Design & implementation in Robot Programming with reference to different platforms & aspects

[1]Nishat A. Khan, [2]Dr. Kalpana Sharma
[1]Research Scholar, [2]Assistant Professor
[1,2] Computer Science
[1,2] Bhagwant University, Ajmer, Rajasthan, India.

Abstract: Programming robot in such a way so that it can interact with humans more easily and they become a part of human life. By developing its AI (Artificial intelligence) so it can interact with us and gives desired response. It can perform given task, which may be easy or complicated. Human-robot interaction is an emerging area of study that is developing an understanding of how to Program for a robot that is useful and effective in helping people performs tasks in particular domains. Robotics Programming help people as capable partners rather than as tools are believed to be of greatest use for applications in entertainment, education, and healthcare because of their potential to be perceived as trusting, helpful, reliable, and engaging. This Proposed Research Plan explores the details about Research had done in this field. Its merits and importance and need in future world. And also covers work and Development we are going to perform in robotic programming. This will contribute towards the goal of building robot that can effectively communicate with and assist humans in a variety of applications in domains that I believe will benefit from robotic programming. This proposed research plan will be specifically based on the different programming languages and software's used for the robot programming. It also includes analyzing how a program interacts with hardware.

Index Terms **- Development, Hardware, Reliable, Humans, Tasks.**

I. INTRODUCTION

Robots of this type can perform difficult and repetitive tasks for humans. This can be a very risky job and using such robots, such dangerous tasks can be done. But it is useful to check the places where there are dangerous poison gases. Robots can also be used to search for people in buildings that have been destroyed by earthquakes. Due to the establishment of a wireless camera in a detective robot, it can be used to remotely enter the dangerous place and to exit, which humans can not. When the user controls the remote controller, then the Spy Robot will move the spying images around the desired destination and robot. User can check and recommend from computer with wireless remote controller. Lightning LED spy is installed on the robot and it has a stand which is a place for rescue equipment. The robot is not huge enough and is designed to be easy to transport. For the entire system, the required power supply is done by the lead acid battery which is connected to the voltage regulator.

Existing robot programming systems have focused primarily on the specification of sequences of robot configurations. This is only a small aspect of robot programming, however. The central problem of robot programming is that of specifying robot operations so that they can operate reliably in the presence of uncertainty and error. This has long been recognized in research labs, but until very recently has found little acceptance in industrial situations. Some key reasons for this difference in viewpoint are:

1) the lack of reliable and affordable sensors, especially those already integrated into the control and programming systems of a robot;

2) existing techniques for sensory processing have tended to existing robot systems fall short of meeting the requirements we can identify today.

The crucial problem in the development of robot programming languages is our lack of understanding of the basic issues in robot programming. The question of what basic set of operations a robot system should support remains unanswered. Initially, the only operation available was joint motion. More recently, Cartesian motion, sensing, and, especially, compliance have been recognized as important capabilities forobot systems. In future systems, a whole range of additional operations and capabilities are to be expected:

1) ***Increasing integration of sensing and motion:*** More efficient and complete implementations of compliant motions are a key priority.

2) Complete object models as a source o f d a t a f o r senso interfaces and trajectory planning: Existing partial models of objects are inadequate for most sensing tasks; they are also limited as a source of path constraints. Surface and volume models, together with appropriate computational tools, should also open the way for more natural and concise robot programs.

3) ***Versatile trajectow specifications:*** Current systems over specify trajectories and ignore dynamic constraints on motion. Furthermore, they severely restrict the vocabulary of path shapes available to users. **A** mechanism such as functionally defined motion can make it easy to increase the repertoire of trajectories available to the user.

4) ***Coordination of multiple parallel tasks:*** Current robot systems have almost completely ignored this problem, but increasing use of robots with more than six degrees of freedom, grippers with twelve or more degrees of freedom, multiple special-purpose robots with two or three degrees of freedom, and multiple sensors will make the need for coordination mechanisms severe.

5) ***The** IIO, **control, and synchronization capabilities of general-purpose computer programming Languages:*** **A** key problem in the development of robot languages has been the reluctance, on the part of users and researchers alike, to accept that a robot programming language must be a sophisticated computer language. The evidence seems to point to the conclusion that a robot language should be a ***superset*** of an established computer programming language, not a subset. The developments should be matched with continuing efforts at raising the level of robot programming towards the task level. By automating many of the routine programming functions, we can simplify the programming process and thereby expand the range of applications availablet o robots systems. One problem that has plagued robot programming research has been the significant "barriers to entry" to

experimental research in robot programming. Because robot control systems on available robots are designed to be stand alone, every research group has to reemployment a robot control system from the ground up. This is a difficult and expensive operation. It is to be hoped that commercial robots of the future will be designed with a view towards interfacing to other computers, rather than as stand-alone systems. This should greatly stimulate development of the sophisticated robot programming systems that we will surely need in the future.

