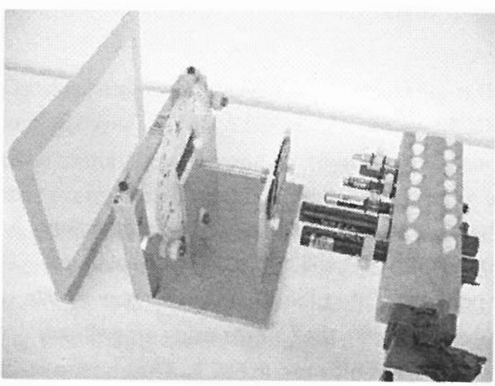


Fig. 2.

There is one important consequence of using laser pointers that needs to be noted here, that is almost all laser pointers are polarized to some degree. When used for this demonstration they need to be aligned so that you can rotate the analyzer at least 80 degrees or more in the needed rotation direction without the laser beam going through a minimum intensity or being extinguished, which would interfere with the collection of data.

Table 1. Rotation Angle Data

Wavelength (nm)	**Rotation Angle** in degrees	**Rotation Angle (Liq. N_2)** in degrees	**Time Reversal Rotation Angle** in degrees
650	7 to 8	13 to 14	8 to 9
632.8	9 to 10	21 to 22	11 to 12
593.5	10 to 11	30 to 31	12 to 13
532	13 to 14	41 to 42	15 to 16
473	17 to 18	52 to 53	19 to 20
405	29 to 30	76 to 77	31 to 32

The Effect of Temperature on Rotation Angle

The terbium doped sample we have is rugged enough to be cooled to liquid nitrogen temperatures but needs to be held in a plastic tube that will also survive the plunge. This is the reason for the impact resistant polycarbonate tubing mentioned above. We drip liquid nitrogen over the area of the tube containing the sample using an ordinary 60 cc plastic syringe with a large bore blunt needle as pictured in Fig. 3. The cooling of the sample in this fashion only takes 20 to 30 seconds of class time with the actual measurement taking another 5 to 10 seconds. We do not cool the magnets as it would take about 2 minutes of class time to bring them down to temperature although this would be the better alternative if we were taking more exacting data. The rotation angle of each wavelength of light when the sample was cooled to liquid nitrogen temperatures is also shown in Table 1. Note that the rotation angle at liquid nitrogen temperature is 2 to 2 ½ times the rotation angle at room temperature.

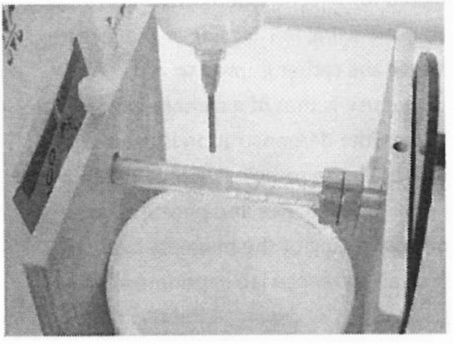


Fig. 3.

Time Reversal

Consider a single wave front that moves from the laser, through the polarizer, sample and analyzer, and onto the screen. At each point in that path you could plot the polarization direction and rotation angle if desired. What you have really done with that exercise is to look at what happens to the wave front as it travels through time on its path from laser to screen. In many ways the desirable way to describe or think of this experiment is as a polarized wave front that travels through time. Time reversal then would be nothing more than switching the position of the lasers and the screen in Fig. 2 so that the laser light enters the sample from the opposite end. Theoretically you should get the same rotation angles no matter which end of the sample the laser light enters. In reality this is rarely the case due to how the metal atoms self organize themselves within the sample. As shown in Table 1, there is several degrees difference in rotation angle of our sample when the laser light was directed through it from the opposite end.

Conclusion

The apparatus as shown and described is optimized for a lecture demonstration setting where time management is of the essence. It is done using full lighting in the lecture room with the instructor's eyes as detectors, hence the rather imprecise measurements reported in Table 1. The only aid we use is that of a camera looking onto the back side of our tracing paper screen when the demonstration is used in our large classrooms so that the laser light coming through the sample is visible to all. Obvious improvements would be the addition of an oscilloscope and photocell to replace the eye as a detector, and more precise positioning of the magnets over the sample, but then this reverts back to the realm of an advanced lab experiment and not a simple lecture demonstration.

References

2. Kigre, Inc., www.kigre.com, # M-18 Faraday Rotator Glass, Terbium doped.

3. McMaster-Carr, www.mcmaster.com, # 8585K51, $1.30 / ft.

4. Forcefield, www.wondermagnet.com, # 0021, $8.50 ea.

5. Lasers of different wavelengths can be had at Laserglow Technologies, www.laserglow.com. Cheap laser pointers may also be found by doing a search on the web, but be aware that many of these are not rated for continuous use which is usually considered to be more than 5 seconds at a time.

Michael Flatté is a Professor in the Department of Physics and Astronomy at the University of Iowa and is also the Director of the Optical Science and Technology Center. He received his Ph.D. from California, Santa Barbara (1992).

Dale Stille is the Instructional Resource Specialist and Outreach Coordinator for the Department of Physics and Astronomy at the University of Iowa. He received a BS in Chemistry from Buena Vista University (1975) and an MS in Chemistry from the University of Iowa (1979).