Interfacing of X-Ray Diffraction Machine using Arduino Uno

Rohith Saradhy

III Year Int. M.Sc.

School of Physical Sciences

National Institute of Science Education and Research.

Under the Guidance of

Prof. P.V SATYAM

Professor(H)

Institute of Physics, Bhubaneswar

Abstract:

X-rays in modern science has enormous number of application. It can be used in Radiography, Computed tomography, Fluoroscopy, Radiotherapy, X-ray crystallography, X-ray astronomy, X-ray microscopy, Airport Security etc. Here we interface a Stepper motor to a PC using an Arduino Microcontroller module, then interface HUBER SMC 9000 Stepper motor controller to a PC running on windows XP or higher through RS-232 port and using the interfaced Stepper motor take control of the Molybdenum XRD machine. The monochromator stage was also interfaced with a PC using an arduino which had DC motor controlled stage. Hyperlinks are provided to the source codes of the programmes and project files. Search for "HERE".

Introduction:

Stepper Motor:

A stepper motor (or step motor) is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor (an open-loop controller), as long as the motor is carefully sized to the application.

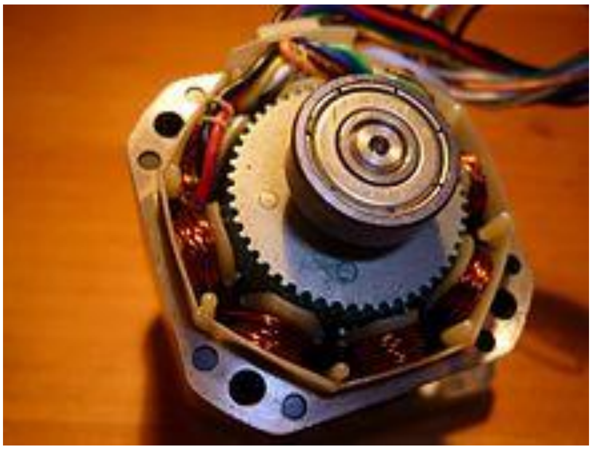


Figure 1: Stepper Motor

DC brushed motors rotate continuously when voltage is applied to their terminals. The stepper motor is known by its important property to convert a train of input pulses (typically square wave pulses) into a precisely defined increment in the shaft position. Each pulse moves the shaft through a fixed angle. Stepper motors effectively have multiple "toothed" electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, such as a microcontroller. To make the motor shaft turn, first, one electromagnet is given power, which magnetically attracts the gear's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. So when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one, and from there the process is repeated. Each of those rotations is called a "step", with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle.

There are two basic winding arrangements for the electromagnetic coils in a two phase stepper motor: bipolar and unipolar. We will be working with unipolar Stepper motor.

Unipolar:

A unipolar stepper motor has one winding with centre tap per phase. Each section of windings is switched on for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the commutation circuit can be made very simple (e.g., a single transistor) for each winding. Typically, given a phase, the centre tap of each winding is made common: giving three leads per phase and six leads for a typical two phase motor. Often, these two phase commons are internally joined, so the motor has only five leads.

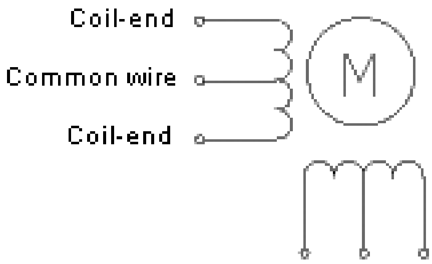


Figure 2: Unipolar Stepper Motor

A micro controller or stepper motor controller can be used to activate the drive transistors in the right order, and this ease of operation makes unipolar motors popular with hobbyists; they are probably the cheapest way to get precise angular movements.

Arduino:

Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. Current models feature a USB interface, 6 analogue input pins, as well as 14 digital I/O pins which allows the user to attach various extension boards.