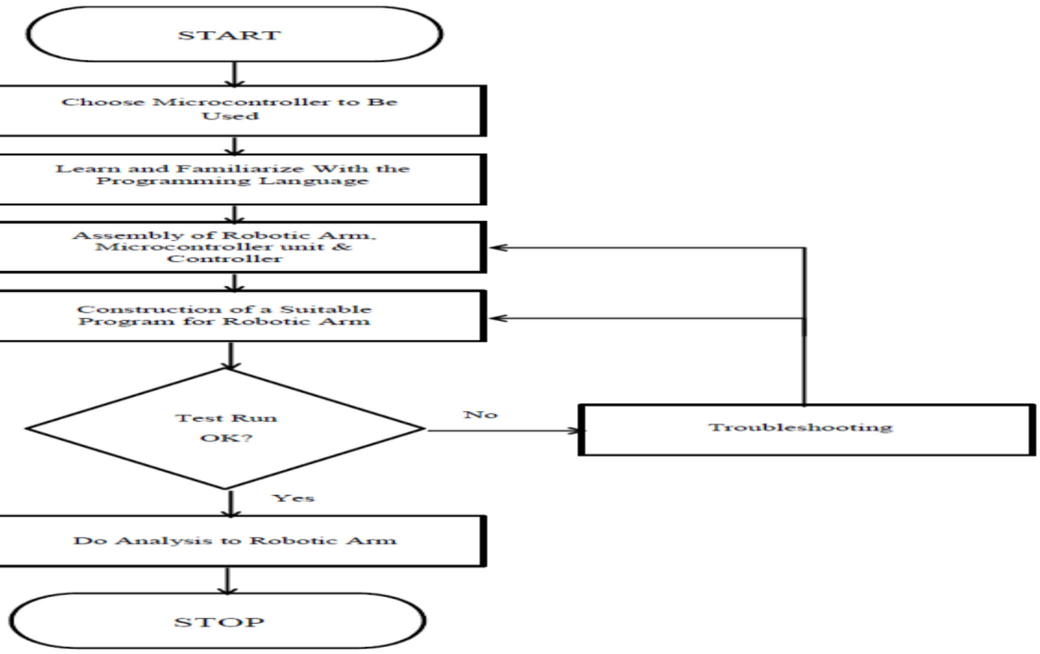
II. Objective

The main research objectives of this study are:

1. To Program a robot in such a way, so that a robot can behave in friendly way with human being.
2. To Programming with different type of microcontrollers.
3. To compare different type of languages used for robot programming.
4. To implement coding with hardware using available devices.
5. To reveal the importance of Robotic Applications and Coding Education for Achieving 21st-Century Skills in India

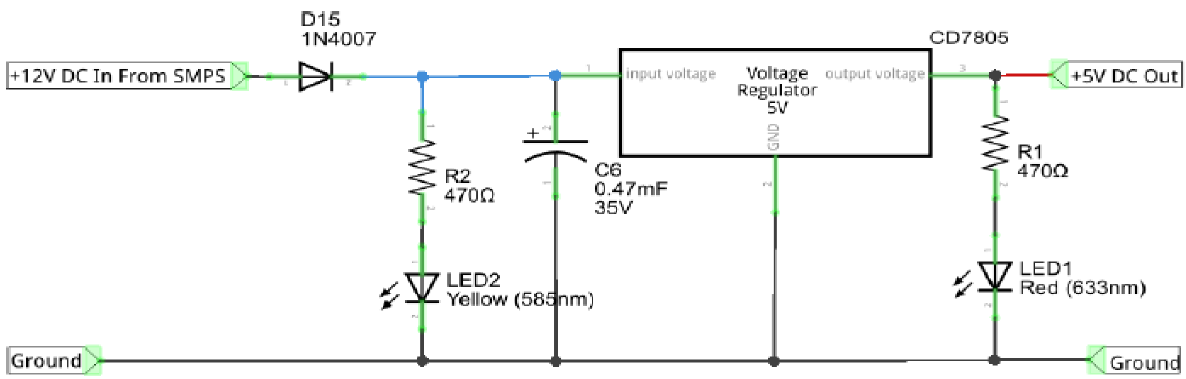
III. Research Methodology

The paper can be separated into two main parts. At first design and implementation of the hardware and circuitries and the second would be development of the Arduino code for the pick & place operation.



