Arduino Based Control And Data Acquisition System Using Python Graphical User Interface (GUI)

Atal Khattak, Muhammad Masood Ahmad, Farid Ullah Khan

Abstract: This paper presents the development of a control and data acquisition system for a machines and equipments.The system is highly beneficial and applicable for internet of things (IoTs) environment. By utilizing the devised system one can sit on a computer and can use a specially made Graphical User Interface (GUI) to control a process and obtain data through sensors. Tkinter toolbox in Python language libraries is used to create the GUI, while Arduino acts as intermediary between the system and the computer. The control section of the GUI is used to control speed of two DC motors through an H-bridge drive module and the data acquisition section is used to obtain data from an accelerometer. The developed system and software is successfully tested for the speed regulation and acceleration measurement for DC motors. In the system, microcontroller reads data and transmits it to the computer using USB connection. Once on the computer data is graphically represented then graphs and data can be saved on the computer or forwarded to another computer using email.

Index terms: Acceleration monitoring, DC motor, graphical user interface, python programming, speed control.

1. INTRODUCTION

With the advancement of computers, digital control, control system and internet of things (IoT) [1] more and more industries are focusing heavily upon the utilization of computers for control tasks and data acquisition [2, 3]. With the computational power of modern day computers, equations that would take hours can now be solved in seconds. Digital Controls are also more accurate and make less errors than the traditional human-controlled manual controls. Nowadays, sensors also rely heavily on electronics and provide output in digital or analog electronic signals [4]. These signals can be fed directly into an electronic system that can store the data and/or use it as feedback for the control operation that the system is operating on. With these advancements a single system can handle both process control and data acquisition. Nowadays, wireless sensor nodes (WSNs) [5] are used to acquire systems data for monitoring and sensing. Moreover, work is also being performed to make these WSNs autonomous by integrating with energy harvesters [6, 7, 8, 9].The most popular digital control system being used is a computer. Through a combination of software and compatible hardware one person can monitor the entire process on his personal computer (PC) sitting in his home. Moreover, the person can provide parameters for the control operations and at the same time check the system regularly for errors or irregular patterns.

- *Author Atal Khattak: Department of Mechatronics Engineering*
- *Co-author Muhammad Masood Ahmad: Department of Mechanical Engineering*
- *Co-author Farid Ullah Khan: Department of Mechatronics Engineering.*
  E-mail: dr_farid_khan@uetpeshawar.edu.pk
  University of Engineering and Technology,
  Peshawar, Pakistan.

On the computer, these applications are usually controlled by software programs developed using various general purpose programming languages, such as, Assembly, BASIC, C, C++, C#, Java, LabVIEW, etc. As shown in Figure 1, the computer communicates with a microcontroller, programmable logic controller (PLC) or other such compatible digital control systems via data cable (like USB, RS232 etc.), wireless networks (e.g. ZigBee) or the internet (using web-based GUI). Moreover, the controller manipulates the operation of the actuator to manage the process and the information of the important parameters of the process are monitored through sensors. The sensors data is utilized as a feedback for the control system as well as it is supplied to the computer for information for any check and balance.

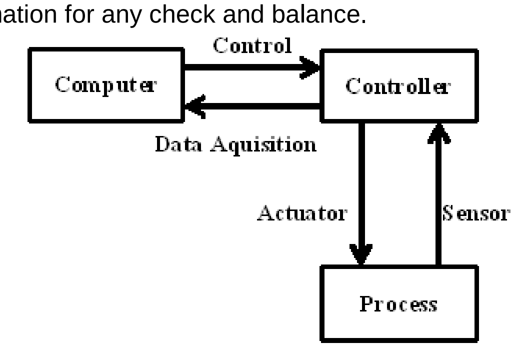


Figure 1. *Schematic diagram of system model for automation and control of a process*

2. Control and Data Acquisition

DC Motors are the most basic electrical devices used as actuators in smart and autonomous machines in industry. They have a wide range of applications from domestic appliances, like washing machines, fans, dryers and microwaves etc., to heavy industrial machine applications in assembly lines, lathe machines, milling machines, drilling machines and pumps. The wide range of applications of DC motors comes from the fact that by controlling the input voltage and current to the DC motor [10], one can control the speed, torque and direction of rotation of a motor. Because of the simplicity of the control of DC motors they are the best choice of actuators for smart, intelligent and automated machines and equipment where just an electronic control is used to produce any desirable motion, speed, force or torque. As shown in figure 2, the speed regulation and monitoring (through vibration) of the DC motor is depicted.