Figure 3: Arduino

Introduced in 2005, the Arduino platform was designed to provide an inexpensive and easy way for hobbyists, students and professionals to create devices that interact with their environment using sensors and actuators. Common examples for beginner hobbyists include simple robots, thermostats and motion detectors. It comes with a simple integrated development environment (IDE) that runs on regular personal computers and allows users to write programs for Arduino using C or C++.

RS-232:

In telecommunications, RS-232 is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit-terminating equipment, originally defined as data communication equipment), such as a modem. The RS-232 standard is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin-out of connectors. The current version of the standard is TIA-232-F Interface between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, issued in 1997.

An RS-232 serial port was once a standard feature of a personal computer, used for connections to modems, printers, mice, data storage, uninterruptible power supplies, and other peripheral devices. However, RS-232 is hampered by low transmission speed, large voltage swing, and large standard connectors. In modern personal computers, USB has displaced RS-232 from most of its peripheral interface roles. Many computers do not come equipped with RS-232 ports and must use either an external USB-to-RS-232 converter or an internal expansion card with one or more serial ports to connect to RS-232 peripherals. RS-232 devices are still found, especially in industrial machines, networking equipment, and scientific instruments.

X-rays:

X-radiation (composed of X-rays) is a form of electromagnetic radiation. Most X-rays have a wavelength in the range of 0.01 to 10 nanometres, corresponding to frequencies in the range 30 petahertz to 30 exahertz (3×10^16 Hz to 3×10^19 Hz) and energies in the range 100 eV to 100 keV. X-ray wavelengths are shorter than those of UV rays and typically longer than those of gamma rays. In many languages, X-radiation is referred to with terms meaning Röntgen radiation, after Wilhelm Röntgen, who is usually credited as its discoverer, and who had named it X-radiation to signify an unknown type of radiation.

X-rays with photon energies above 5–10 keV (below 0.2–0.1 nm wavelength) are called hard X-rays, while those with lower energy are called soft X-rays.[4] Due to their penetrating ability, hard X-rays are widely used to image the inside of objects, e.g., in medical radiography and airport security. As a result, the term X-ray is metonymically used to refer to a radiographic image produced using this method, in addition to the method itself. Since the wavelengths of hard X-rays are similar to the size of atoms they are also useful for determining crystal structures by X-ray crystallography. By contrast, soft X-rays are easily absorbed in air and the attenuation length of 600 eV (~2 nm) X-rays in water is less than 1 micrometre.

There is no universal consensus for a definition distinguishing between X-rays and gamma rays. One common practice is to distinguish between the two types of radiation based on their source: X-rays are emitted by electrons, while gamma rays are emitted by the atomic nucleus. This definition has several problems; other processes also can generate these high energy photons, or sometimes the method of generation is not known. One common alternative is to distinguish X- and gamma radiation on the basis of wavelength (or equivalently, frequency or photon energy), with radiation shorter than some arbitrary wavelength, such as 10−11 m (0.1 Å), defined as gamma radiation. This criterion assigns a photon to an unambiguous category, but is only possible if wavelength is known. (Some measurement techniques do not distinguish between detected wavelengths.) However, these two definitions often coincide since the electromagnetic radiation emitted by X-ray tubes generally has a longer wavelength and lower photon energy than the radiation emitted by radioactive nuclei. Occasionally, one term or the other is used in specific contexts due to historical precedent, based on measurement (detection) technique, or based on their intended use rather than their wavelength or source. Thus, gamma-rays generated for medical and industrial uses, for example radiotherapy, in the ranges of 6–20 MeV, can in this context also be referred to as X-rays.

UART:

A universal asynchronous receiver/transmitter, abbreviated UART, is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. The universal designation indicates that the data format and transmission speeds are configurable. The electric signalling levels and methods (such as differential signalling etc.) are handled by a driver circuit external to the UART.

A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. UARTs are now commonly included in microcontrollers.

Interfacing Stepper Motor to PC through Arduino:

As we discussed earlier, we are using a Unipolar Stepper motor.