IV. Hardware Implementation & Programming

Creation and Design of Power Module Circuitry



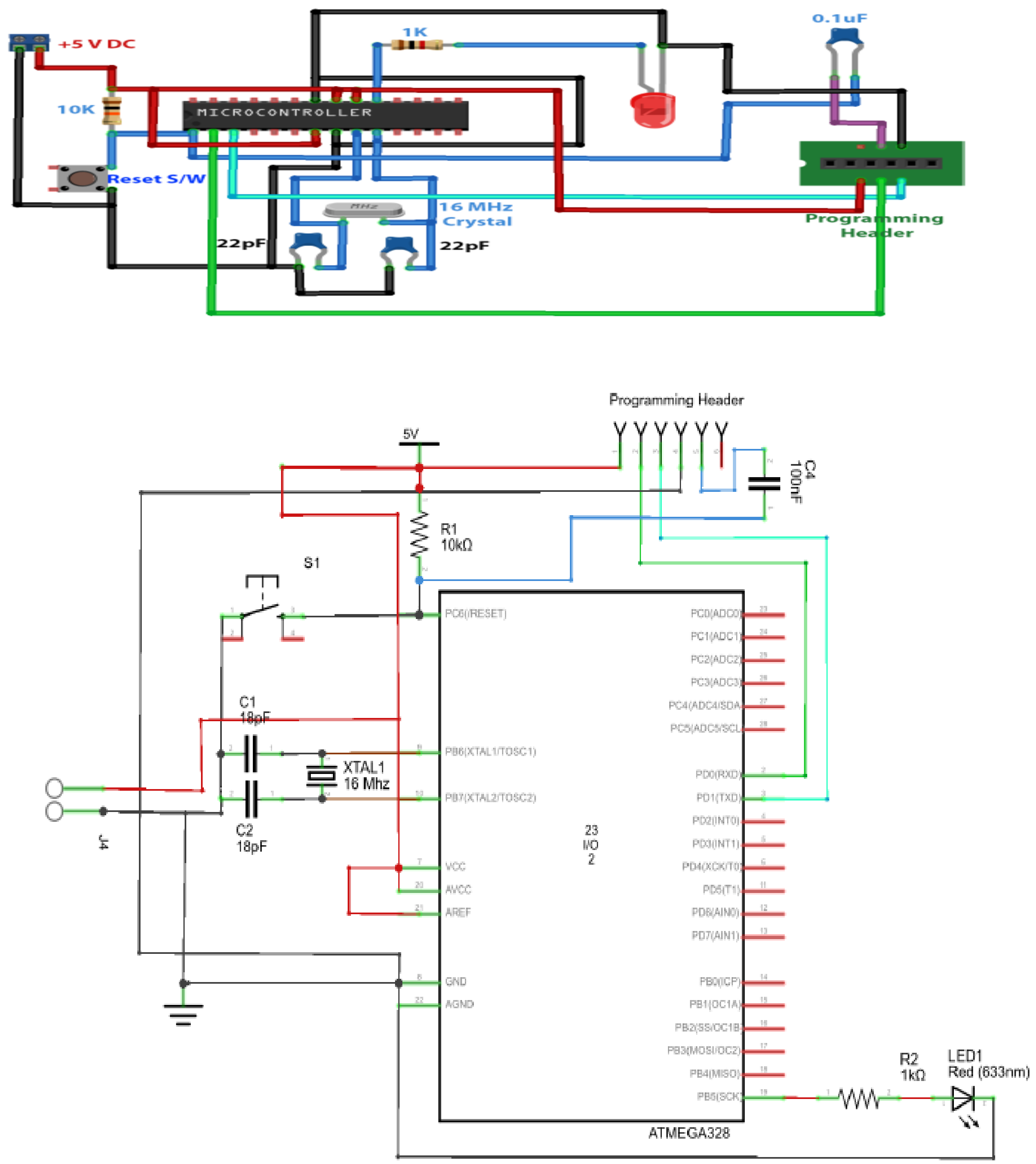
Creation and Design of Microcontroller Board

Basic Parts

1. 1 Veroboard
2. 1 Pushbutton
3. 1 10k Resistor
4. 1 LED
5. 2 22pF Capacitors

6. 1 220 Ohm Resistor
7. 1 16MHz Clock Crystal

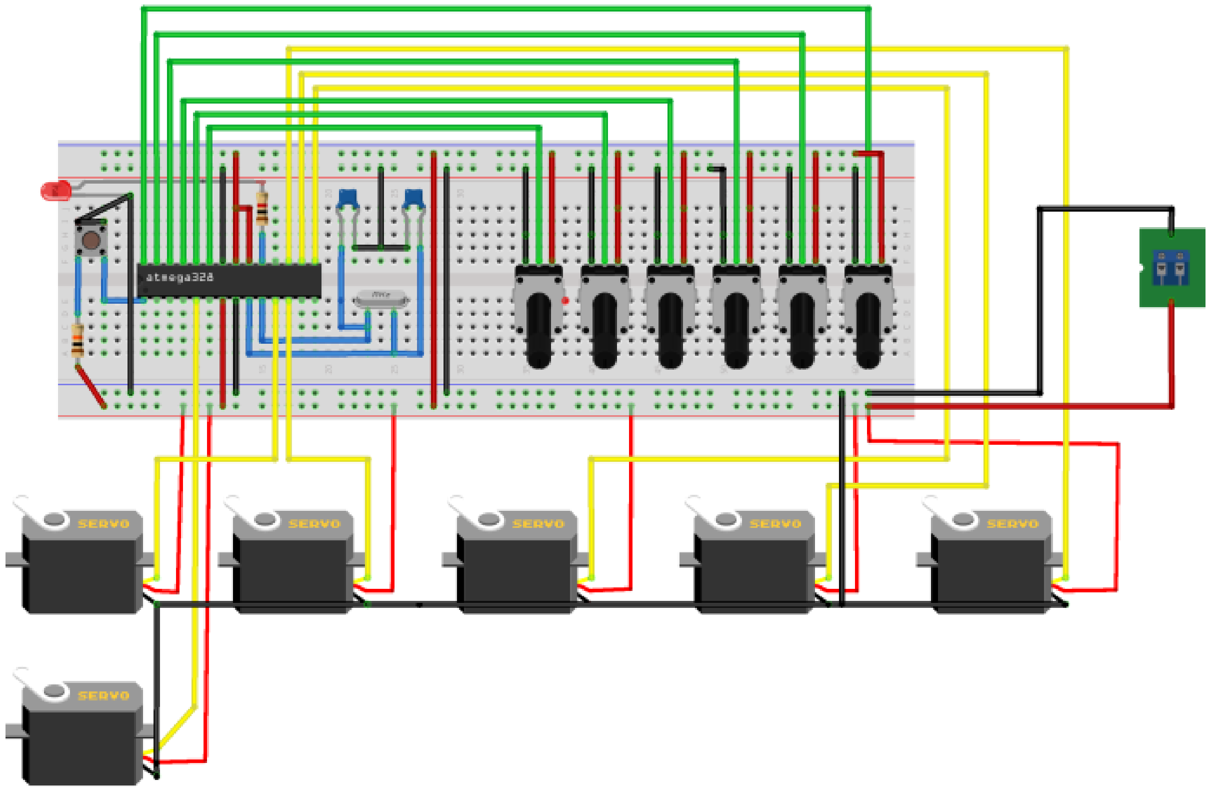
Developed Board



Breadboard View of Microcontroller Board and Schematic View of Microcontroller Board

