

Workshop on Graphical Programming for the Physics Laboratory*

Muhammad Sabieh Anwar
Centre for Experimental Physics Education
LUMS School of Science and Engineering

April 21, 2016
Version 2016-1

Labview Basics.

1. Input a number. Test whether it is even or odd. An LED will blink if the number is even.
2. Generate a sine curve and display it graphically. This can be achieved through building the following block diagram. We will go through the diagram together.

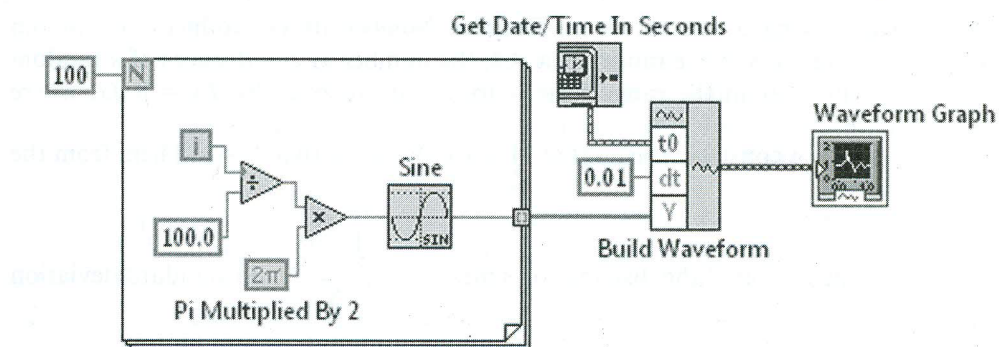


Figure 1: Block diagram for generating a sine wave.

3. Make a graphical program which allows the user to input a whole number N and returns its factorial $N!$.
4. Construct a program that tests whether an input number is a prime number or not. Use a case structure within a `while` loop. Check division with respect to 2 and follow the flowchart provided in Fig. 2.

*No part of this document can be re-used without explicit permission from the author(s). I like to thank Umar Hassan for help in illustrations. I also like to thank Hafiz Muhammad Rizwan for meticulously constructing the hardware assemblies.

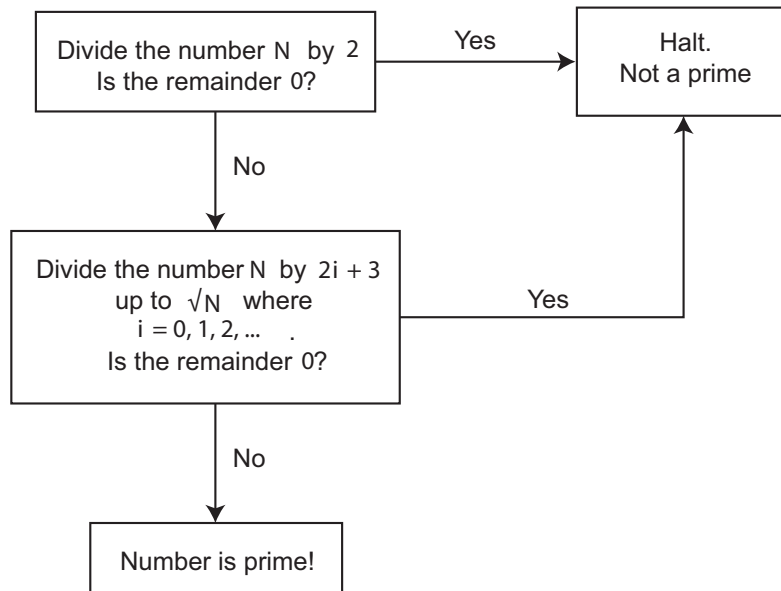


Figure 2: Possible prime number detection algorithm.