How Stepper Motors Work

Stepper motors consist of a permanent magnetic rotating shaft, called the rotor, and electromagnets on the stationary portion that surrounds the motor, called the stator. Figure 4 illustrates one complete rotation of a stepper motor. At position 1, we can see that the rotor is beginning at the upper electromagnet, which is currently active (has voltage applied to it). To move the rotor

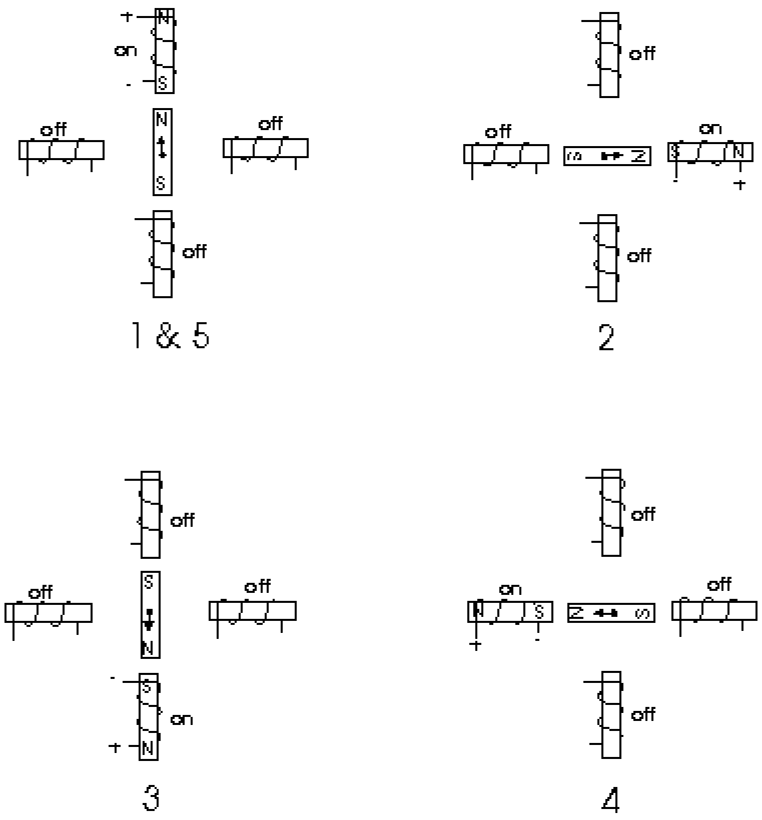


Figure 4

clockwise (CW), the upper electromagnet is deactivated and the right electromagnet is activated, causing the rotor to move 90 degrees CW, aligning itself with the active magnet. This process is repeated in the same manner at the south and west electromagnets until we once again reach the starting position.

In the above example, we used a motor with a resolution of 90 degrees or demonstration purposes. In reality, this would not be a very practical motor for most applications. The average stepper motor's resolution -- the amount of degrees rotated per pulse -- is much higher than this. For example, a motor with a resolution of 5 degrees would move its rotor 5 degrees per step, thereby requiring 72 pulses (steps) to complete a full 360-degree rotation.

You may double the resolution of some motors by a process known as "half-stepping". Instead of switching the next electromagnet in the rotation on one at a time, with half stepping you turn on both electromagnets, causing an equal attraction between, thereby doubling the resolution. As you can see in Figure 5, in the first position only the upper electromagnet is active, and the rotor is drawn completely to it. In position 2, both the top and right electromagnets are active, causing the rotor to position itself between the two active poles. Finally, in position 3, the top magnet is deactivated and the rotor is drawn all the way right. This process can then be repeated for the entire rotation.

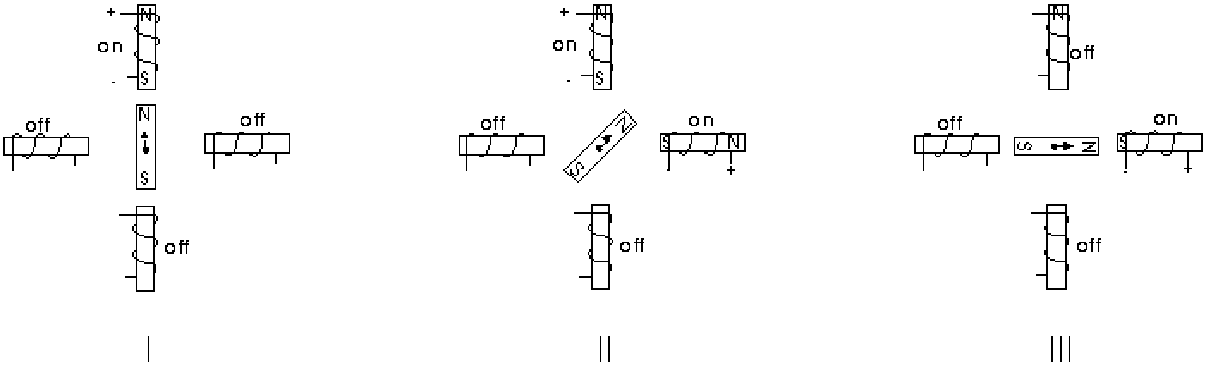


Figure 5

There are several types of stepper motors. 4-wire stepper motors contain only two electromagnets, however the operation is more complicated than those with three or four magnets, because the driving circuit must be able to reverse the current after each step. For our purposes, we will be using a 6-wire motor.

Unlike our example motors which rotated 90 degrees per step, real-world motors employ a series of mini-poles on the stator and rotor to increase resolution. Although this may seem to add more complexity to the process of driving the motors, the operation is identical to the simple 90 degree motor we used in our example. An example of a multipole motor can be seen in Figure 6. In position 1, the north pole of the rotor's permanent magnet is aligned with the south pole of the stator's electromagnet. Note that multiple positions are aligned at once. In position 2, the upper electromagnet is deactivated and the next one to its immediate left is activated, causing the rotor to rotate a precise amount of degrees. In this example, after eight steps the sequence repeats.

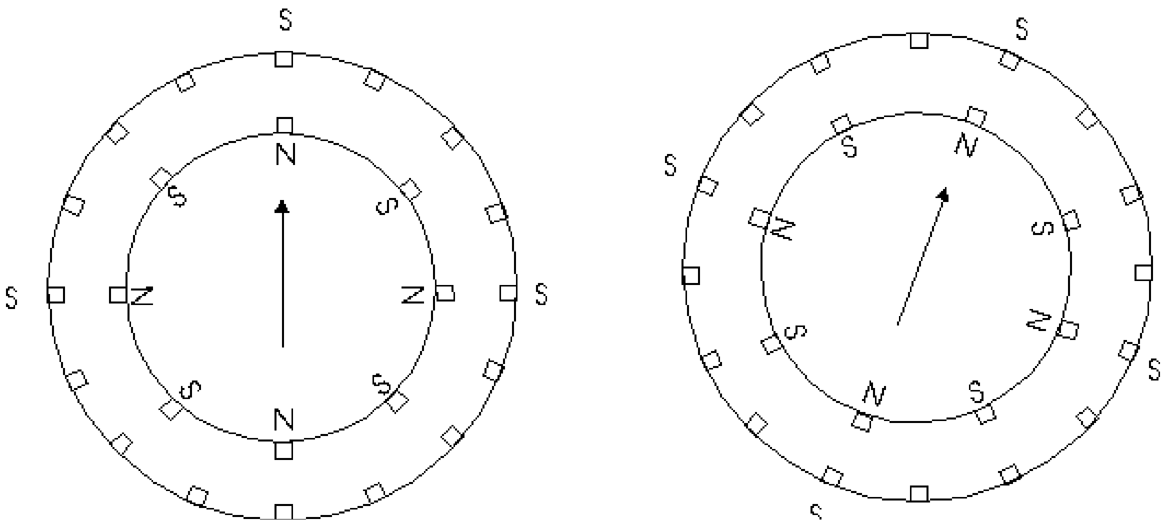


Figure 6

The specific stepper motor we are using for our experiments (ST-02: 5VDC, 5 degrees per step) has 6 wires coming out of the casing. If we follow Figure 7, the electrical equivalent of the stepper motor, we can see that 3 wires go to each half of the coils, and that the coil windings are connected in pairs. This is true for all four-phase stepper motors.