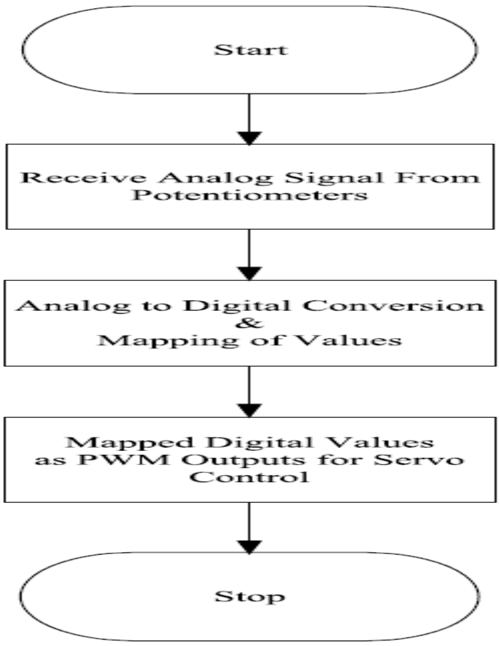
V. PROTOTYPE MANUAL CONTROLLER (DEVELOPMENT)

Manual controllers are developed using prototypes instead of servos with six potentiometers. In this case, the prototype arm is moved manually. The potentiometer will feed analog values (positions) to the microcontroller. Then the microcontroller will receive analog data and process it and the conversion of data will be done with the help of ADC inside the microcontroller.
The microcontroller is programmed with the help of the Arduino IDE so that it can map analog input to its digital output. So that the analog value (voltage value) can be mapped to the digital value from the potentiometer. For specific voltage values we will get a specific angle to set the servo in the robotic arm.



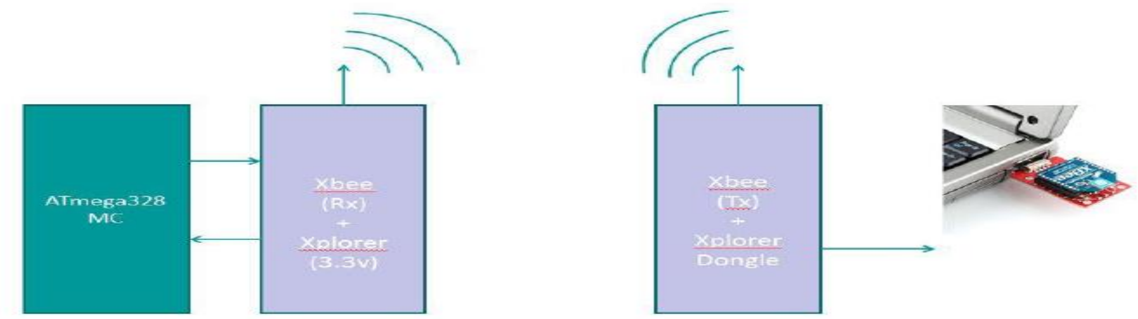
Cardboard view of manual control Interfaced with robotic Arm

VI. CONTROL OF THE ROBOT ARM BY REFLECTING THE MOVEMENTS OF THE MANUAL CONTROLLER PROTOTYPE

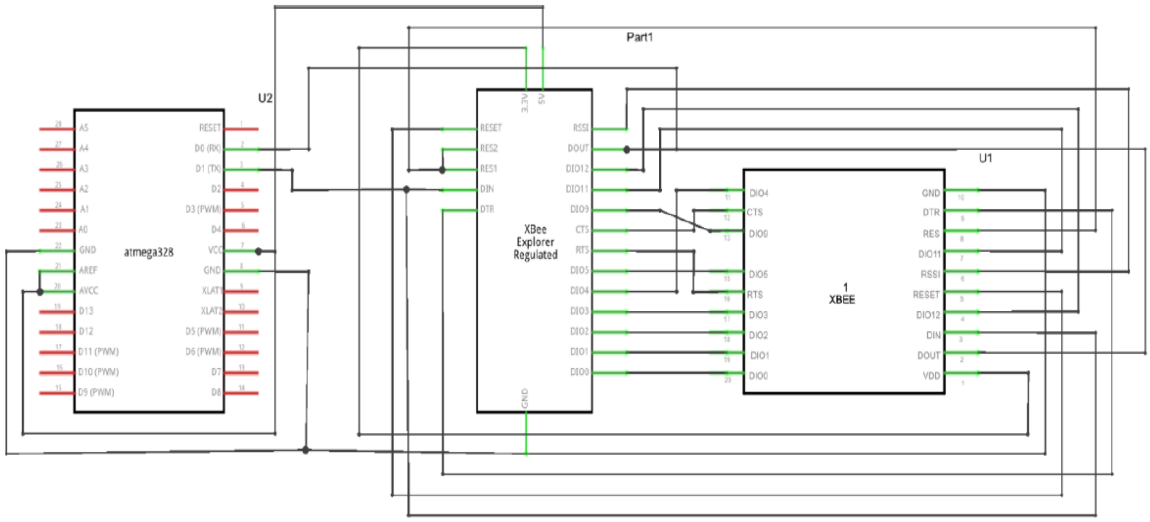


Flow chart of Control of the robot arm by reflecting the movements of the manual controller prototype

Wireless control of the robotic arm using a GUI

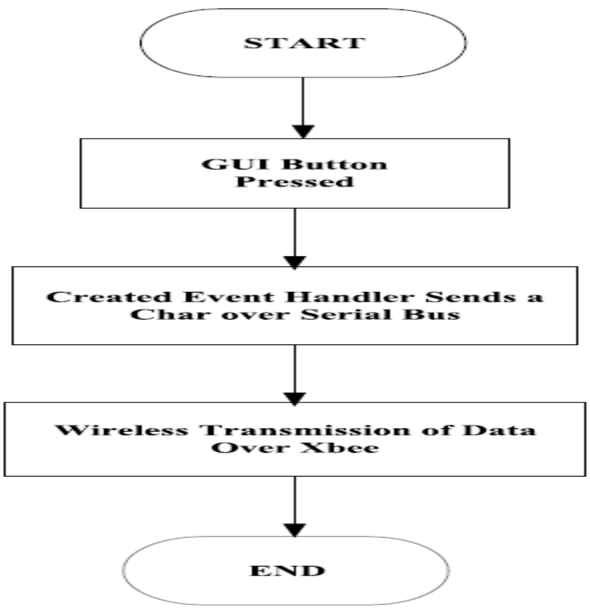


The microcontroller board will interface with an xbee module using an xbee xplorer regulator. This will be communicating with the PC with the help of another xbee module which is connected to the PC using an xbee xplorer dongle. Here the PC will act as a transmitter and the interfering xbee with the microcontroller will act as a receiver that controls the servo of the robot arm.

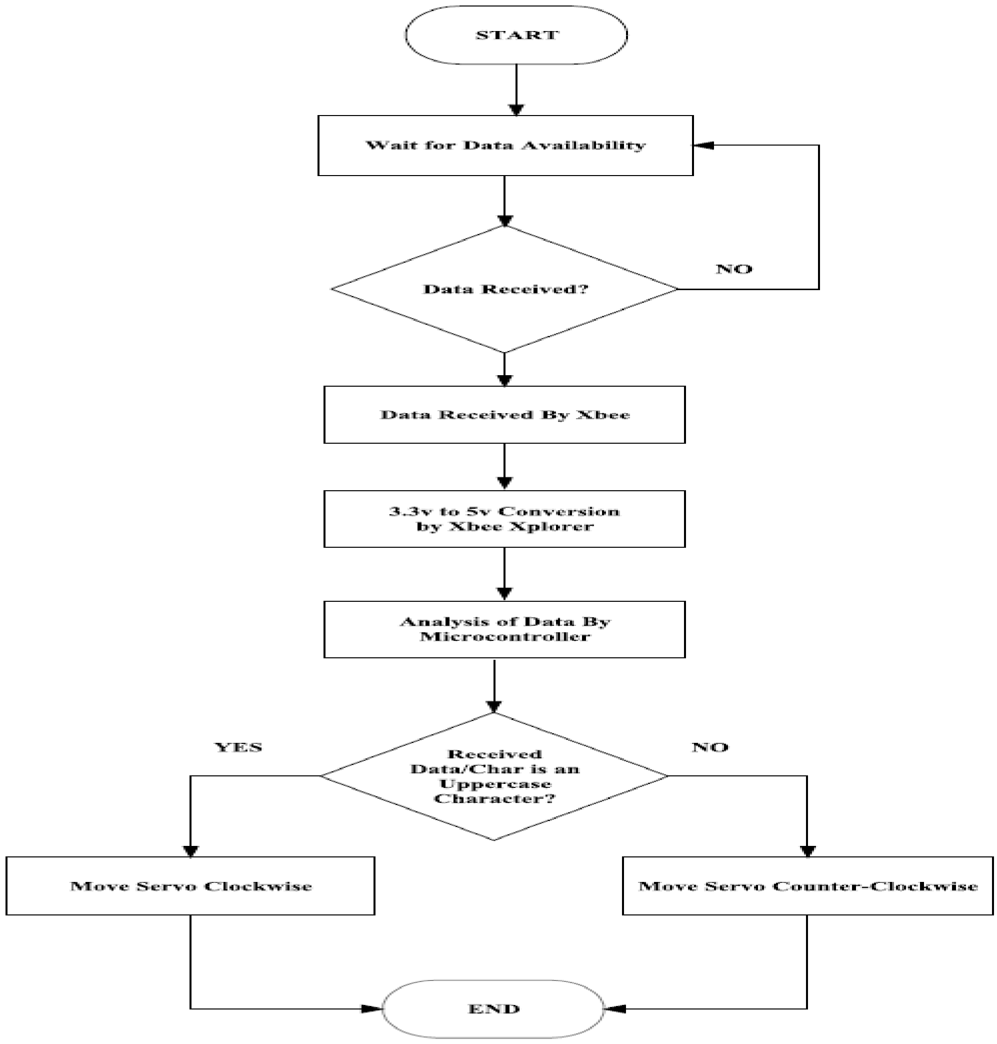


Reciving End Interfacing

After power up, the explorer dongle board will make a virtual COM port in the PC on the transmitting end, and you can communicate with the module with serial port devices. The Explorer board is built from FTDI with FT232RL chip and drivers are available for Windows, Linux and Mac.



