

LabView Interface with Arduino Robotic ARM

B. Harish¹, U. Jyothsna²

¹Vizag Institute of Technology, Visakhapatnam, India

²Assistant Professor at Vizag Institute of Technology, Visakhapatnam, India

Abstract: The main focus of this work was to design, develop and implementation of competitively robot arm with enhanced control and stumpy cost. To control the robot we used Lab view, which performs inverse kinematic calculations and communicates the proper angles serially to a microcontroller that drives the servo motors with the capability of modifying position, speed and acceleration.

Keywords: Relay, 12V Battery, DC Motors, and ARDUINO, Gripper mechanism, worm gear, spur gear, transistors, UART Communications.

1. Introduction

A branch of technology that deals with a robot is Robotics. A robotics is a mechanical or virtual intelligent agent that can perform tasks automatically or with guidance, typically by remote controlled. In practice a robot is usually an electro mechanical machine that is guided by computer, and electronic programming robot can autonomous or semi autonomous. The term robotics is practically defined as the study, design and use of robot systems for manufacturing. Robots are generally used to perform unsafe, hazardous, highly repetitive, and unpleasant tasks. There are mainly two different kinds of robots: a service robot and an industrial robotic. Industrial robot is officially defined by ISO as an automatically controlled and multipurpose manipulator programmable in three or more axis. Industrial robots are designed to move material, parts, tools, or specialized devices through variable programmed motions to perform a variety of tasks. An industrial robot system includes not only industrial robots but also any devices and/or sensors required for the robot to perform its tasks as well as sequencing or monitoring communication interfaces. So, Due to increase using of industrial robot arms, we have developed a LIFA to control robotic ARM.

2. System Analysis

The main objective of this, Lab View Interface for Arduino (LIFA) allows users to control robotic arm without using the DAC system and acquire data through an Arduino microcontroller using the graphical programming environment Lab View.

3. Block Diagram



Figure1: Block diagram

The above block diagram shows the overall co-ordination of the LIFA robotic ARM. Design the front panel, and back panel in the graphical view of Lab View. The output values from LabView are given as input to Arduino controller through UART Communications. The output values from the controller are used to drive the motors using relays.

3.1 Hardware Required

- ARDUINO UNO
- Relays
- DC Motor
- 12V Battery

3.2 Software Required

- Arduino IDE
- Lab View

3.3 Hardware descriptions:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

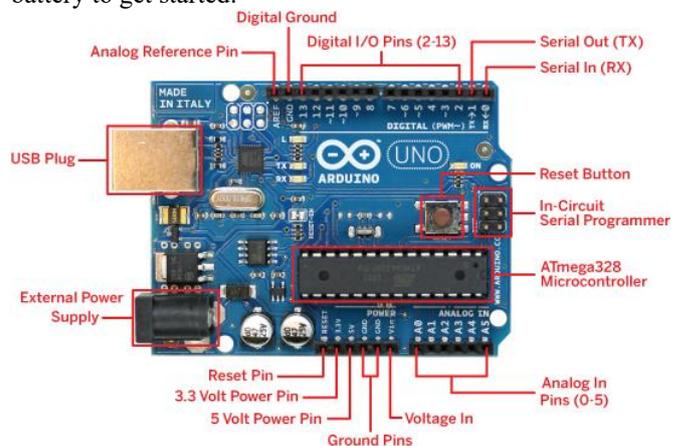


Figure 2: Arduino features

3.4 Tables

Table 1: Margin specifications

sno	Descriptions	Controller
1	Microcontrollerr	At mega 328p
2	Operatingvoltage	5v
3	Digital I/O Pins	14 & provide PWM output)
4	Analog In pins	6
5	Flash	32kb
6	SRAM	2kb
7	EEPROM	1kb
8	Clock speed	16Mhz

3.5 Sections Headings

Section headings come in several varieties:

- 1) first level headings: 1.Introduction
- 2) second level: 2. System Analysis
- 3) third level: 3 Hardware Required
- 4) forth level: 4. Making of the project
- 5) fifth level: 5. DC motors
- 6) sixth level: 6. Battery

3.6 Relay

Relay is an electromagnetic device which is used to isolate two circuits electrically and connect them magnetically. They are very useful devices and allow one circuit to switch another one while they are completely separate. They are often used to interface an electronic circuit (working at a low voltage) to an electrical circuit which works at very high voltage. For example, a relay can make a 5V DC battery circuit to switch a 230V AC mains circuit. Thus a small sensor circuit can drive, say, a fan or an electric bulb.