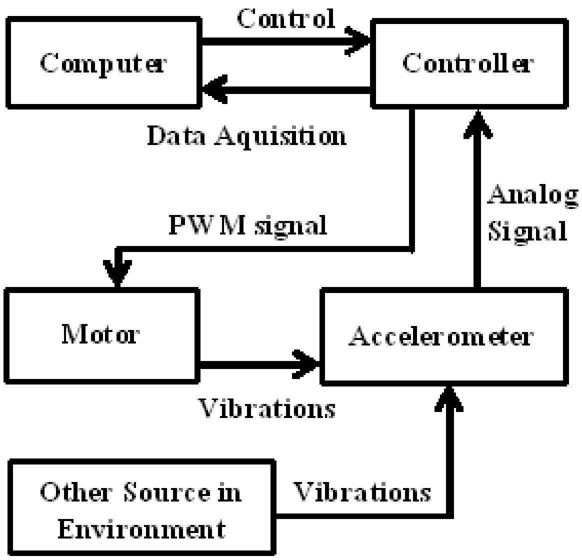


Figure 2. Control and Data Acquisition for our model

3. Literature review

Due to the usefulness of DC motors, a lot of work has been done in the past on the control of DC motors. Data acquisition for a process, machine or equipment is also a research field of interest where much work has been performed previously. Sensors are employed to obtain information on position and speed of a DC motor to form a closed loop control system in[11]. A GUI is created using LabVIEW software and the control is implemented on hardware using ATMEGA32 microcontroller. It used optical encoders to obtain information on the angular position and speed of the motor. This data is transmitted to the computer from the microcontroller, via ZigBee protocol, and is then compared with the desired position and speed input by the user using GUI. A ZigBee wireless communicator is implemented to wirelessly control the DC motor directly from the computer[12]. This work targets industrial environment where wireless communication is an integral part of the communication system. The computer is interfaced with a ZigBee wireless communicator. The computer performs analysis on the data and creates a backup internally providing real-time monitoring, control and protection of the system. The sensors are interfaced with an 8-bit AVR microcontroller to take data like voltage, current, winding temperature, rpm and torque, and transmit that information to computer using IEEE 802.15.4 standard and ZigBee protocol. X-CTU software designed to work with RF transmitters is used as GUI in the computer. A remote monitoring and control system for DC motor has been developed in[13]. The setup involved the design of remote start/stop, control and monitoring system of a DC motor through computer interface using a ZigBee wireless motor control module. The module also included the continuous online monitoring of the motor's parameters, such as, current, voltage, temperature, speed through a radio frequency (RF) data acquisition system and storing the data in a database designed using Visual basic. The setup is oriented towards improving the remote controlling abilities of the system, while keeping the hardware requirements minimum. It used IEEE 802.15.4 standard to interface sensors with AVR microcontroller and communicated with computer using ZigBee protocol. A peripheral interface controller (PIC) microcontroller has been used to communicate with a computer based MATLAB program in[14].The PIC microcontroller controlled both the speed and direction of the DC motor based on the data it received from the computer using serial communication. The user entered the desired values of speed and direction to the computer using a MATLAB based GUI. This system is able to control both a stepper motor and a DC motor at the same time using the same program and setup. A speed control and monitoring system for a DC motor (12 V, 1000 rpm) has been reported in[15]. Speed control is done with the help of pulse width modulation (PWM) pins on Arduino board and H-bridge IC. Feedback speed monitoring is based on infrared (IR) pair based interrupt monitoring. The computation on the sensor's data are performed by Arduino board. On the computer side a GUI is developed in C# language. The user interacts with the program by controlling the speed using a track bar and observe speed data in rpm using the text window. In the work discussed in[16] a DC motor speed control and monitoring system is developed. The hardware and software for DC motor speed control has been produced. The microcontroller has been connected to the computer using a 32-bit parallel port connector. There are different maximum speed (rpm) for different load conditions and based on this information a control algorithm has been developed. The output of the given system is observed using the GUI developed in LabVIEW and the coding is performed in the Visual Basic. The speed and direction control of a DC motor has been discussed in[17]. In this work the GUI has been developed using visual basic 6.0 programming language on the computer. The computer is connected with the PIC18F452 microcontroller through a serial communication. The analysis on the data is performed within the microcontroller program. The motor speed is varied using the PWM of the microcontroller. Speed, direction and ON/OFF state of the motor is also controlled with the developed system. This work developed a motor control and data acquisition system regulate and monitors the machine, process or equipment. The devised system monitors sensor's data values and simultaneouslycan modify motor speed using a computer based graphical user interface (GUI). Work has been done previously on such systems, they used different controllers, modes of communication and programming languages (for developing GUI) to monitor different working parameters of motor operation and control motors based on data acquired from these sensors. These parameters include motor speed, temperature and position. This work monitors motor health and safety by analyzing data on vibrations caused by the motor and by other sources in the environment of the motor. Accelerometer measures the amplitude and frequency of acceleration. Microcontroller reads data and transmits it to the computer using USB connection. Once on the computer data is graphically represented then graphs and data can be saved on the computer or forwarded to another computer using email.