- I like you to make a graphical program that allows you to plot the range of a projectile R as a function of the launch angle θ . The user is able to input a fixed value of the initial speed v and the height from which the object launches h . You will plot an XY graph. The front panel will look like Fig. 3. The relationship between the range and other parameters is given by,

$$R = \frac{v^2}{2g} \left(\sin(2\theta) + \sqrt{\sin^2(2\theta) + \frac{8gH}{v^2} \cos^2(\theta)} \right). \quad (1)$$

These variables are defined in Fig. 4.

Acquiring signals.

At this stage, we will orient you towards data acquisition: first fetching signals into the computer, their display and analysis.

- We will generate a signal from a function generator of a reasonable frequency and use an Express VI called DAQ Assistant to input the signal and display it on a Chart.
- In the previous exercise you will see that the signal appearing on the Chart is not steady, rather it flickers continuously. We can correct this by hardware triggering. Learn how to do this.

You should make yourself familiar with the different **kinds of ways of inputting analog signals into a data acquisition device**. We will frequently use the *referenced single ended* and *differential* modes. These modes are shown in Fig. 5.

- We will now attempt to make a more sophisticated digital oscilloscope using DAQmx VI's. These low level VI's offer greater versatility and control over the data acquisition process. Use the DAQmx Timing and DAQmx Triggering options to control the process.

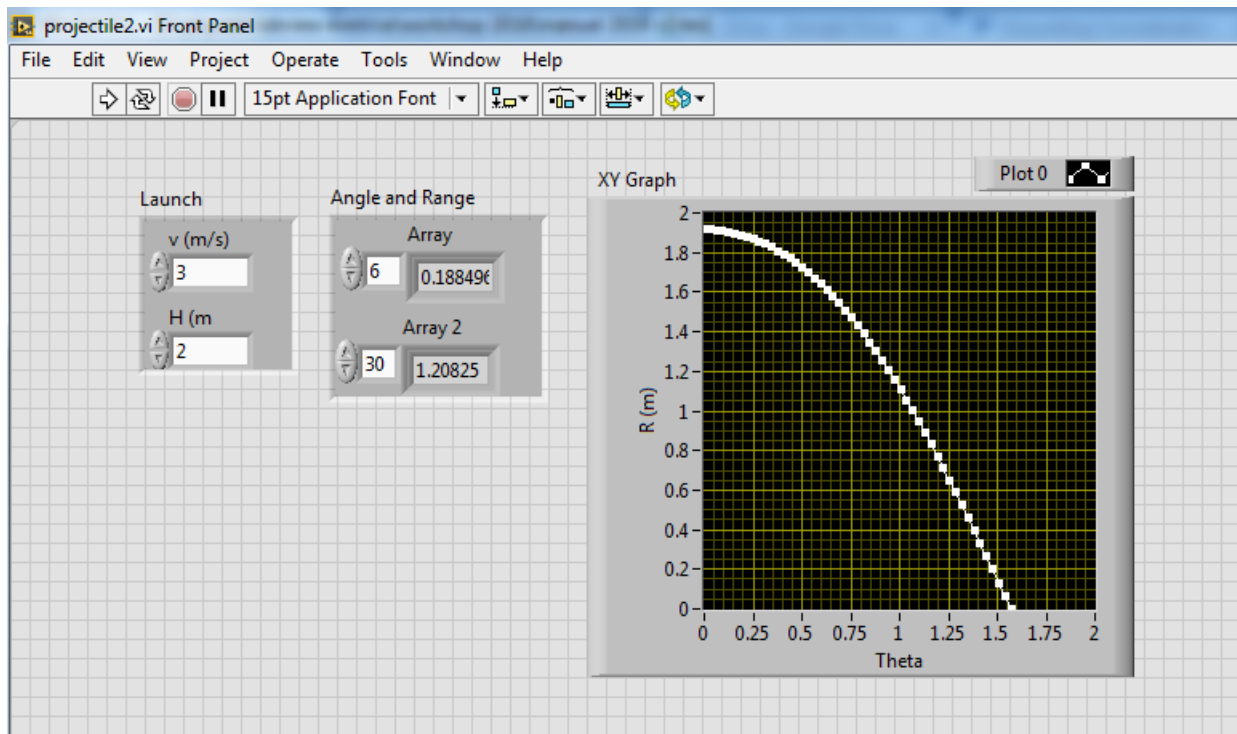


Figure 3: Desired front panel for projectile calculation.