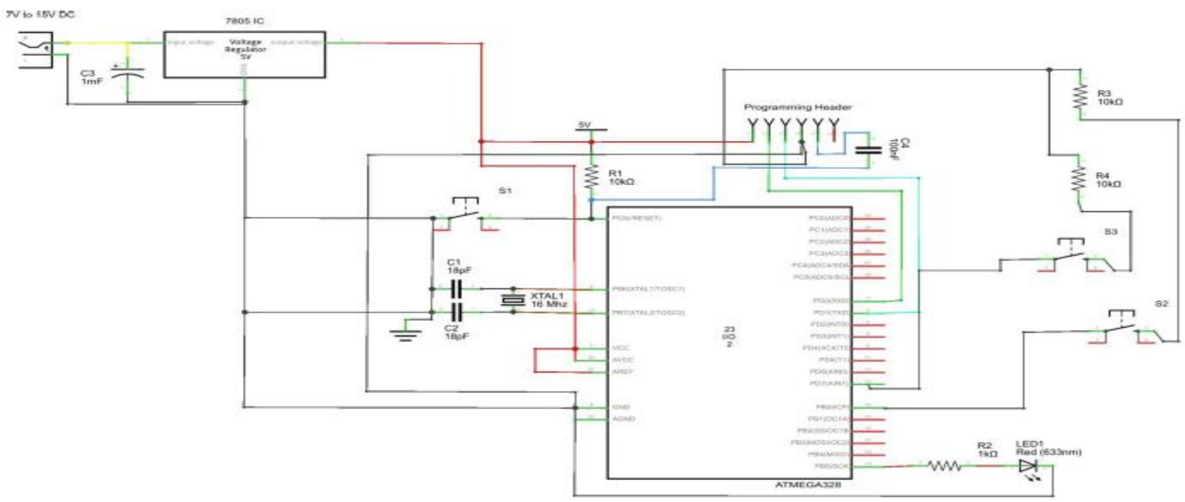
Flowchart for Data Transmission over Xbee



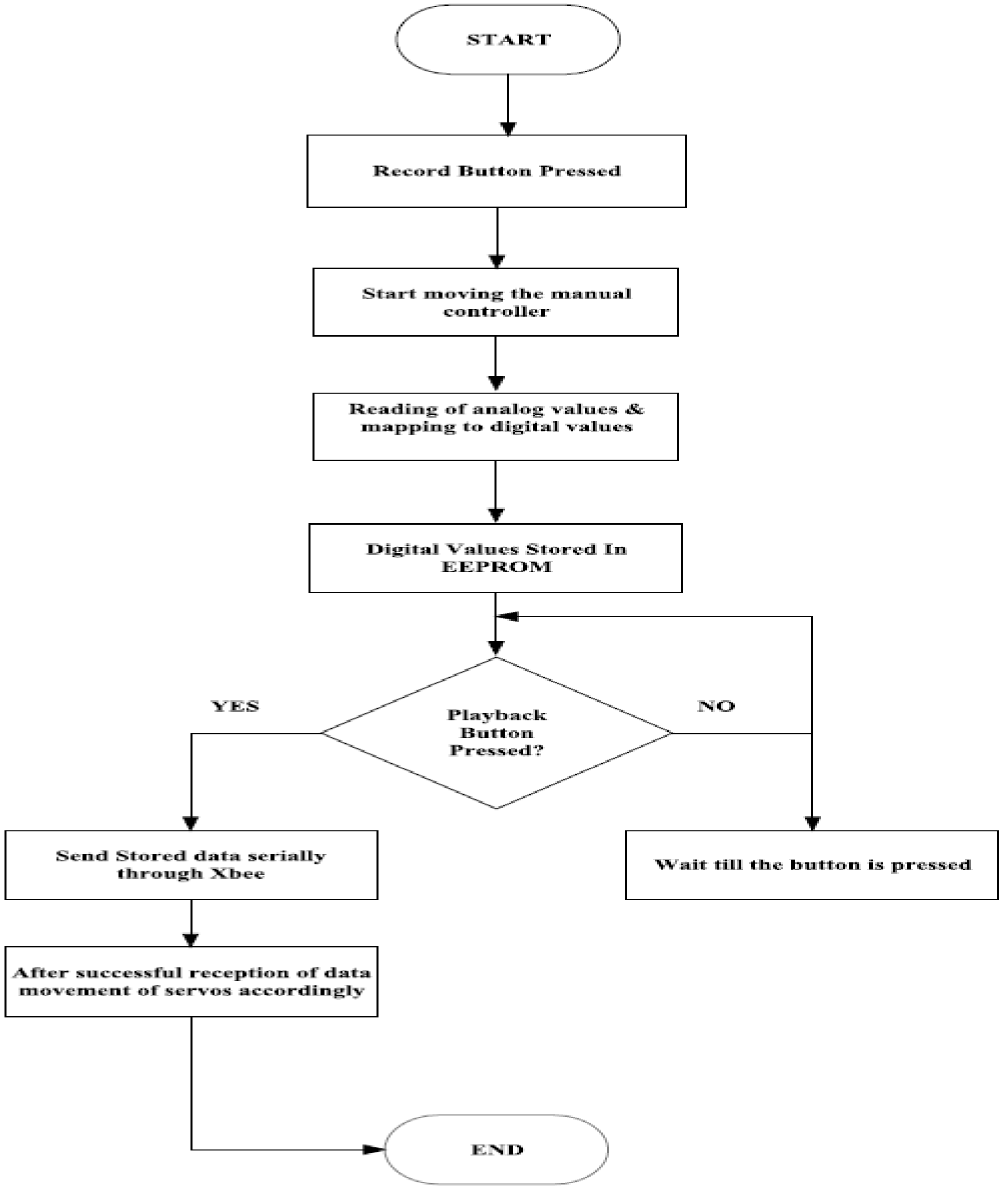
Flowchart for Data Reception over Xbee

Here both the transmitting and receiving ends are microcontroller units interfaced with xbee modules. The transmitting end have 2 extra push buttons attached to microcontroller pins 7and 8. The push button connected to pin 7 will act as the recording button n the other one as the playback button.