4. Python Programming

In this work for the computer based graphical user interface (GUI), a python toolbox tkinter is used. Python is a general purpose high level programming language favoured by companies, like Google, because it is easy to use and much more comprehensive than any other language. Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible. Python can also be embedded in existing applications that need a programmable interface. Its comprehensive libraries like, matplotlib, pyserial, pylab, py2exe and tkinter offer the solution to almost any programming problem which any one can think of. Another useful feature of python is the fact that, the code can be compiled into an executable exe file that can be utilized on almost any computer without having to install python libraries on the computer since most operating systems come with inbuilt python readers. The GUI has the following features:

4.1 Menus of GUI

The program has three main menus as shown in Figure 3. Using the options menu users can save the data, open or close the serial port, set the port address and baud rate for serial communication and safely exit the program. Moreover, using the playback menu users can start, stop or pause any one of the two motors. The help menu provides user with the information how to use this program.

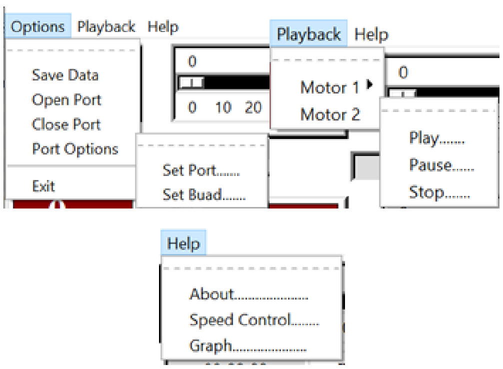


Figure 3. Menus in Python GUI

4.2 Sliders of GUI

The program creates two sliders (as shown in Figure 4) with which users can control the speed of motors as a percentage of top speed. This allows compatibility with almost any motor type regardless of the motor speed rating. Furthermore, the time duration for which the control is applied is displayed below the sliders, while the percentage of speed (the point of the slider is at), is displayed on the left side of the slider. Additionally, the speed control is not implemented until the done button on the right of the slider is not clicked, if done is not clicked then the user can have the option of returning the slider to the last implemented speed point using the revert button. The playback buttons for each motor are displayed below the sliders to the right of time display.

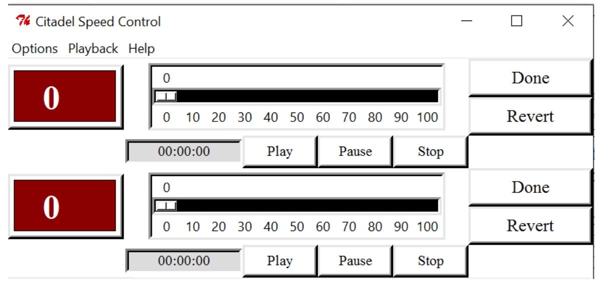


Figure 4. Speed Control Display

4.3 Graph Display in GUI

The two graphs time-vs-acceleration and the frequency-vs-acceleration and their options are displayed below the sliders in two different tabs as shown in Figure 5. The graph manipulation widget is displayed at the bottom of the graph and the graph options, playback, save data and extract data (from data saved in Arduino), are displayed to the right of the graph. The graphs are updated every 0.1 sec and each of the graphs has their own separate options. Due to the method of graph generation used by python programming the data is displayed on the graph after a specific time lag. This lag depends upon how much data is displayed at any point in time and how long it takes for the graph to update. Currently, the program has been set to display 5 min of data and update the data every 30 sec. It must be noted that this has no effect on the data being received by the computer, i.e., the computer receives real time data without any lag thus any control algorithm based upon the data will be executed in without any lag.

(a)

(b)

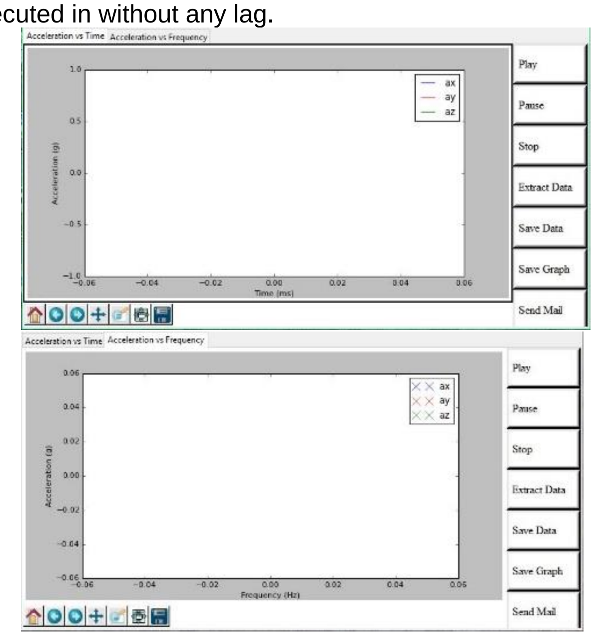


Figure 5. Data acquisition graph display: (a) acceleration vs time, (b) *acceleration vs frequency*