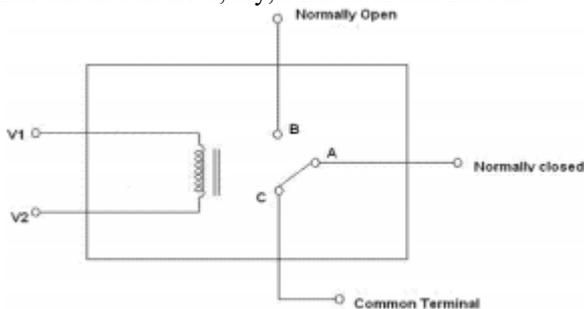


Figure 3: Relay operation

These device is consist of a coil of wire is covered with an iron core. When electricity is applied to a V1 & V2 means coil of wire it becomes magnetic, hence the term electro magnet. The A, B & C terminals are SPDT Switch controlled by an electro magnet. When electricity is applied to v1 & v2, the electromagnet acts upon the SPDT switch so that the B, C terminals are connected. When the electricity is disconnected, then A, C terminals are connected. If electricity is disconnected, some current is present in that coil. During turn off the reverse EMF doesn't worked means, we are using flywheel diode. This diode doesn't allow the current in the previous block.

4. Making of the Project



Figure 4: Attaching the Motor for base Board

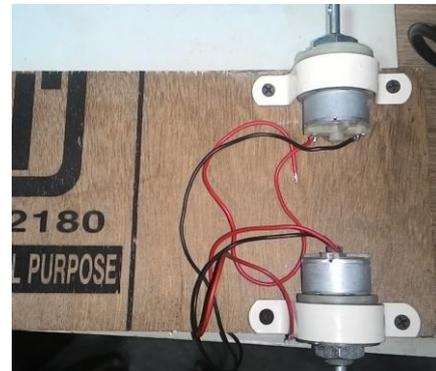


Figure 5: Two DC motors 3kg 30rpm



Figure 6: Worm Gear for 12v 5Kg motor



Figure 7: Overall View of the project

5. DC Motors

An electric motor is an electromechanical device that converts electrical energy into mechanical energy. Most electric motors operate through the interaction of magnetic fields and current carrying conductors to generate force. A DC motor is an electric motor that runs on direct current (DC) electricity. DC motors were used to run machinery, often eliminating the need for a local steam engine or internal combustion engine. Dc motors can operate directly from rechargeable batteries, providing the motive power for the first electric vehicles.

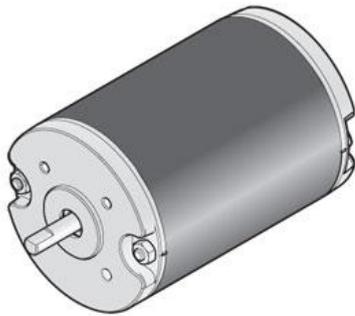


Figure8: DC motor

DC motors are found in applications as small as toy and disk drives, or in large sizes to operate still rolling mills and paper machines. Modern Dc motors are nearly always operated in conjunction with power electronic devices.

6. Battery

The above figure shows the DC Rechargeable battery which provides 12V DC voltage.



Figure 9: DC battery

The Arduino integrated development environment (IDE) is a cross platform application written in Java, and is derived from the IDE for the processing programming language and the wiring projects. It includes the code editor with features such as syntax highlighting, brace matching, and automatic identification, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