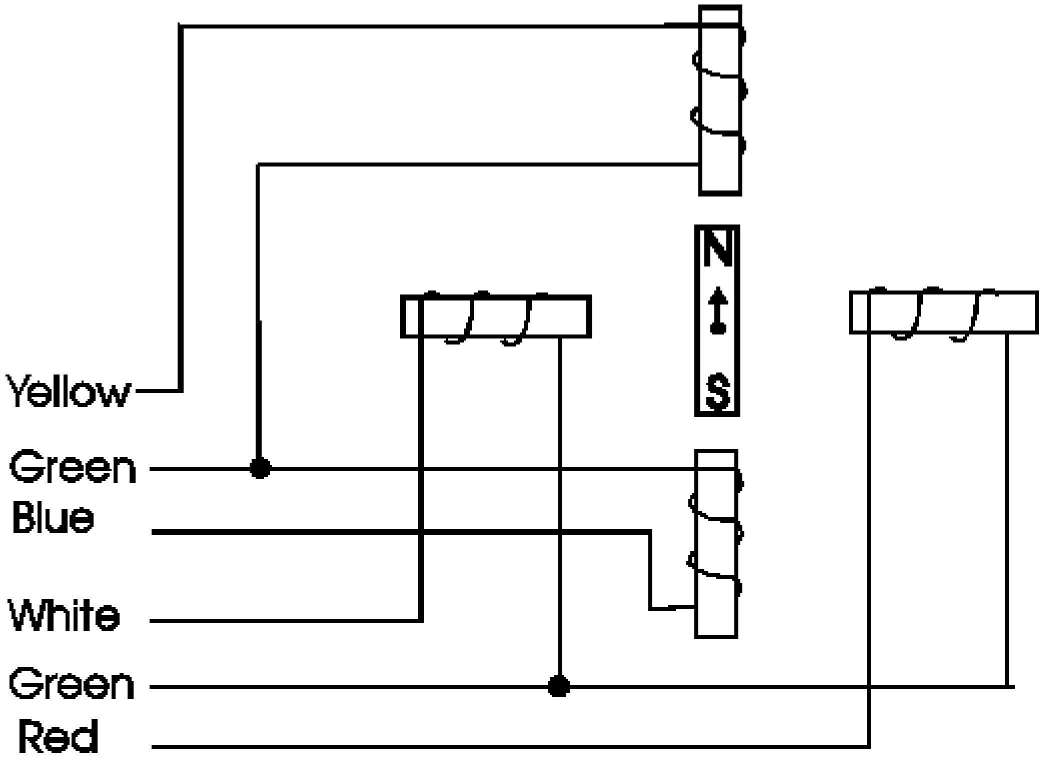


Figure 7

Creating the TTL Pulse

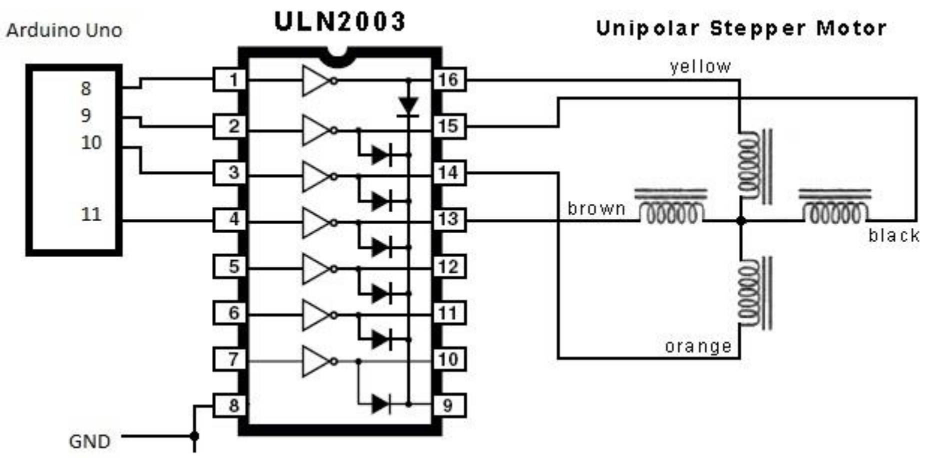
In order to drive the stepper motor, we need to create TTL pulses which will switch on the solenoid in a particular pattern. We achieve this by using Arduino. Arduino as mentioned earlier is a programmable microcontroller which has 13 digital pins (I/O) and 6 analogue inputs (each of 10 bit resolution). Here we will write a programme to utilize 4 of the 13 pins to generate a TTL pulse so that we can control the Arduino.

The programme can be found at **HERE** we are using pins 7, 8,9,10 of the digital pins)

This code will create a TTL pulse which will turn the stepper motor for 50 steps, furthermore, adjusting the “timedelay” and the “steps” variables in the programme we can change the speed and angle through which the motor moves.

Driving the Stepper Motor

Creating the TTL pulses is just the first part of the problem. The microcontroller produces around 10 mA of current, and normally Stepper motors requires around an ampere or more. How do we come around this problem? We do this by creating a driving circuit, which can be connected to an external power supply that would then be able to supply the necessary current to the stepper motor. An IC called ULN 2003 coupled with four relays can be used to create the driving circuit, the workings of which is beyond the scope of this project, but the circuit is as follows:



This will successfully drive the motor clockwise for 50 steps then anticlockwise for 50 steps.

Interfacing the Huber Motor Controller Series 9000

The basic unit is a single-axis model with a 16-bit NEC micro-controller and on-board memory. Configuration data are permanently stored on E 2 PROMs. It may be extended to up to a maximum of eight individually and independently controllable axes. Two standard interfaces for remote control are included: A serial (RS232 C) interface and a parallel GPIB (IEEE 488) interface. Additionally, an opto-coupled 16 bit I/O-port is available to control- or react on external events. For our purpose we will use RS232 Serial channel with 9600 baud rate.

The commands for remotely controlling this unit can be found **HERE.**

I wrote a simple programme in visual basic to control the HMC using the HUBER Manual, in the theta-two-theta configurations for getting the Bragg peak . The project can be found **here**.

Interfacing the Arduino to control the Monochromator table.

