9-2017

An SPM Stage Driven by 3 Stepper Motors

Jianghua Bai
Portland State University

Andres H. La Rosa
Portland State University, andres@pdx.edu

An SPM Stage Driven by 3 Stepper Motors

Jianghua Bai, Andres La Rosa

Department of Physics, Portland State University, USA

Abstract: A Scanning Probe Microscope (SPM) stage controlled by 3 stepper motors is designed in this project. The SPM stage controlled by 3 steppers is more versatile than a stage controlled by one motor, but the control of the system is more complicated. In this project, we build the stage actions in an Arduino microcontroller. A finite state machine (FSM) is also built in Arduino to communicate with a PC and an RF controller. A special displaying scheme which has 5 states, is also employed to indicate the operation of the stage. Finally, the SPM stage is fully tested and has a 700 nm mechanical resolution in the Z direction.

Keywords: SPM, Stepper motors, RF control, LabVIEW

1. Introduction.

Scanning Probe Microscopes (SPM) are important tools in modern nanoscale research. Commercial SPMs, like the one shown in fig 1, have fine, stable and expensive mechanical structures. Usually, during operation, a commercial SPM employs a servo motor or a stepper motor to move the scanning head up and down.

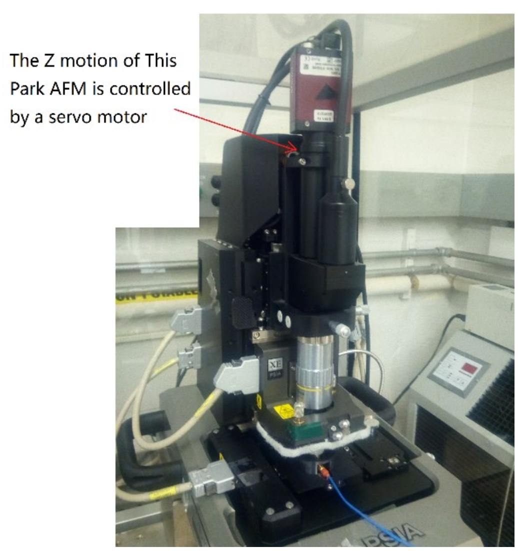

Fig 1, A Park PSIA XE-120 Atomic Force Microscope

One wants a house made SPM to be versatile, flexible, and able to perform non-standard measurements. A heavy stage supported by 3 high definition screws is a good solution for a house made SPM system. Before the experiment, the operator manually turns the screws and lowers the stage down toward a sample. When the probe is close enough to the sample, an electric piezo will take over the control of the probe. As shown in fig 2, turning 3 screws, one is able to move the stage upward and downward. Furthermore, by adjusting the 3 screws, one is able to change the tilt of the plane, which introduces more flexibility.

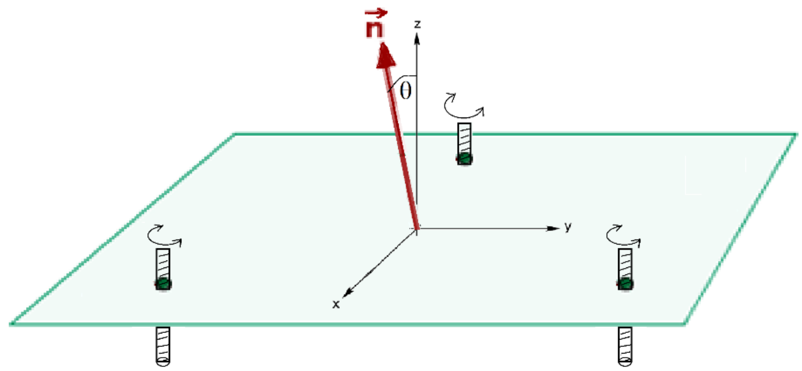

Fig 2, The stage has the ability to adjust the plane direction.

There is another benefit with this structure. The scanning tube is mounted at the center of the stage. See fig 3. According to the lever mechanism, the travel distance of the tip is half of that of the screw. Comparing with stages held by one screw, this stage has finer mechanical motion.

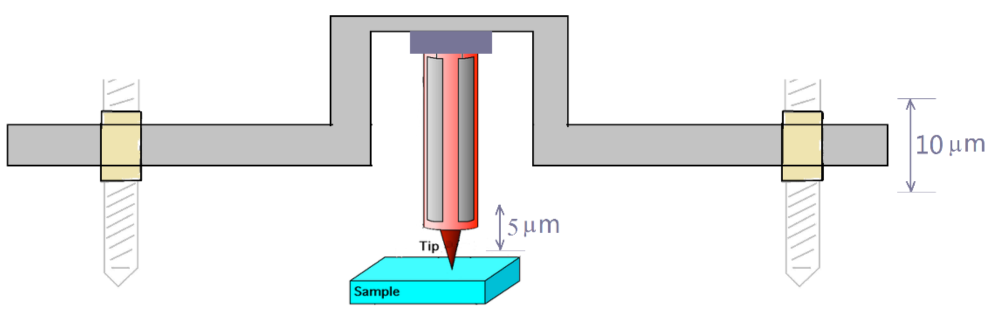

Fig 3, when the screw travels 10 μm, the tip only travels 5 μm.

2. The Mechanical Structure of the SPM Stage.

But the stage shown in fig 2, has several problems. Adjusting the screws manually does not guarantee stable operations all the time. In some applications, say, the stage needs to be put into a humid chamber or a vacuum chamber, it is impossible to control the stage manually. In order to have a stable operation and meet the requirements of new applications, we want to use motors to drive our stage.

Following a simple path, we attach 3 stepper motors to the 3 fine adjustment screws, as shown in fig 4 and fig 5. Grouping all the wires to a specific location, now the stage is ready to be put in a vacuum chamber or other controlled environments.

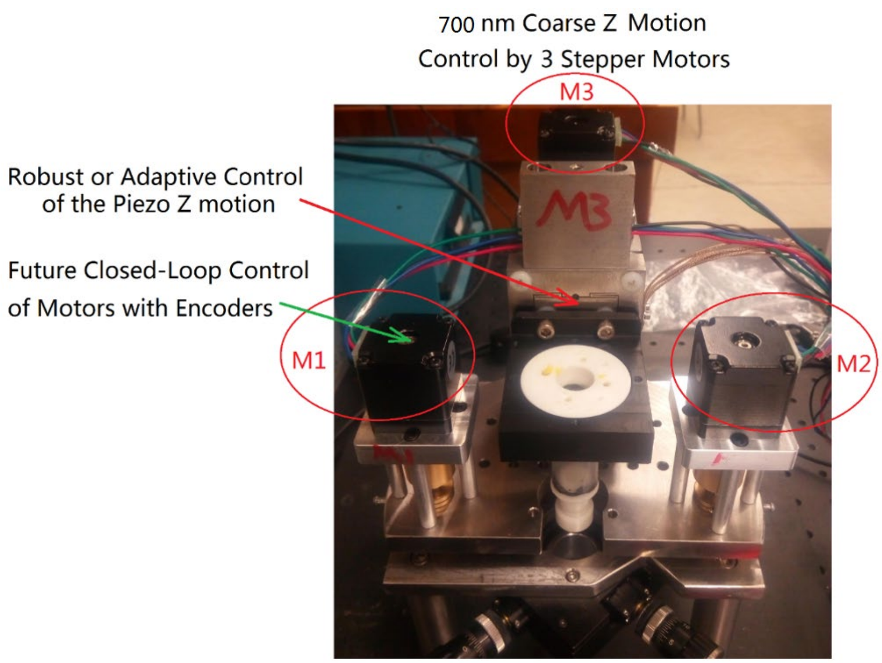

Fig 4, the front side view of the stage.

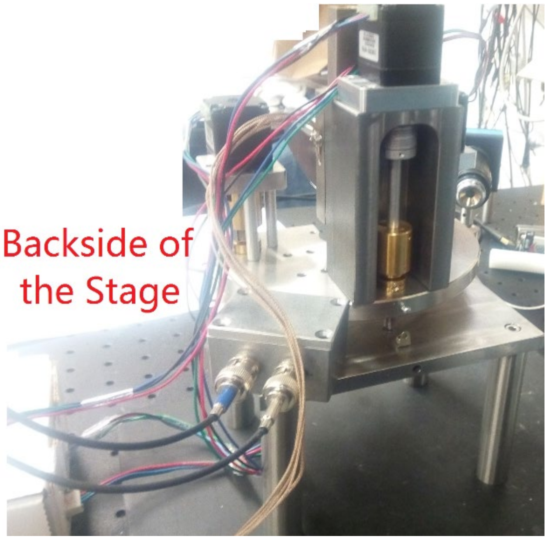

Fig 5, backside view of the stage.