4.4 Saving Data on Computer with GUI

The program also saves data on the computer in "C:\Users\username\Documents\Citadel" directory, this procedure is shown in Figure 6. If the folder does not exist, the software creates the folder. It saves the data in .xls format to be opened using Microsoft excel spreadsheet. One sheet for acceleration vs time data and other for acceleration vs frequency data. The software can also save both graphs in .jpeg picture format in the same folder. If the graph toolbar is used to save the graph, then only the graph currently displayed will be saved. But if the 'Save Graph' button on the right side is used then it will save graph for the entire data-set currently saved within the computer.

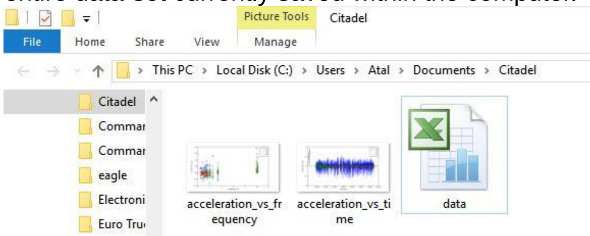


Figure 6. Data folder in documents for data saving in computer.

4.5 Sending data using email with GUI

Another good feature in this program is that using in-program option one can send the saved graphs and data file from one email address to another email address. The files are automatically attached with the email and the user can add subject line and message body to the email as shown in Figure 7.

(a)
(b)
(c)

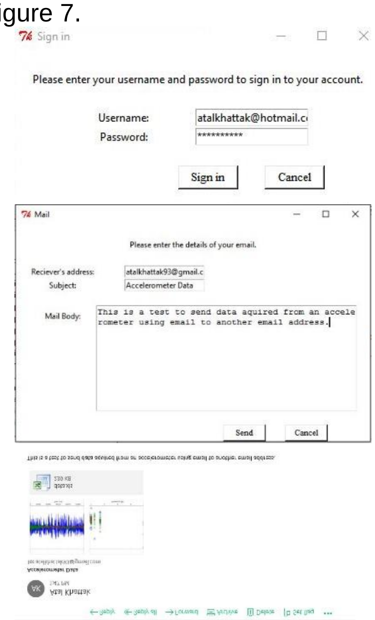


Figure 7. Sending data through email: (a) Sign in message box, (b) Email message attachment message box and (c) Received email

A flow chart is presented in Figure 8 to show the options avaiable for execution for change of motor speed. The program starts and display is generated. The startup menu is displayed that shows three options that is" Play", "Pause", and "Stop". When "Play" is selected the timer starts and the motor starts rotating. Option is available for speed change. When speed is changed then the command is executed only if done is selected otherwise no effect is taking place. The data is transmitted and the startup menu is displayed. When "Pause" is selected from the startup menu then the timer is stoped and the motor is stoped and the data is transmitted. When the third option of "Stop" is selected then aentered variablesare cleared and the timer and motor are stoped. The data is transmitted and the startup menu is displayed.