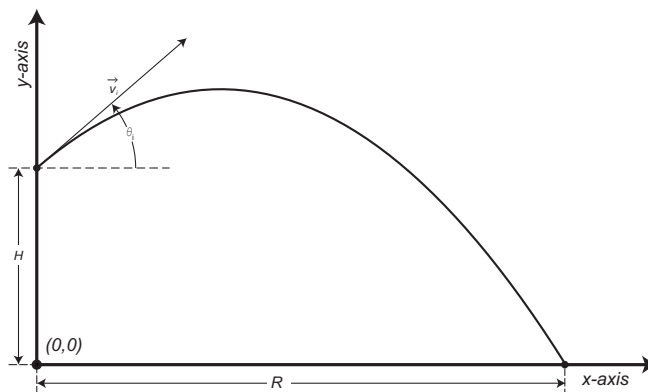


Figure 4: Defining the variables for projectile motion.

We will build this high end oscilloscope together. We prefer a sampling of 10 kHz and our input signals from the function generator will be in the hundreds of Hz range. Your actual block diagram may look similar to Fig. 6.

Generating signals.

We will now proceed on the **generation of signals**, using a DAQ card as an output device. This is an essential part of the regimen of experimentation in the physical sciences.

9. Generate a sine wave output with a variable frequency and a sampling rate of 1 kHz. Read this wave on an oscilloscope and confirm your observations. We will first generate

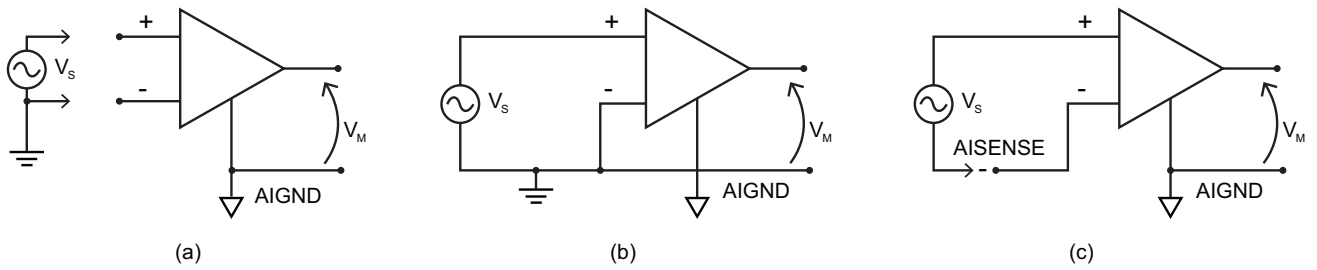


Figure 5: Distinct ways of connecting an input signal: (a) differential, (b) referenced single ended (RSE) and non-referenced single ended (NRSE). The source voltage is V_s and the measured voltage is V_m .