3. The controller of the stage.

In order to lower the cost and make the controller portable, we build the controller on a protoboard with three A4988 motor driver boards and one 315 MHz RF module. See fig 6. Finally, the board is mounted on an Arduino microcontroller board. The motors have their own mechanical and electrical characteristics. It does not mean one can control the motors with any amateur motor driver boards. According to the load of the motor and the friction of the screws, compensate circuits are needed to couple the drive boards to the motors. For the details of the motor control, one can check Ref [1] [2] [3]. A good electrical board can make sure the motors will run smoothly, quietly, without warming up or skidding or jerk or loss of steps.

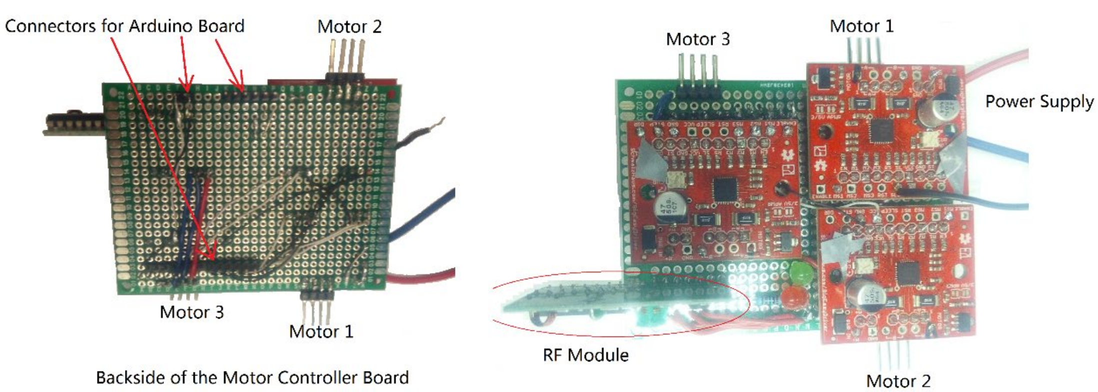

Fig 6, the control board of the three motors.

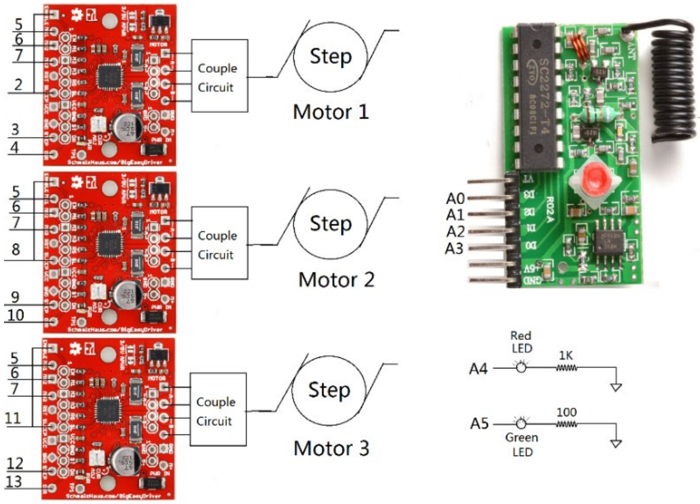

Fig 7, electric connection of the controller.

As shown in fig 7, the A4988 driver board inputs are connected with the Pin 2 to Pin 13 of the Arduino boards. Indeed, one can simplify the logic by a Karnaugh map or a similar technique. This way, 4 pins can be saved. The RF module is connected with A0 to A3 of the Arduino board. While A4 and A5 of the board are connected with 2 LEDs for display purposes. One can read Ref [4] [5] [6], to get more ideas about simplifying the logic and the spec of the A4988 chips. The RF module is paired with a four button controller. Usually, when one button is pushed, one line of the RF module will be pulled to High. The correspondence among the controller buttons and the RF data lines and the Arduino pins should be carefully verified. Here we have the correspondence like the one shown in fig 8.