The transmitting end will receive the analog inputs from the potentiometers of the manual controller. The main code polls the state of the record and replay buttons. When the record button is pressed, the on board LED lights up to let you know it's time to move the manual controller in the positions you want. The analog feedback values are read then converted into servo degrees and finally written to the EEPROM. When the replay button is pressed the stored servo positions are read from EEPROM and sent over the serial bus.the positions for 6 servos are being sent serially.so the first 6 bits of data will contain information about the six servos serially. After reading this first six bits of data again the next six bits will be analyzed and will be fed to the servos.



Transmitting End Schematics of Trainable Robotic Arm

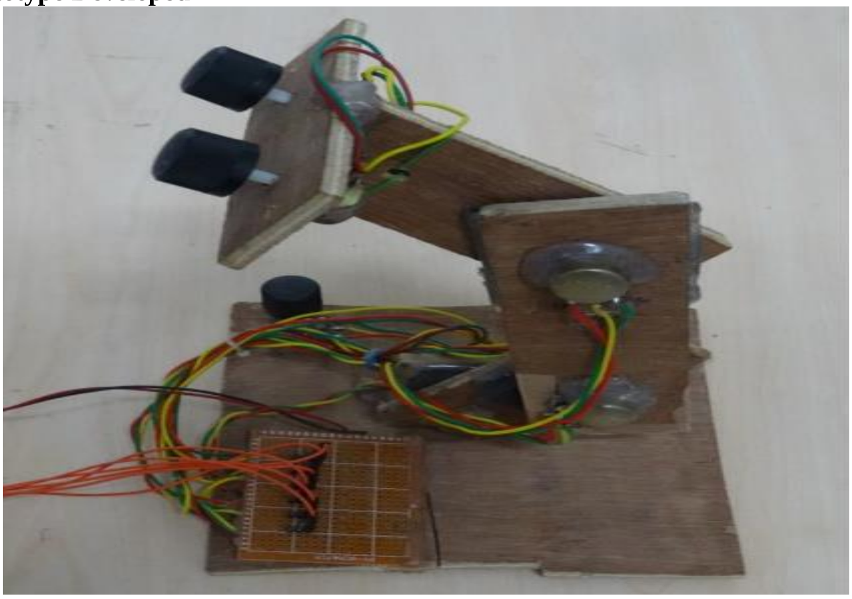


Flowchart for Trainable Robotic Arm

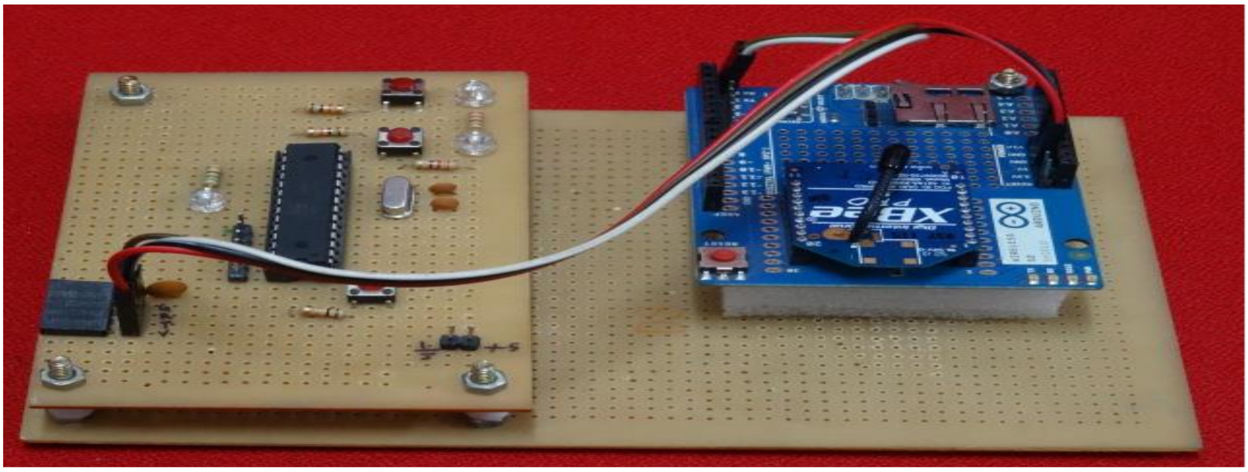
VII. RESULTS AND DISCUSSIONS

The overall results for the thesis will be presented and explained. The results include the actual structure, circuitry and programming source code.