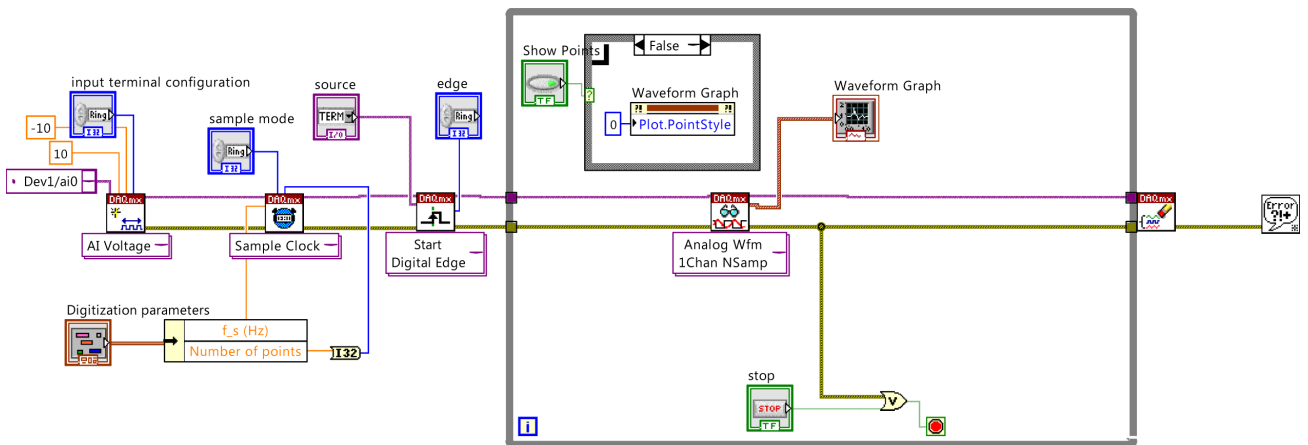


Figure 6: Candidate block diagram for a high-end digital oscilloscope.

a *software timed* signal followed by *hardware timing*. I will explain the difference between these two types. We could use an express VI, the DAQ Assistant here.

- Now let's put all of this together. We will generate a square wave of variable frequency and amplitude keeping its baseline at 0 V. We will input this excitation into a series combination of a variable resistor R and capacitor C . We will measure both of these voltages in real time and see if we observe the desired characteristics of these components. This printed circuit has been constructed for its ease of use and should be available in the kits we've provided. The circuit diagram is shown in Fig. 7.

Acquiring signals from sensors and transducers.

Most physics experiments rely on **capturing data from sensors**. Each sensor works on a distinct physical principle. I have chosen two of the most commonly used sensors, the thermistor and the thermocouple. Both measure temperature but work on absolutely different principles.

- We provide you with a thermistor (TCS-610) which is a negative temperature coefficient semiconductor with a nominal resistance of 10 K at 25° C. It is placed in series with a

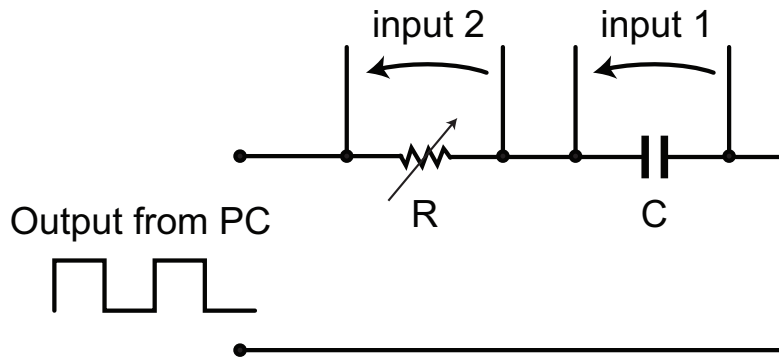


Figure 7: Scheme of the RC circuit.

known resistor. We will excite it with a 2.5 V DC signal generated from the data acquisition device. We will subsequently measure the voltage drop across the thermistor and from this reading, deduce the resistance of the thermistor R_T which in turn determines the measured temperature. A reasonably good approximation for the temperature (in Kelvins) is

$$T_K = \frac{1}{A + B \ln R_T + C(\ln R_T)^3}, \quad (2)$$

where A , B and C are coefficients that can be extracted from the manufacturer's data sheets. In the course pack I've included the data sheets. You will construct a program that excites the thermistor, measures R_T and interactively reads off the temperature and even displays it in a Chart. You may, for example, use the `Convert thermister reading.vi` available in `Mathematics`→`Numeric`→`Scaling` or write a formula in `Mathscript`.