Controller	A	B	C	D
RF Module	D2	D0	D3	D1
Arduino	A1	A3	A0	A2

Fig 8, signal correspondence between the controller and the Arduino pins.

4. The system structure of the SPM stage controller.

Modeling this relatively simple project, a 3 layer model is good enough. But for future integration with a large project, we still build the system with a 4 layer model. Here, the test schemes are on the 4th layer. The stage motion functions are on the 3rd layer. The instructions to control the specific motion of each motor are on the 2nd layer. The 1st layer is described in section 3 already. Finally, the system has the structure as shown in fig 9. The benefits of a 4 layer model are covered in Ref [7] [8].

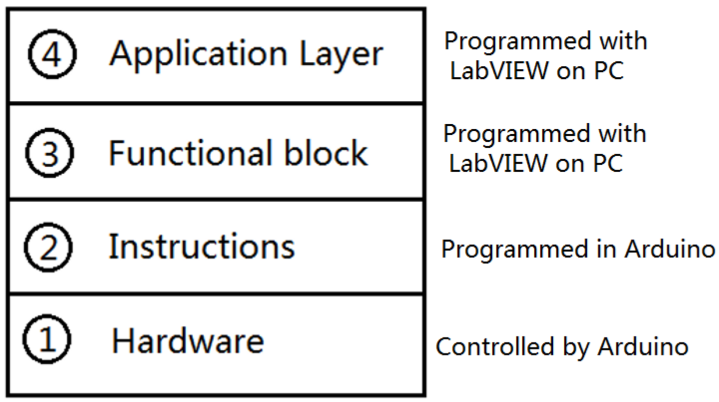

Fig 9, the system structure of the SPM stage motion controller.

Layer 4	Test Programs
Layer 3	LabVIEW FSM Controlling Stage Actions
	Human with An RF Controller
Layer 2	RF Signal Decoding Module
	A4988 Encoding Module
	Stage Action Module
	FSM for Computer and RF Communications
Layer 1	A4988 Motor Driver Boards
	RF Receiver Module
	Arduino Board
	LEDs
	4 Button RF Controller

Fig 10, detailed components of the system.

5. Layer 2 of the system.

Fig 10 shows the detailed system components on each layer. The layer 2 of the system is completely implemented in an Arduino UNO microcontroller board. The RF Signal Decoding Module builds a transparent bridge between the RF controller button actions and the Arduino instructions. The RF Signal Decoding Module will interpret the button pushing actions as different command sending to the Arduino board. Human behaviors and reactions are also integrated into the module already. This makes the RF control intuitive and easy.

The A4988 Module masks the detailed signal waveforms of the motor controllers. This module takes commands from the Stage Action module and actually controls the movements of the motors.

The Stage Action Module controls the stage to move down or up, fast or slow, continuously or one step, etc.

The Computer and RF Finite State Machine (FSM) is the interface between the Arduino board and the outside world. The FSM is running in a loop, taking commands from the 3rd layer of the system and passing them to the Stage Action module. Fig 11 shows the state diagram of the this FSM.

Not All of the USB Commands and Their Triggered States Are Shown on the Diagram See the List Below !

Cmd	States
LCN	Motor 3 Up
LCM	Motor 2 Up
LCL	Motor 1 Up
LCK	x1 Speed Up
LCJ	x5 Speed Up
LCI	x10 Speed Up
LCH	x20 Speed Up
LCG	Motor 3 Down
LCF	Motor 2 Down
LCE	Motor 1 Down
LCD	x1 Speed Down
LCC	x5 Speed Down
LCB	x10 Speed Down
LCA	x20 Speed Down
...	...

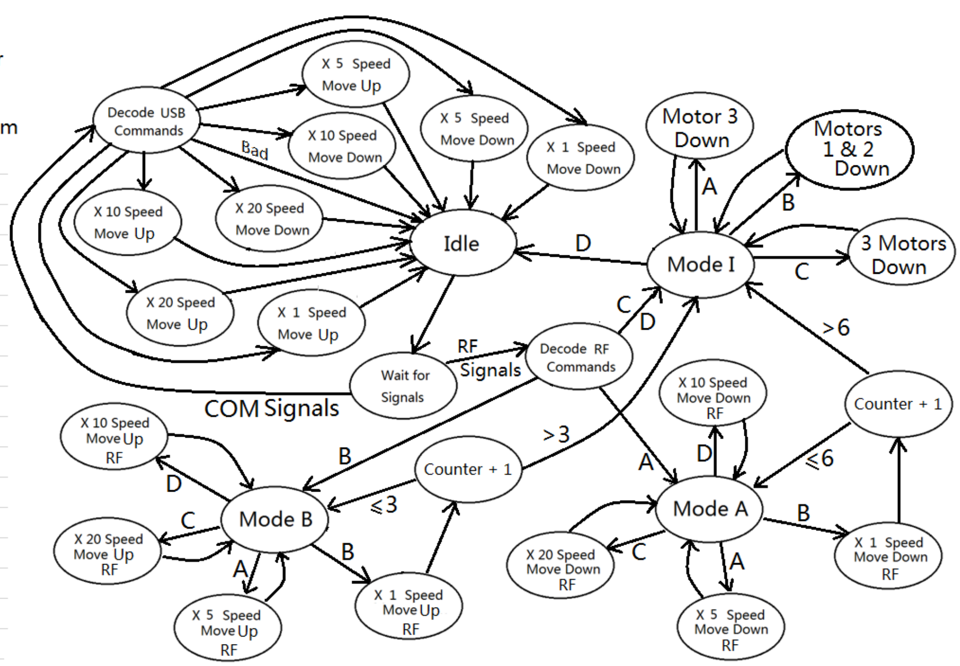

Fig 11, the state diagram of the FSM running in Arduino, which is the interface between the Arduino and the PC

The FSM also controls two 5 state LEDs to indicate the current state of the FSM. The 5 states are ON, OFF, BLINKING, FLASHING and DIM. See fig 12. Combining the two LEDs, one can get 10 to 25 state capacity. When the stage is put into a vacuum chamber or other controlled environments, the operator can not see the movement of the stage. This time, these state indicators are very important to show the motions of the stage.

Computer Control	Green ON
Menu D	RED ON
Menu Select	OFF
Function Is Running	Dim
Menu A,B,C	Blink
Stop,Quit	Flash

Fig 12, the 5 state display scheme of the LED indicators.

The FSM uses a 3-byte array cmd[3], to communicate with the computer. Every time, the FSM reads 3 bytes from the computer, then it will verify the first 2 bytes. If the first 2 bytes are L and C, the 3rd byte will be decoded. Then with different decoding results, different functions will be called to carry out the corresponding operation. See fig 13. The strategy to format the control commands can be found in Ref [7] [9].

```
void loop() {
//int  t=0, tA=0, tB=0, tC=0, tD=0;
char LowOverTimes=0;
char HighOverTimes=0;
char cmd[3];
while(1) {
 if(Serial.available()) {                        //LabVIEW control.
    // digitalWrite(A4,HIGH);                    // TEST POINT.
     for (char i=0;i<3;i++){
      cmd[i]=Serial.read(),delay(3);
     }
      if(cmd[0]=='L'&& cmd[1]=='C'){
       switch(cmd[2]) {
          case 0x41:  Low200St3();
                      break;
```

Fig 13, the FSM uses a 3-byte command to communicate with the computer.

6. Layer 3 of the system.

Due to space, we will not explain how to control the stage manually. One can get enough insight of the control schemes from fig 11.

The layer 3 programs are running on a host PC. a good example would be the one shown in fig 14. Two latch buttons control the continuous movement of the stage. When they are hit, the stage will move up or down continuously. When they are released, the stage will stop. M1, M2, and M3 are used to control the 3 motors separately, during the final approach. They can also be used to change the tilt of the stage. The LOCK button is used to terminate the mechanical movement control of the stage. When this button is hit, the 3 motors will be electrically disconnected.

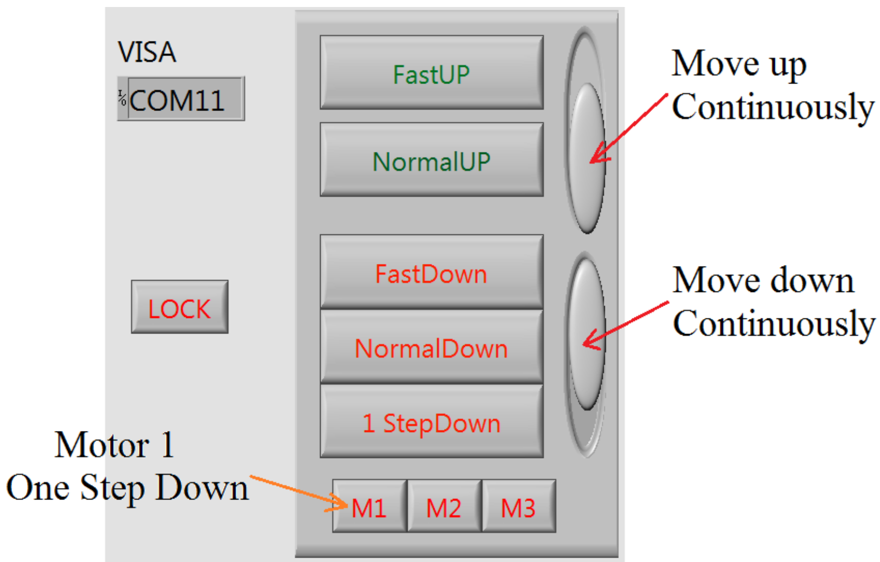

Fig 14, the user interface of the stage motion controller (Layer 3)

The Layer 3 program is also coded as an FSM. See fig 15. The good point of this programming method is that Layer 3 and Layer 2 can work with their own speed, and talk to each other asynchronously. This greatly simplifies the programming and makes the whole system stable and reliable. The methods about how to coordinate between layer 3 functions and layer 2 functions can be found in Ref [7] [9].

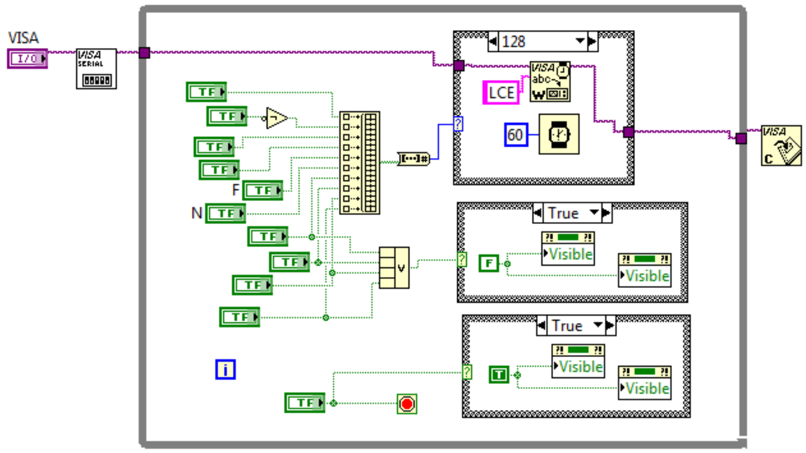

Fig 15,   the LabVIEW block diagram of the motion controller (FSM)

7. Test and Tune of the Whole System.

Skids, jerks, losing steps, noise and vibrations are always associated with the motor controlled system. Several adaptations are made in this project. Knowing the property of motors always helps a lot. The technique of motor ID and control can be found in Ref [1] [2] [3] [10].

A. The screws should be smooth enough. Make sure the motors are not overloaded. When the motors are overloaded, they may become very hot.
B. The driving current should be carefully adjusted. Make sure the motors are starting, running and stopping smoothly and quietly.
C. When the sample surface is engaged, the motors should be locked firmly. When the motors are not running, the driving currents will be turned off completely.

We tested the RF control and the computer interface extensively to make sure the stage can be controlled remotely. After careful tunes and tests, finally, we can get an open-loop 640 nm resolution mechanically. Since the piezo plate has a 60000 nm range, so this stage is good enough for any house made SPM applications.