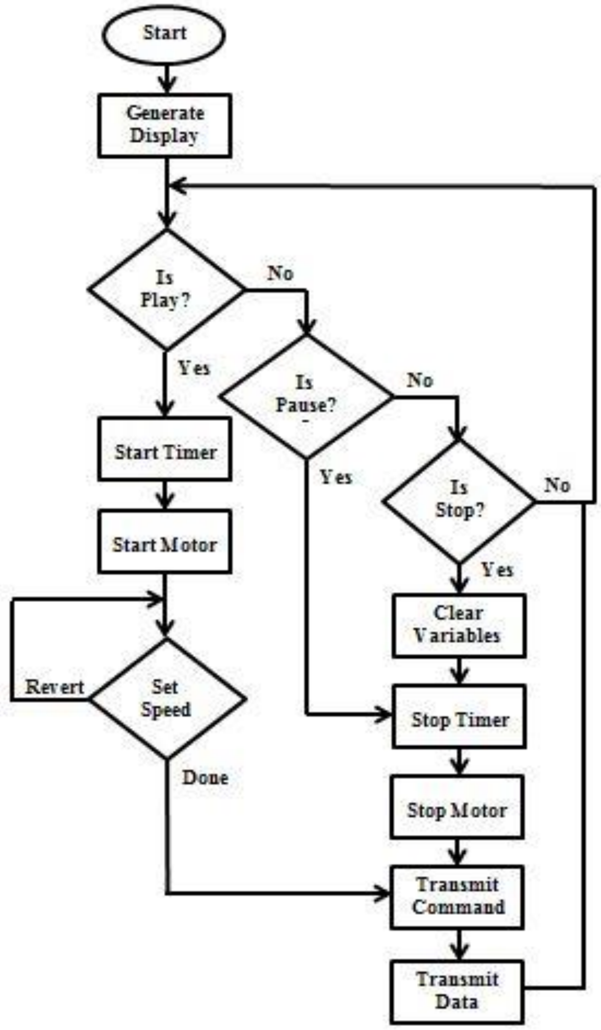


Figure 8. Flow chart for motor speed control

Figure 9 is showing the flow chart for the data acquisition. When the program starts a display is generated. The display shows options of "Play", "Pause" and "Stop". An infinite loop is executed until an option is selected. The selection of "Stop" command clears all variables and the graph is also cleared. The selection of "Pause" command keeps the startup menu available. When the command "Play is selected then command is executed, data is transmitted and graph is updated when the data is received. The user is prompted to save data. If save data option is selected then data is saved and startup menu is displayed. If save data option is not selected then save graph option is displayed. Entering a key saves the graph and the startup menu is displayed again.

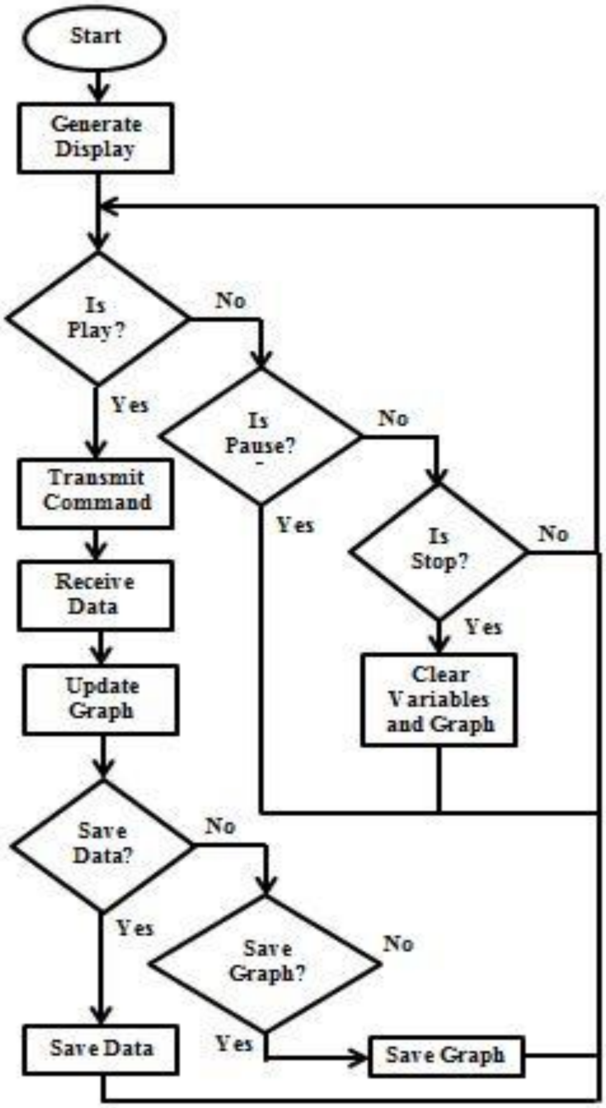


Figure 9. Flow chart for data acquisition

5. Arduino Programming for the developed system

To interface the computer with the DC motors and sensors an Arduino board (ArduinoMega2560) is used. Arduino is a computer hardware and software company and the community provides open-source contents for building and programming microcontroller based boards and kits for a wide range of control applications. At the base of the Arduino project lies the microcontroller board designs that have been manufactured by several vendors, using various microcontrollers. These designs offer a distinct set of digital and analog input/output (I/O) pins that can be interfaced to various expansion boards (shields) and other circuits using connecting wires with the headers connected to the I/O pins. These boards also include serial communication interfaces, including universal serial bus (USB) on some models for loading programs from computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the project, which includes support for the C and C++ programming languages.