For heating, I am providing you with a heating element that resistively heats a small cylindrical piece of aluminum perched on a teflon holder. The heater can be directly powered from a 12–24 V DC supply. The heating assembly is shown in Fig. 8.

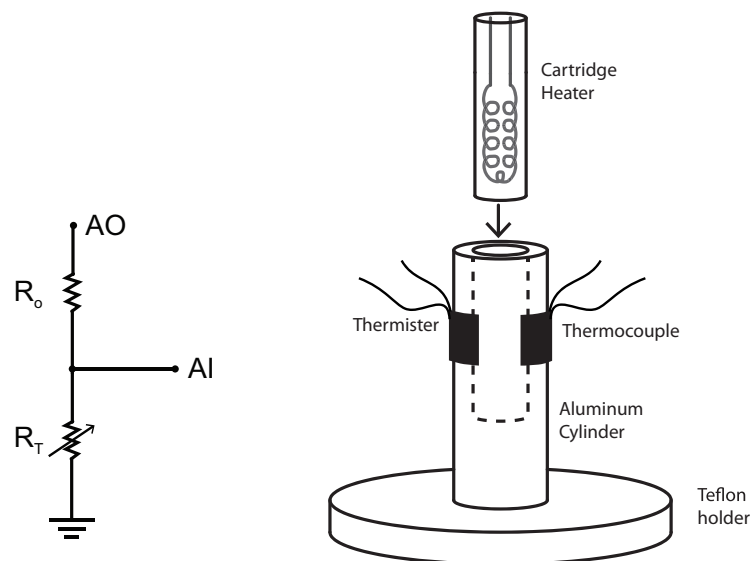


Figure 8: Scheme for measuring the resistance of the thermistor and heating assembly for temperature measurement and thermal control.

12. Measuring temperature with a thermocouple is easy but requires some thinking. I will explain the philosophy of using a thermistor in conjunction with a thermocouple through Fig. 9. This technique is called cold junction compensation. For details on implementing a cold junction and reading in temperatures using a thermocouple, I've included as an appendix another document that we wrote some years ago. In short, read in the temperature from a thermocouple attached to the heater assembly and display it in real time on a chart.

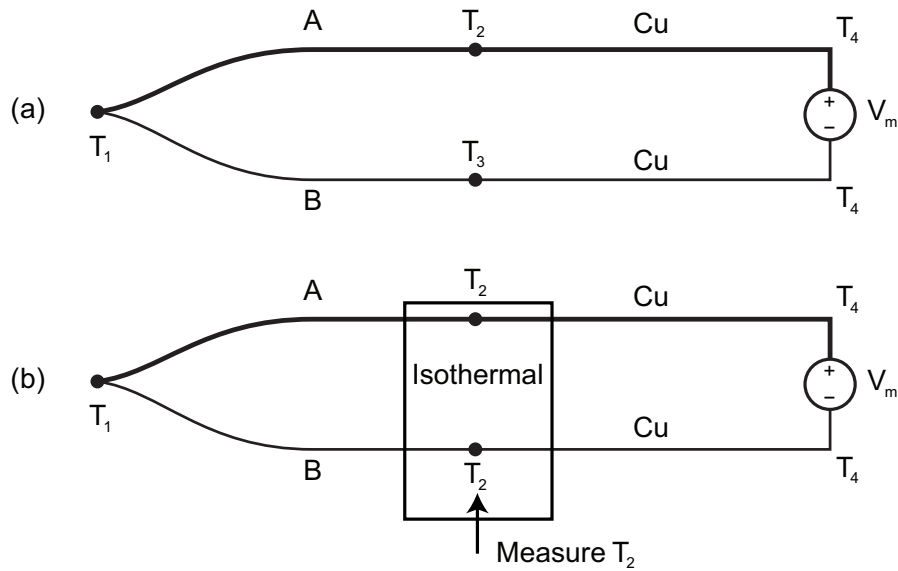


Figure 9: A thermocouple circuit without (a) and with (b) a cold junction.

