Megan Mair

Practical Introduction to Programming for Scientists Final Project

Prothrombin time, the time it takes for blood to clot, along with its derived international normalized ratio (INR) formula, is an important measure of blood coagulation in patients on warfarin therapies. The development of at-home and point-of-care devices is important for rapid diagnoses, and, importantly, encourages patient compliance to clinical treatments. Few at-home/point-of-care INR testers exist, and those that do are often cost prohibitive for at-home use by patients. The following project acts as a basic proof-of-concept for a Python and Raspberry Pi-based INR tester.

The program utilizes Shelve to save user data. When initially loading up the program, a GUI interface from tkinter prompts the user to enter their name, which is used to manage the user data as well as personalize the user's experience in the interface. Once the user enters their name, they are given a welcome message on the main start screen. Upon subsequent start-ups of the program, the user will be greeted by this default starting page.

The home page has three options for the user: New INR Test, History, and a Quality Check option. We will first discuss the INR Test portion of the program. The INR is calculated based on the amount of time it takes blood to clot when exposed to thromboplastin. Traditionally, blood clotting is measured based on changes of how light passes through or reflects off a sample as blood changes from a liquid to gel-like state. This program uses photoresistor and laser Raspberry Pi modules to create a similar interface to previous INR testers. When a liquid sample changes to a more solid state, less light passes through to the photoresistor, reflecting a change in its measured value. The sample is considered "coagulated" when the photoresistor reaches a certain threshold in its light measure. Due to the obvious safety hazards associated with using real blood samples, the device currently calculates INR using sodium acetate, a common household chemical. In a concentrated form, sodium acetate ("hot ice") can rapidly change from a liquid to solid state induced using a few solid sodium acetate crystals to seed the reaction.

The Raspberry Pi components of the INR Tester are arranged as follows: The photoresistor component is covered by a straw to control the amount of light received, and placed directly behind a clear sample chamber. The laser component faces the chamber directly opposite the photoresistor. During the INR test, a timer begins and the laser immediately turns on and shines directly through the clear chamber to the photoresistor. The user can then add sodium acetate liquid and crystals to the sample chamber. In a few seconds, the sodium acetate begins to crystalize, blocking the amount of light received by the photoresistor. Once the photoresistor measure reaches an experimentally-defined threshold, the program ends the timer and turns off the photoresistor and laser. The final count on the timer minus the initial count is equivalent to the Prothrombin Time used for the INR equation, found below.

Also important for the INR calculation are the MNPT and ISI value. When selecting the INR test option, the user is immediately prompted to enter custom MNPT and ISI values. Allowing for custom values offers the program versatility, as these values can come from user-preferred thromoplastin aliquots or by test strips used by other INR testers. Before accepting the values, the program tests to see if a float number is input; if not, the user is prompted to check their values. If the values are of accurate type, the Check Values button is converted to a Continue button. When Continue is pressed, the user is asked to confirm the values, and is allowed to change the values as necessary. If the user approves the values, they are directed to the main INR test page, where the main Raspberry Pi modules can be run.

$$INR = \frac{[Prothrombin\ time\ for\ patient]}{[Mean\ Normal\ Prothrombin\ Time]}^{International\ Sensitivity\ Index}$$

Once the Prothrombin Time is calculated, the program feeds the input and measured values through the INR equation, and directs the user to a results page reading out their INR. The user is asked if they want to save their result; if the Save button is selected, the data is added to the Shelve dictionary using a time and date stamp as the key, and the INR result as the value. This format assists in results display later found in the History screen. If the user selects Cancel, they are directed back to the home screen and their data is not saved. If the data is saved, the Save button is eliminated and the Cancel button changes to prompt the user to return to the home screen.

The next option on the home screen is the History button. If the user has previous INR reads saved, they can be viewed on this page in ascending order (the most recent reading is first). The data is displayed with a date/time stamp, as well as the calculated INR. The user can choose to export this data to a CSV file called UserName_INRdata.csv. This option is useful if the user would like to share the information with their doctor. If the user has no previous history, the History page instead displays a message letting them know no history is available.

Finally, the last option on the home screen is the QC option. Primarily, the Quality Check can be used to physically adjust the laser/photoresistor placements to ensure accurate test results. When selecting this option, the laser and photoresistor turn on, and the measure of the photoresistor will print to the terminal (a measure of 0 is ideal); if the measure reaches above the threshold used to calculate coagulation, the message “threshold reached” also appears. On top of ensuring the laser is reaching the photoresistor, this mode is useful to run experiments and determine thresholds for other sample types (for example, blood as the project develops).

Directions

Dependencies: Raspberry Pi with Python 3; shelve, time, datetime, RPi.GPIO, ADC0832.py (script provided)

Circuit set-up: Pin for laser: 15 | Pins for ADC Converter: CS to 11, CLK to 12, I/O to 13 | Signal of photoresistor goes to CH0 of converter

1) Launch the project python file and open the GUI window. Enter your name, then click enter.
2) On the home screen, navigate to the History page. No results should be displayed.
3) Navigate to the QC page, run a test. In the terminal, a series of photoresistor results is output. Adjust the laser and photodiode until the readout is “0”. Use CTL+C to exit test.
4) Navigate to the New INR Test page. Enter MNPT and ISI values (recommended MNPT: 12, ISI: 1). Press the enter button; approve the values.
5) Begin the INR Test. The GUI screen will not work while the test is running. Add the sample to a clear sample chamber. In absence of a sample, you can end the test by placing a solid object in between the laser and photoresistor.
6) Once prompted, save the results and return to the home screen. Go back to the History page to view your results. Click save to export results to a CSV file.