The programming of an Arduino is very easy and user friendly. There are two main functions void setup () and void loop (). For initializing any features, the programmer wants to use at the start of the program, the programming is done in the void setup () function. In the void loop () function lies the main body of the code that the microcontroller continuously executes in an unending loop. The board that is chosen for this project is ArduinoMega2560 that contains an Atmega2560 microcontroller of the Atmel AVR series. The operating voltage for the microcontroller is 5 V. It has a 12 V power input pin, a 5 V pin and a 3.3 V pin and additionally, can be powered from a computer using USB connection. The USB connection is used to communicate with the computer and the IDE, also it is utilized for programing the controller or observe data using serial monitor tool. There are 54 digital I/O pins 15 of which are designated as PWM pin, i.e. they can be used as PWN pins without having to specifically initialize the timer, the programming of the timer is done automatically by the IDE. There are a total of 16 Analog I/O pins whose reference voltage is 5 V unless specified using programming in which case it will take the voltage at Vref pin as reference. Moreover, there is a 16 MHz internal clock which ensures fast code line execution. For data and program storage it has 256 kb flash memory, 8 kb SRAM and 4 kb EEPROM.

For the Arduino programming, the flow chart is shown in Figure 10. The Arduino program initializes setting serial port and baud rate. The program then enters an unending loop waiting for commands to be received from the computer. If the command is to send data transmission, it starts a function that reads the data from time variable and each of the analog pins and transmits that to the computer. Also depending upon the command sent from the computer the Arduino sets desired PWM duty ratio (also transmitted from the computer) on the respective I/O pin. To stop the motors duty ratio is set at 0.

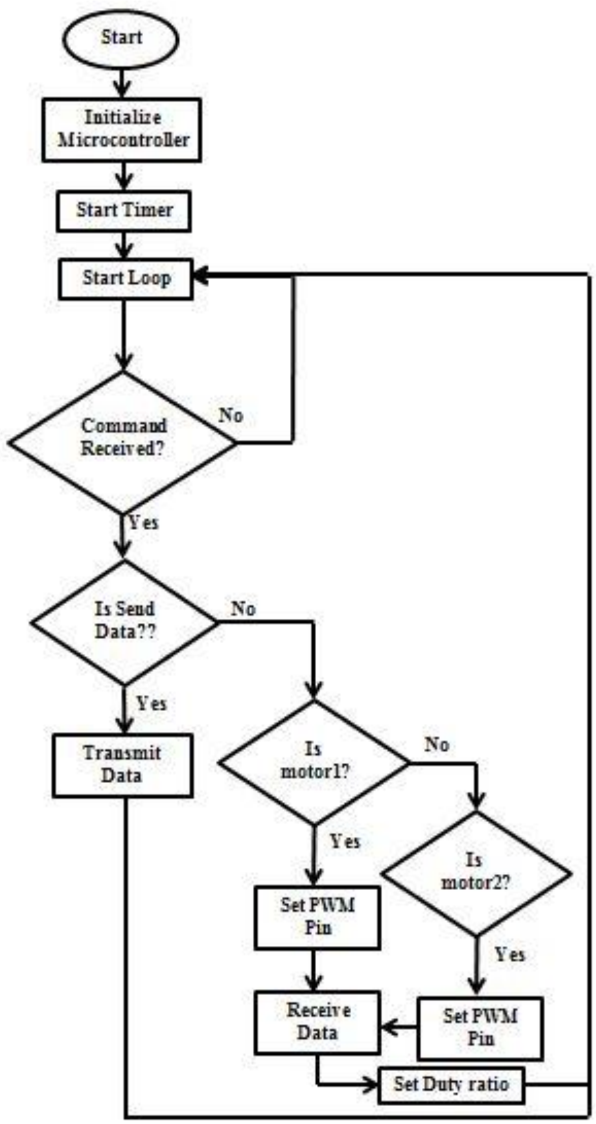


Figure 10. Flow chart for Arduino Program

6. Experimental Setup for the developed system

For testing the developed system, initially the experimentation is performed on a simple setup as shown in Figure 11. The computer is connected to the Arduino using USB connection. The pins defined for motor 1, PWM (pin 8) and for motor 2, PWM (pin 9) are connected to LEDs. By varying the PWM duty ratio the brightness of the LEDs is varied to confirm that the speed control is working. The brightness control of LED and speed control of motor use the same concept of PWM and can be used to verify the code. The V_{in} and GND pins of the ADXL335 sensor were connected to the 3 V pin and GND pin of the Arduino thus regulated power to the sensor is provided from the Arduino without using external power source. The data axes pins of the sensor, x-pin, y-pin and z-pin, are connected to the Arduino analog I/O pins, A3, A2 and A1, respectively. The Arduino board was also powered using the USB connection from the computer. All the functionalities of the GUI were tested and yielded positive results.