PID control of a thermal assembly.

At this juncture, there will be dedicated lecture on basics of **control and the proportional-integral-derivative (PID) algorithm for control.**

13. Construct a graphical program that can help achieve a desired temperature for the thermal object. the heater is actuated by a pulse-width modulated (PWM) square wave whose duty cycle d is controllable. The DAQ card supplies a switching signal to a transistor that helps operate a relay and hence, the heater supply current is indirectly modulated. The data acquisition device itself is not capable of delivering high power to the ≈ 25 W heater. The circuit diagram that switches on a relay through a transistor and hence provides variable d to the heater is shown in Fig. 10. The conceptual diagram that illustrates the control mechanism is shown in Fig. 11. Our PI algorithm will exercise control over the variable d .

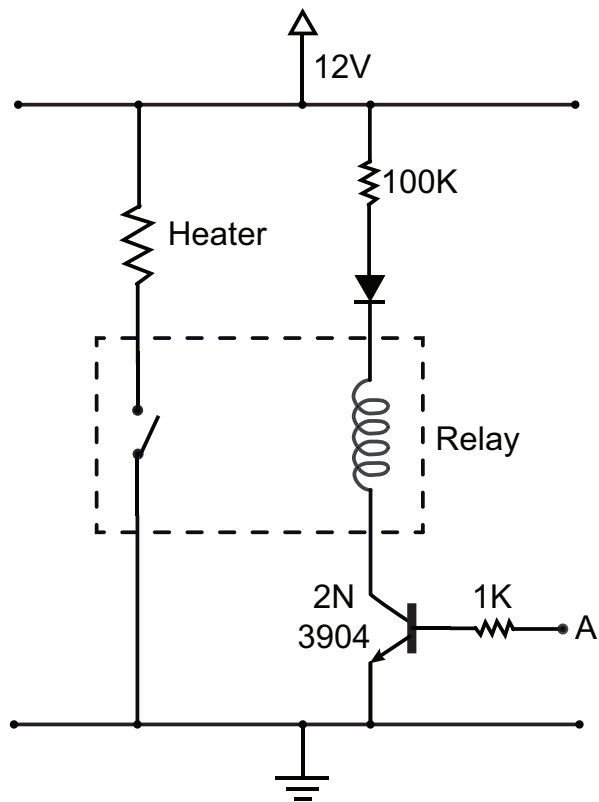


Figure 10: Circuit for actuating the heater under instructions from the computer.

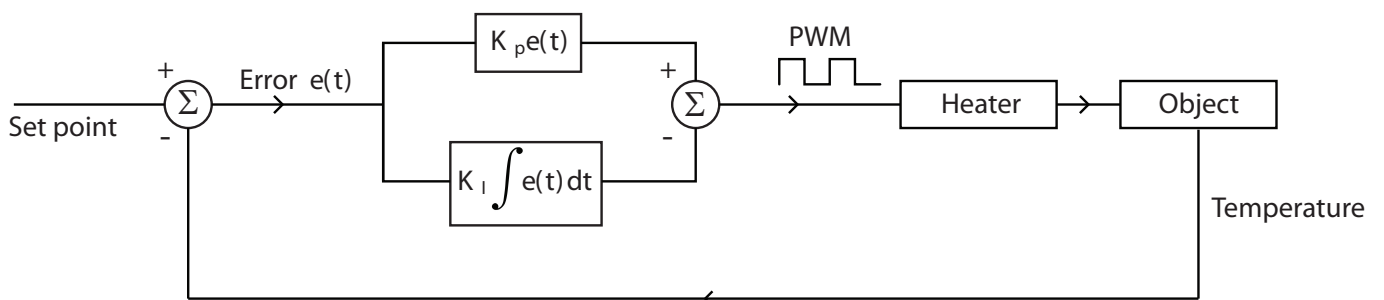


Figure 11: Feedback algorithm for implementing PI control on the temperature of the heating element.