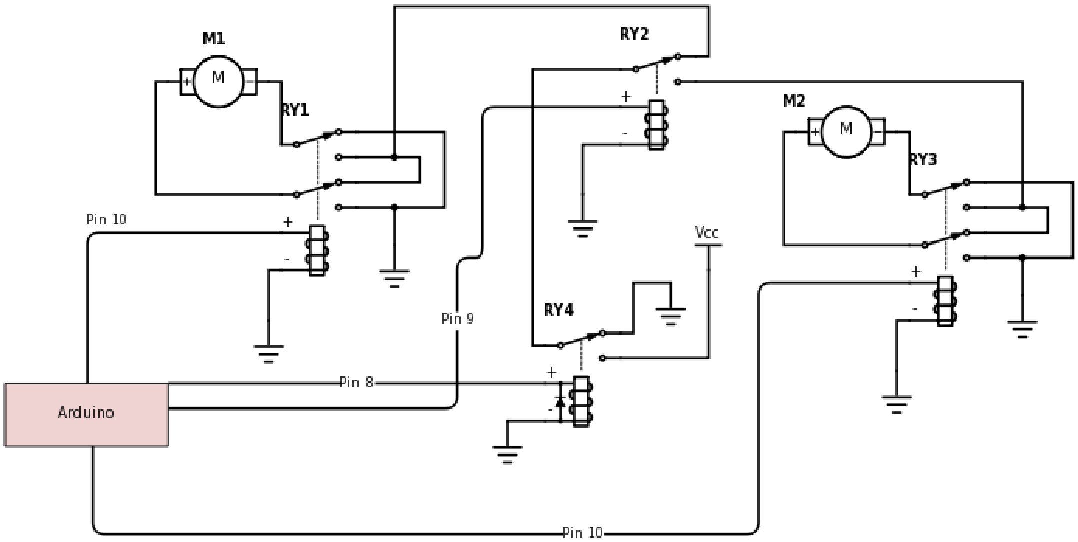


Figure 8. Schematic of the DC motor controller.

The monochromator table of the XRD already had DC motorised stage. I connected the motors to relays according to the following figure 8. RY1 and RY3 are relays which change the polarity of the DC motors. RY2 relay is used to choose which motor to functions. And we use RY4 to switch on and off the motor.

The monochromator table was calibrated to determine relationship of time of function was the degrees rotated (circuit delays included). **Here** is the programme which collects the time required to rotate in milliseconds, and the degree rotated can be read off the scale on the bench. The relationship was found to be as follows after plotting a graph of time vs degrees rotated.

$$\textbf{Time(milliseconds)=13370*angle(degree)-3874} \qquad \textbf{(1)}$$

This relationship was used to write the programme to rotate the bench by specific degrees. The arduino programme is given **HERE**. Note that PushBtn(Pin 8) is the button for switching on and off the motors which is connected to relay RY4. With Motor_selector(Pin 9) we can select which motor to run. And with FlipSwt(Pin 10) we can reverse the polarity of the motors.

Commands for the software.

The software works by sending it serially (through USB port) at 9600 baud rate **{0/1} {+/- degrees to move.}** The 1/0 will select the motor to move and +/- the direction to move and degrees the degrees to move according to equation (1). You can you any software, to send the serial command to the arduino. Note you have to select the COM Port before initialising communication.

Example:

If I want to rotate motor 1 by +2 degree, then I will serially send the following command.

1 +2

Note that there is a space between 1 and plus sign but no space between plus and two. If I want to send motor 1 back by 1 degree, then:

1 -1

Discussion:

Interfacing can be used as a powerful tool to automate mundane tasks, as well as implement new experiments. The project introduced the idea of interfacing between electronics, as well as electrical appliances to electronics. Now it's easy, more than ever to interface between electronics because of prototyping boards like arduino. As a side task I had used the knowledge of stepper motor to interface the X-Y sample stage in the micro-beam line of the Ion Beam Lab of Institute of Physics to a PC using an arduino since its original stepper motor driver broke down. This was done with a same circuit as of the ULN2003 driver IC. Arduino can be used as a cheap solution for these types of problems. Furthermore, I implemented an analogue triple-axis accelerometer to measure the angle that the monochromator bench is subtending with respect to the horizontal ground. The monochromator arduino programme was sensitive enough to be moved 0.01 degree which is really good, although it could not be detected on the accelerometer. After following the manual of HMC 9000, it is very easy to control it using serial RS232 channel. When I was writing the programme, it didn't work initially as I was not sending the return character after each command. After making a serial port listener circuit using arduino, and using the original software provided by the company I was able to realize this problem and rectify it.

Reference:

- Microprocessors and Microcomputers by V.M Rooney and A.R Ismail.
- Interfacing Microcomputers to the Real World by Sargent and Shoemaker.
- X-Ray standing wave and ion scattering studies of metal-semiconductor interfaces and epitaxial layers by P.V Satyam.
- https://www.arduino.cc/en/Reference/HomePage
- http://homepage.cs.uiowa.edu/~jones/step/physics.html
- https://courses.cs.washington.edu/courses/cse466/02au/Lectures/steppermotors.pdf
- Instrument Engineers' Handbook: Process Control and Optimization by Liptak, Bela G.