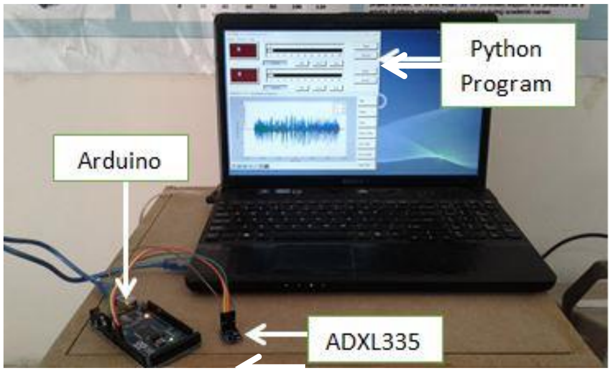
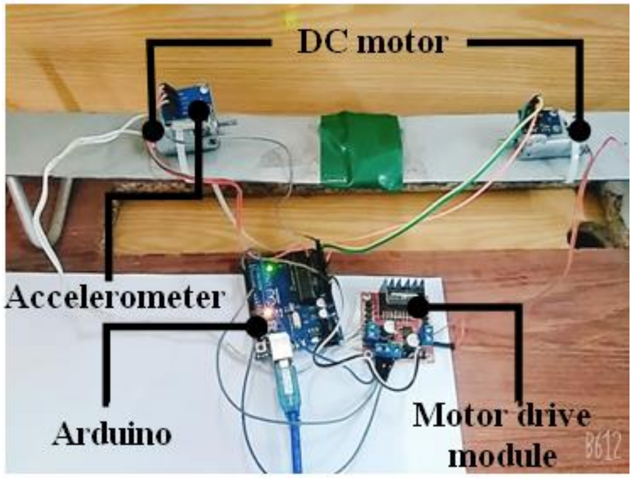


Figure 11. Experimental setup for simple testing the developed system

Finally, the actual motor control and data acquisition test is carried on DC motors as shown in figure 12. Two DC motors are attached to the Arduino through an H-bridge motor drive module (L298N) and two accelerometers (ADXL335) are mounted on these motors for acceleration measurements.By varying the PWM duty cycle ratio the speed of the motors issuccessfullyregulated and simultaneous the acceleration data is acquired. Additionally, all the functionalities of the GUI were tested and yielded positive results.

(a)



(b)

Figure 12. *Testing the developed system for regulating the motor speed and monitoring the vibrations: (a) Experimental setup and (b) close-up view of the setup*

7. Comparison and discussion

The developed system is compared with the similar systems reported in literature as shown in Table 1. From the data presented in the table, it can be seen that there are various advantages provided by using different works. In terms of programming language [this work] has the clear advantage, as by using Python, a portable application can be developed that does not require Python to be installed on the computer to operate, also Python language is easy to learn and use, thus, making modifications to the program is easy for new programmers. In terms of controllers depending upon the familiarity and preference of programmers, all reported works will be advantageous to these programmers. In terms of mode of communication, [11], [12], [13]offer an advantage since they use wireless communication. The works reported in [11] and [16] offer an advantage in terms of control system since they use the more stable closed loop control systems. However, in terms of generalization [this work] provides an advantage since it is not model specific in terms of speed data, speed control, uses speed in terms of percentage of full speed rather than actual speed. Moreover, if talking in term of data backup then [this work] is at an advantage followed by systems reported by [12] and[13]. While [12] and[13] only create a database to save data, [this work] can save the data in graphical and spreadsheet formats and can also backup the data on a server or another computer by using internet to send email from one address to another email address, where the receiver can archive the mail (in the email server) or download it on to the computer.

Table 1. *Comparison of this Project and other Projects in Literature Review*

Programming Language/ Software	Microcontroller	Mode of Communication	Control Type	Sensor for Data Collection	Data Saving	Ref.
LabVIEW Software	Atmega32	Wireless ZigBee protocol	Closed Loop	Optical Encoders(for speed and position)	-	[11]
X-CTU Software	Atmel 8-bit AVR	Wireless ZigBee protocol	Open Loop	Temperature Sensor, Shaft Encoder, Voltage Sensor, Current Sensor	Visual Basic Database	[12]
Visual basic Language	AVR microcontroller	Wireless ZigBee protocol	Open loop	Temperature Sensor, Shaft Encoder, Voltage Sensor, Current Sensor	Visual Basic Database	[13]
MATLAB	PIC microcontroller	Serial Wired Communication	Open Loop	-	-	[14]
C# Language	Arduino Board	serial Wired Communication	Open Loop	IR Pair	-	[15]
LabVIEW Software/Visual Basic Language	PIC18F452	32-bit Parallel Wired Communication	Closed Loop	Voltage Sensor	-	[16]
Visual Basic6.0 language	PIC18F452	Serial Wired Communication	Open Loop	-	-	[17]
Python Language	Arduino Mega2560	Serial Wired Communication (primary), Email (secondary)	Open Loop	ADXL335	Graph Pictures (.jpeg), Data file (.xls)	This Work