8. Conclusion

We make a mechanically stable and accurate SPM stage driven by 3 stepper motors, which controls the mechanical approaches of the probe before the probe engages the samples. The stage has multiple accesses and controls through a computer interface or RF remote. The stage motion is optimized for tilt control. The system's 3rd layer is programmed with LabVIEW, which makes it easy to be integrated with other applications.

9. The Innovation in this project and future exploits.

We made an innovation in this project. An intelligent Finite State Machine (FSM) is built inside of the Arduino, which can communicate with a 4-button RF remote. With one or two button pushes, the 4 –button controller can control as many as 24 functions. With a little modification, this FSM can be used by other projects.

In this project, we demonstrate the capability of virtual instrumentation with Arduino and LabVIEW. Indeed, from the 4 facts listed below, one can build cheap and versatile virtual instruments combining Arduino and LabVIEW. Fig 14 shows the block diagram of a wireless virtual instrument implemented with Arduino and LabVIEW.

a) Arduino is much easier than traditional microcontroller programming. Arduino can be programmed either graphically or with a text C language. Transparent across many processor platforms.
b) Arduinos are cheaper and easier for prototyping.
c) LabVIEW on PC is not capable of real-time control. Arduino is not good for signal processing. Combining LabVIEW and Arduino, one can finish the whole process of lab automation with real-time control and sophisticated signal processing capabilities.
d) Combine RF module for remote control and data acquisition or wireless communication between human beings and the controlled targets.

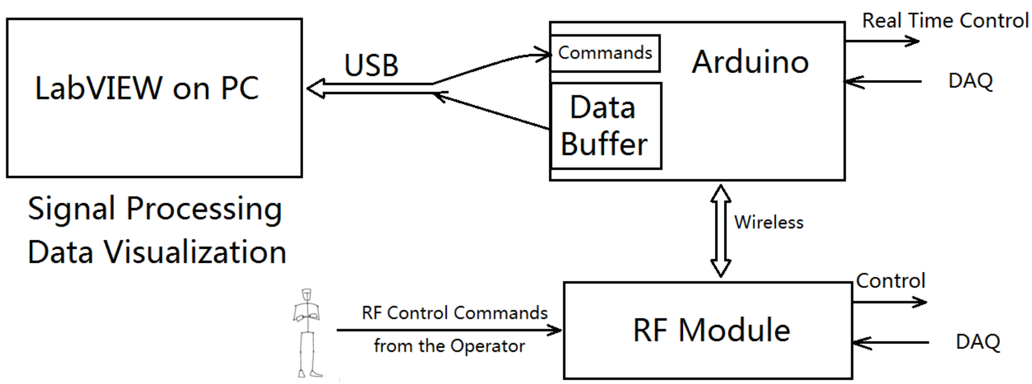

Fig 14, an illustration of a virtual instrument built with Arduino and LabVIEW.

References:

[1] Gene Franklin, Feedback Control of Dynamic Systems, Pearson Higher Education, Inc. 2010,

[2] Richard Dorf, Modern Control Systems, Pearson Higher Education, Inc. 2015

[3] Katsuhiko Ogata, Discrete-Time Control Systems, Pearson Higher Education, Inc. 2015

[4] John F. Wakerly, Digital Design: Principles and Practices (4th Edition) Pearson; 4th edition, July 31, 2005

[5] Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw-Hill Companies, 2012

[6] Allegro, A4988 DMOS Microstepping Driver with Translator and Overcurrent Protection, http://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers/A4988.aspx

[7] Jianghua Bai, Andres La Rosa, "Essentials of Building Virtual Instruments with LabVIEW and Arduino for Lab Automation Applications", International Journal of Science and Research (IJSR), https://www.ijsr.net/archive/v6i5/v6i5.php, Volume 6 Issue 5, May 2017, 640 – 644

[8] Jianghua Bai, Jingwei Chen, John Freeouf, Andres La Rosa, A 4-layer method of developing integrated sensor systems with LabVIEW, Journal of Measurement Science & Instrumentation,2013

[9] Arduino programming references, https://www.arduino.cc/en/Reference/HomePage

[10] Solarbotics, Industrial Circuits Application Note Stepper Motor Basics http://solarbotics.net/library/pdflib/pdf/motorbas.pdf