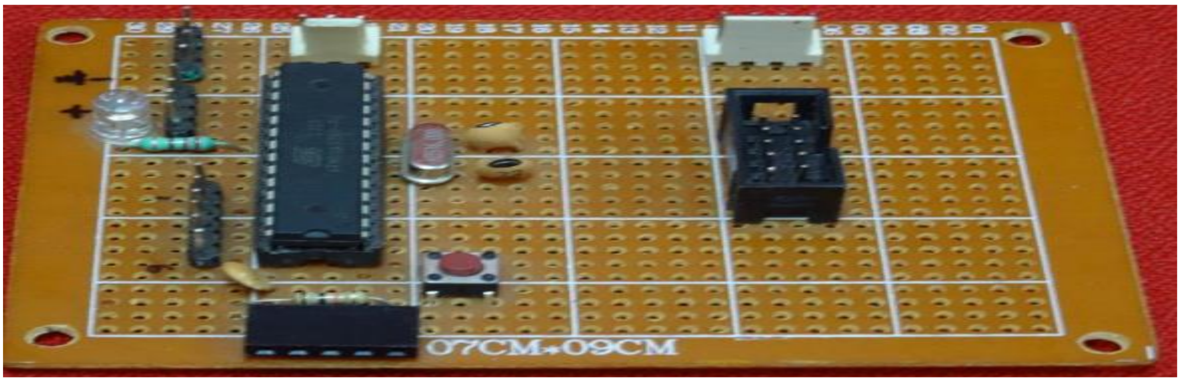
Manual Controller Prototype Developed



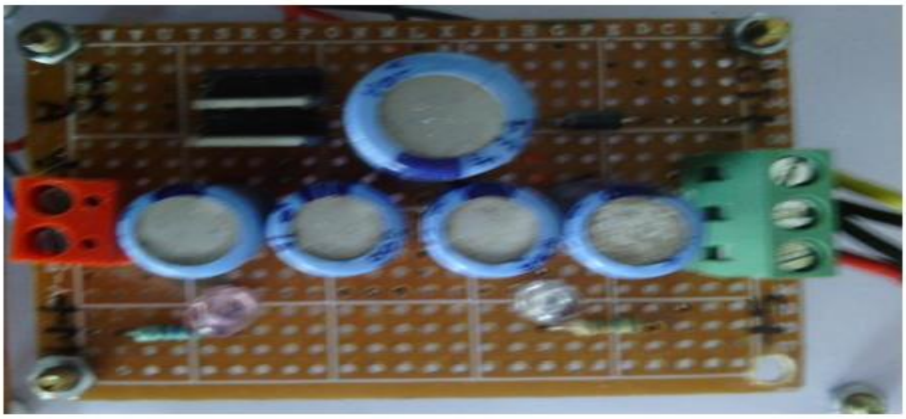
Circuitry Developed



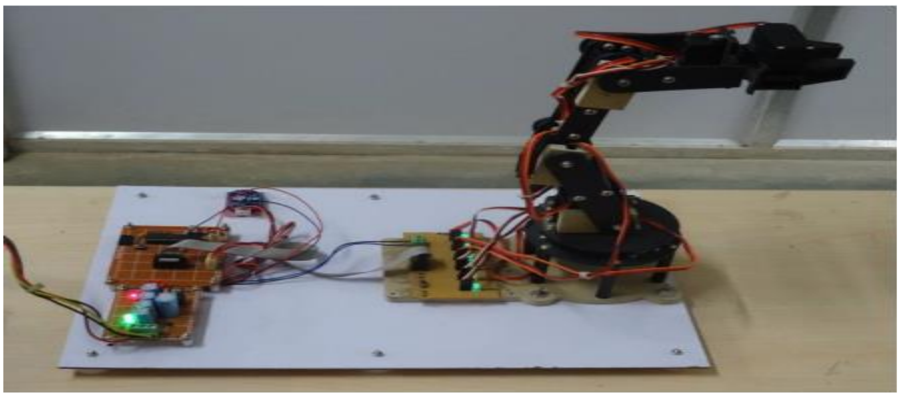
Microcontroller Boards developed



Power Module Developed



Actual Robotic Arm Structure



In the end, of the thesis are accomplished. A successful development of a 4-degree freedom automatic robotic arm with a 2-finger grip is quicker to lift and hold. Skills to solve the problem were put to use to overcome any hardware, software, or circuitry problems. Thesis offers analytical skill training, hardware assembly training, program writing training and circular design training.

Generally, this thesis gives students the opportunity to incorporate robot principles and applications into their thesis. Future developments on thesis related to robotic studies are necessary to advance the robotic field in our country and to bring the robotic field to a new level internationally.

REFERENCES

[1]. Fritzing Software, http://fritzing.org/home/
[2]. Svec, Robert. "Trainable Robotic Arm." Overview. Adafruit, n.d. Web.
[3]. Arduino References, Online Contents,
[4]. http://www.arduino.cc/en/Main/Standalone
[5]. http://arduino.cc/en/guide/Environment,
[6]. http://arduino.cc/en/Main/ArduinoBoardDiecimila
[7]. Processing References, Online Contents,
[8]. https://processing.org/
[9]. https://processing.org/reference/libraries/
[10]. G4P tools : http://www.lagers.org.uk/g4p/index.html
[11]. Atmel, "Atmega328p." Retrieved from http://www.atmel.com. 2014
[12]. Arduino, "Introduction: Arduino Uno Overview." Retrieved from arduino.cc/en/Main/ArduinoBoardUno, 2014
[13]. Quigley, Morgan, Alan Asbeck, and Andrew Ng. "A low-cost compliant 7-DOF robotic manipulator." In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 6051-6058. IEEE, 2011.
[14]. Krishna, R., G. Sowmya Bala, S. Sastry ASC, B. Bhanu Prakash Sarma, and Gokul Sai Alla. "Design and implementation of a robotic arm based on haptic technology." Int. J. of Eng. Research and Applications 2, no. 34 (2012).
[15]. Cheng, Heng-Tze, and Pei Zhang Zheng. "Real-Time Imitative Robotic Arm Control for Home RobotApplications." (2011).
[16]. Zhang, Yong, Brandon K. Chen, Xinyu Liu, and Yu Sun. "Autonomous robotic pick-and-place of microobjects." Robotics, IEEE Transactions on 26, no. 1 (2010): 200-207.
[17]. Liu, Xia, Heping Chen, Mengliu Wang, and Shanshan Chen. "An XBee-Pro based energy monitoring system." In Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian, pp. 1-6. IEEE, 2012.
[18]. B Goldy Katal, Saahil Gupta, Shitij Kakkar. "Design And Operation Of Synchronised Robotic Arm", International Journal of Research in Engineering and Technology, Volume: 02 Issue: 08 | Aug-2013.