8. Conclusions

This work developed a motor control and data acquisition system that monitors sensor's data values and can modify motor speed using a computer based graphical user interface (GUI). Work has been done previously on such systems, they used different controllers, modes of communication and programming languages (for developing GUI) to monitor different working parameters of motor operation and control motors based on data acquired from these sensors. These parameters include motor speed, temperature and position. This work monitors motor health and safety by analyzing data on vibrations caused by the motor and by other sources inthe environment of the motor. Accelerometer measures the amplitude and frequency of acceleration. Microcontroller reads data and transmits it to the computer using USB connection. Once on the computer data is graphically represented then graphs and data can be saved on the computer or forwarded toanother computer using Email. Moreover, this work can be used in future tasks to monitor other parameters for motor. Wireless communication between controller and computer or controller and sensor can be used. Data can be acquired and presented from multiple sensor nodes as well. The program can be made portable and Arduino based backup or data can also be developed

References

[1] Khan FU. A vibration based electromagnetic and piezoelectric hybrid energy harvester. Int J Energy Res. 2020;44: 6894–6916. https://doi.org/10.1002/er.5442

[2] Khan F, Sassani F, Stoeber B. Nonlinear behaviour of membrane type electromagnetic energy harvester under harmonic and random vibrations. Microsyst. Technol. 2014;20:1323-1335. http://doi.org/10.1007/s00542-013-1938-1.

[3] Farid Khan, Farrokh Sassani and Boris Stoeber Copper foil-type vibration-based electromagnetic energy harvester 2010 J. Micromech. Microeng. 20 125006. https://doi.org/10.1088/0960-1317/20/12/125006

[4] Khan, F.U.; Ahmad, S. Flow type electromagnetic based energy harvester for pipeline health monitoring system. Energy Convers. Manag. 2019, 200, 112089. https://doi.org/10.1016/j.enconman.2019.112089

[5] Khan FU. Review of non-resonant vibration based energy harvesters for wireless sensor nodes. J

Renewable Sustainable Energy. 2016;8:044702. http://doi.org/10.1063/1.4961370.
[6] Khan F U, Qadir M U. State-of-the-art in vibration-based electrostatic energy harvesting. J Micromech Microeng. 2016;26:103001.
[7] Ahmad MM, Khan FU. Review of vibration-based electromagnetic–piezoelectric hybrid energy harvesters. Int J Energy Res. 2020;1–40. https://doi.org/10.1002/er.6253
[8] Khan FU, Khattak MU. Contributed Review: Recent developments in acoustic energy harvesting for autonomous wireless sensor nodes applications. Rev. Sci. Instrum. 2016;87:021501. http://doi.org/10.1063/1.4942102.
[9] Ahmad MM, Khan NM, Khan FU. Review of frequency up-conversion vibration energy harvesters using impact and plucking mechanism. Int J Energy Res. 2021;1–37. https://doi.org/10.1002/er.6832
[10] N. Khan, F. Rafiq, F. Abedin and F. U. Khan, "IoT Based Health Monitoring System for Electrical Motors," 2019 15th International Conference on Emerging Technologies (ICET), 2019, pp. 1-6, doi: 10.1109/ICET48972.2019.8994398.
[11] U. Talavaru and R. Nagaraj Naik, "Microcontroller based closed loop speed and position control of DC motor," 2014.
[12] V. S. Mohini Reddy, "WSN based Parameter Monitoring and Control System for DC Motor," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. III, 2014.
[13] V. S. Mohini Reddy, "Remote Monitoring and Control System for DC Motor using Zigbee Protocol," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. III, 2014.
[14] M. P. Ozwin Dominic Dsouza, Babu Naik, "BI DIRECTIONAL SPEED CONTROL OF DC MOTOR AND STEPPER MOTOR THROUGH MAT LAB USING PIC MICRO CONTROLLER," International Journal of Research in Engineering and Technology(IJRET), vol. II, 2013.
[15] A. G. M. Prithviraj R. Shetti, "DC MOTOR SPEED CONTROL WITH FEEDBACK MONITOR BASED ON C# APPLICATION," International Journal of Research in Engineering and Technology(IJRET), vol. III, 2014.
[16] A. K. Remya Ravindran, "A DC Motor Speed Controller using LABVIEW and Visual Basic," International Journal of Electronics & Communication Technology(IJECT), vol. III, 2012.
[17] D. U. I. Ademola Abdulkareem, Adewale A. Ajao, Adelakun A. A, "Programming a Computer and a Microcontroller to Control the Speed and Direction of DC Motors," International Journal Of Engineering And Computer Science(IJECS), vol. III, 2014.