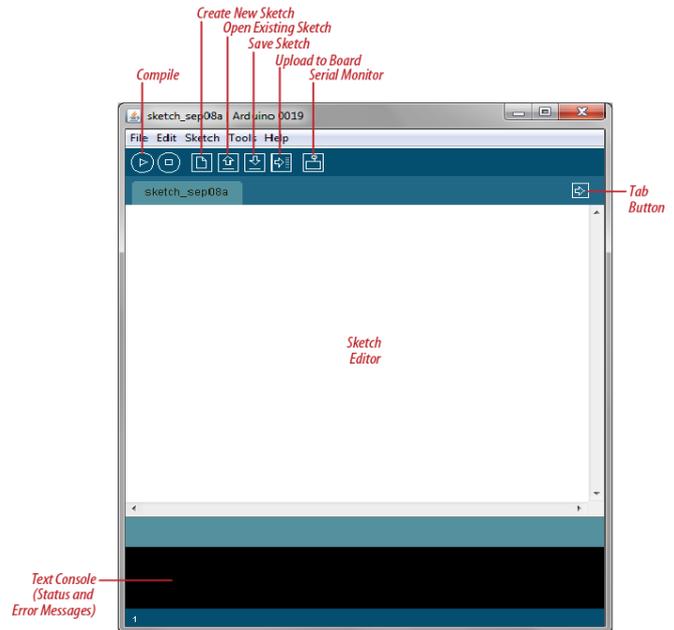


Figure 10: Arduino Compiler

6.1 LABVIEW

The NI LABVIEW software is used for a wide variety of applications and industries. LABVIEW is a highly productive development environment for creating custom applications that interact with real-world data or signals in fields such as science and engineering. The net result of using a tool such as LABVIEW is that higher quality projects can be completed in less time with fewer people involved. So productivity is the key benefit, but that is a broad and general statement. It is unique because it makes this wide variety of tools available in a single environment, ensuring that compatibility is as simple as drawing wires between functions. It is a development system for industrial, experimental, and educational measurement and automation applications based on graphical programming, in contrast to textual programming - however, textual programming is supported in it. It has a large number of functions for numerical analysis and design and visualization of data. It now has several toolkits and modules which brings it to the same level of functionality as MATLAB and Simulink in analysis and design in the areas of control, signal processing, system identification, mathematics, and simulation, and more. The programming language used in LABVIEW, also referred to as G, is a dataflow programming language. Execution is determined by the structure of a graphical block diagram (the LV-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallel execution. Multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler, which multiplexes multiple OS threads over the nodes ready for executions. LABVIEW ties the creation of user interfaces (called front panels) into the development cycle. It programs/subroutines are called virtual instruments (VIs). Each VI has three components: a block diagram, a front panel and a connector panel. The last is used to represent the

VI in the block diagrams of other, calling VIs. Controls and indicators on the front panel allow an operator to input data into or extract data from a running virtual instrument. However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface, or, when dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the given node through the connector panel. This implies each VI can be easily tested before being embedded as a subroutine into a larger program. The graphical approach also allows non-programmers to build programs by dragging and dropping virtual representations of lab equipment with which they are already familiar. The LABVIEW programming environment, with the included examples and documentation, makes it simple to create small applications. This is a benefit on one side, but there is also a certain danger of underestimating the expertise needed for high-quality G programming. For complex algorithms or large-scale code, it is important that the programmer possesses an extensive knowledge of the special LABVIEW syntax and the topology of its memory management. With LABVIEW, it is very easy to program different tasks that are performed in parallel by means of multithreading. This is, for instance, easily done by drawing two or more parallel while loops. This is a great benefit for test system automation, where it is common practice to run processes like test sequencing, data recording, and hardware interfacing in parallel. Due to the longevity and popularity of the its language, and the ability for users to extend the functionality, a large ecosystem of 3rd party add-ons has developed through contributions from the community. This ecosystem is available on its Tools Network, and is a marketplace for both free and paid LABVIEW add-ons. There is a lowcost LABVIEW Student Edition aimed at educational institutions for learning purposes. There is also an active community of LABVIEW users who communicate through several e-mail groups and Internet forums.

6.2 UART Communication

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 signaling levels and methods (such as differential signaling 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. A dual UART, or DUART, combines two UARTs into a single chip. An octal UART or OCTART combines eight UARTs into one package, an example being the NXP SCC2698. Many modern ICs now come with a UART that can also communicate synchronously; these devices are called USARTs(universalsynchronous/asynchronousreceiver/transmitter).

7. Results

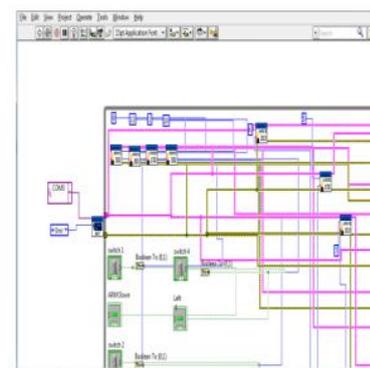
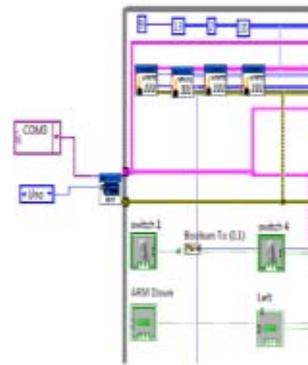


Figure 11: Front Panel

References

- [1] labviewhacker.com/doku.php?id=learn:libraries:lifa:getting_started.
- [2] decibel.ni.com/content/groups/labview-interface-forarduino
- [3] Wikipedia.org/wiki/relay [4]. programming arduino by simon monk.
- [4] programming arduino by simon monk
- [5] wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter