

FABRICATION AND DEVELOPMENT OF A SERVO CONTROLLED PICK AND PLACE ROBOTIC ARM

Abstract

This paper addresses the automation of a material handling system responsible for orchestrating the motion of a robotic arm. The objective is to enable the arm to successfully grasp objects that are in transit on a conveyor belt oriented along the vertical axis. This study focuses on constructing a model designed to identify items positioned on a conveyor. Subsequently, the model retrieves and positions these items in their predetermined locations through automation. This approach eliminates repetitive tasks that were previously carried out by humans, while simultaneously enhancing work precision and speed. The primary objective of this project involves the automated manipulation of objects through a robotic arm for picking and placing tasks. The system is additionally equipped with an infrared (IR) obstacle sensor and an audible buzzer alert. The microcontroller transmits signals to the robotic arm's motors, orchestrating their movement to grasp and position objects accurately according to predetermined coordinates. Responding to object detection, the robotic arm navigates to the designated location, releases the object, and subsequently returns to its initial position. The system also incorporates a light-dependent resistor (LDR) sensor to verify the presence of and changes in containers.

An IR obstacle sensor is employed to detect the presence of the object. The optical sensor is a fusion of an infrared sensor and a phototransistor. As the object interrupts the passage of infrared light to the phototransistor, a square wave output signal is generated at the sensor's output stage. This digital signal is then transmitted to the Microcontroller. The fundamental firmware

Author

A. Johnson

Senior Assistant Professor
KITS(S) Singapur, Huzurabad
Karimnagar, Telangana, India.
ajs.kits@gmail.com

for the microcontroller is coded in Embedded C language. The system also integrates a comparator circuit to compare and identify obstacles, activating a buzzer alarm for alerts.

A DC motor propels the movement of the conveyor belt, while a servomotor facilitates the grasping and positioning of objects into a container at a specific angle. The project encompasses components like a microcontroller, an IR obstacle sensor for object detection, servomotors, and DC motors. The hardware's block diagram is depicted below. Alongside this, a DC motor is employed to manage the vertical movement of the conveyor belt and to trigger user notifications upon object detection using the IR obstacle sensor. By utilizing the IR sensor for object detection, the system is capable of discerning object presence on the conveyor belt. The microcontroller is programmed and its code is written in the C programming language.

Keywords: Microcontroller, Phototransistor, LDR sensor.

I. INTRODUCTION

- 1. Robotic Arm:** Robotic arms represent a category of articulated robot manipulators designed to facilitate interactions between robots and their surroundings. Many of these arms incorporate onboard controllers or translators, streamlining communication, although their control can be direct or achieved through diverse methodologies. This characteristic has led to the categorization of independent arms as comprehensive robotic systems. A robotic arm, often programmable, resembles a mechanical arm and serves functions akin to those of a human arm. It can function as a standalone mechanism or serve as a constituent within a more intricate robot setup.

These manipulators consist of interconnected links linked by joints that enable either rotational movement (as seen in articulated robots) or linear displacement (translation). These links collectively form a kinematic chain within the manipulator. The ultimate point in this kinematic chain is termed the end effector, analogous to the human hand, and serves a comparable purpose in the robot's functionality.

- 2. Robotic Hand:** The robotic hand, commonly referred to as the end effector, possesses the flexibility to be tailored for a multitude of tasks, including but not limited to welding, gripping, and spinning, contingent upon the specific application. Consider the case of robot arms within the realm of automotive assembly, where an array of functions such as welding, manipulating parts, and facilitating rotation find implementation during the assembly process. In certain scenarios, there exists a requirement to closely mimic the capabilities of the human hand, exemplified by robots meticulously crafted for tasks like bomb disarmament and safe disposal.



Figure 1: Articulated Robotic Arm

- 3. Types of Robotic Arms:** A diverse array of robotic arms exists, yet their mechanical configurations often fall within six principal categories. Cartesian robots, also known as Gantry robots, incorporate three joints aligned with the standard X-Y-Z Cartesian axes. Cylindrical arms possess varying numbers of joints operating along a cylindrical axis, usually revolving around a single fixed rod. Spherical arms, alternatively known as polar arms, are equipped with joints that enable unrestricted rotation across a spherical spectrum.

SCARA robots boast a pair of parallel rotary joints, empowering unrestricted motion within a defined plane. Such robots are typically deployed for tasks such as pick-and-place operations. Complex assembly tasks find a suitable match in articulated robots, distinguished by their possession of three or more rotary joints. Parallel robots, on the other hand, are outfitted with three simultaneous prismatic or rotary joints, offering the capability to tilt substantial or delicate platforms with finesse.

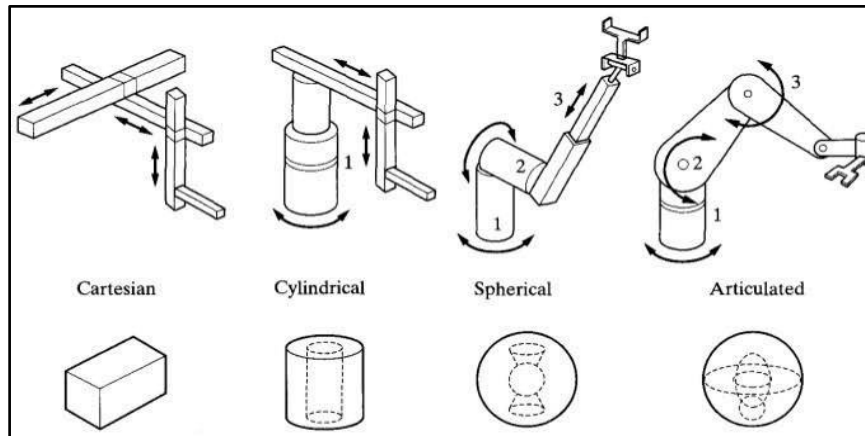


Figure 2: Types of Robotic Arms

II. WORKING PRINCIPLE OF PICK AND PLACE ROBOTIC ARM

- 1. Working of Servo-Controlled Pick and Robotic Arm:** A pick and place robot serves the purpose of lifting an object and accurately positioning it in a specified location. This can manifest in the form of a cylindrical robot, capable of maneuvering along horizontal, vertical, and rotational axes; a spherical robot, offering two rotational and one linear movement; an articulated robot; or even a SCARA robot, characterized by its fixed position with three vertical axes and rotary arms.

The fundamental role of a pick and place robot centers around its joints, which draw a parallel to human joints and facilitate the connection between consecutive rigid segments within the robot's structure. These joints can either be rotary, enabling rotational motion, or linear, providing linear displacement. Incorporating a joint into a robotic link hinge upon comprehending the degrees of freedom and degrees of movement pertinent to that specific component. Degrees of freedom dictate the extent of linear and rotational movement a segment can undertake, while degrees of movement outline the count of axes along which the segment can navigate.

A basic pick and place robot is composed of two rigid bodies positioned atop a mobile base, linked by means of a rotary joint. This rotary joint facilitates rotation along a chosen axis. The operational control of such a straightforward pick and place robot revolves around managing the motion of its end effector. This movement can be achieved through hydraulic motion, involving the utilization of pressurized hydraulic fluid to propel the robot, or through pneumatic motion, wherein compressed air instigates mechanical movement.

However, the most optimal and efficient approach entails the utilization of motors to generate the required motion. These motors necessitate precise control to furnish the necessary motion to both the robot and its end effector.

III. SPECIFICATION OF PICK AND PLACE ROBOTIC ARM HARDWARE REQUIREMENTS

1. Hardware Components

- Microcontroller (16F877A)
- Reset button
- Crystal oscillator
- Regulated power supply (RPS)
- LED indicator
- IR Obstacle sensor
- DC Motors
- DC motor driver
- Servo motor
- LDR sensor
- Buzzer.

2. Software Required: The execution of this project involves the utilization of the subsequent software tools:

Express PCB: Employed for circuit design

PIC C compiler: Utilized for the compilation process

Proteus 7 (Embedded C): Employed for simulating the project's functionality

- **Express PCB:** Breadboards serve as excellent tools for prototyping equipment, offering the flexibility to make design modifications as necessary. However, for the ultimate project outcome, it's preferable to transition to a polished PCB setup with minimal cables, capable of enduring shake tests without complications. A well-structured PCB not only enhances tidiness but also enhances durability by eliminating the risk of cables becoming dislodged.

Express PCB is specialized software designed for crafting PCBs intended for manufacturing by the Express PCB company, as other PCB manufacturers do not support Express PCB file formats. While it boasts user-friendly features, it does come with certain limitations. It's akin to a more basic tool rather than a comprehensive professional CAD program. The part library is limited, but creative workarounds can be implemented. However, it lacks the ability to import or export files in diverse formats and is unsuitable for preparing boards for DIY production.

Express PCB has been effectively used in designing numerous PCBs, including layered ones featuring surface-mount components. PCB patterns are printed out and the toner transfer method, coupled with an Etch Resistant Pen, is employed to create the boards. Nevertheless, it's worth noting that Express PCB lacks an optimal

print layout. Below is the procedure for designing with Express PCB and refining the patterns to ensure a clean and presentable print outcome

- **Preparing Express PCB for First Use:** Express PCB offers a somewhat modest array of components, which may not be overly captivating. Therefore, before embarking on any project, it's advisable to navigate to Audio logical and acquire additional components such as morsel, ppl, and tangent. Extract these components into your Express PCB directory. Following this, launch the program and proceed to configure your workspace according to your preferences.

Navigate to View Options. Within this menu, configure the units to "mm" or "in," aligning with your preferred measurement system. Additionally, check the box for "see through the top copper layer" located at the bottom. While the standard color scheme features red and green, many opt for the more aesthetically pleasing combination of red and blue.

- **PIC Compiler:** The PIC compiler serves as a software tool used for crafting and compiling machine language code. Following compilation, the machine source code undergoes transformation into hex code, a format essential for programming the microcontroller for subsequent processing. Notably, the PIC compiler is also compatible with C language code.

Proficiency in C language, particularly in its embedded form known as Embedded C, is essential for effectively working with microcontrollers. Given the utilization of the PIC Compiler, it's often referred to as PIC C. The suite comprises distinct compilers: PCB for 12-bit opcodes, PCM for 14-bit opcodes, and PCH for 16-bit opcode PIC microcontrollers. This reference manual encompasses all three compilers, owing to their shared characteristics. Specific microcontroller-related features and constraints are duly highlighted.

Tailored to meet the distinct requirements of PIC microcontrollers, these compilers streamline the development of application software using a more comprehensible, high-level language. Nonetheless, when compared to conventional C compilers, the trio of PCB, PCM, and PCH do exhibit certain limitations. For instance, function recursion is prohibited due to these constraints.

The absence of a stack for variable storage within the PIC architecture and the compilers' optimization strategies underlie the prohibition of function recursion. Despite these limitations, the compilers adeptly handle standard C constructs, input/output operations, and bitwise operations. They encompass support for all conventional C data types, including pointers to constant arrays, fixed-point decimals, and arrays of bits.

The structure of a PIC C program largely resembles that of a regular C program. Familiarity with assembly language can facilitate the creation of C programs. In the PIC context, the central point is the main function, where application-specific tasks are defined. Embedded C, in contrast to other environments, lacks an operating system. Consequently, ensuring the perpetual execution of your

program or main file is essential. This can be achieved through straightforward constructs such as a while (1) or for (;;) loop, designed for infinite execution.

It's imperative to incorporate a header file tailored to the specific microcontroller in use. This step is essential for accessing registers linked to peripherals. For instance:

```
#include <16F877A.h> // Header file for PIC 16F877A
```

- **Proteus:** Proteus software exclusively accommodates hex files. Once machine code undergoes transformation into hex format, it must be transferred to the microcontroller. This process is facilitated by Proteus. Acting as a programmer, Proteus houses a microcontroller distinct from the one intended for programming. This embedded microcontroller contains a program that efficiently accepts the hex file generated by the PIC compiler. Subsequently, it transfers this hex file into the targeted microcontroller for programming.

The Proteus programmer operates through a designated power supply. This power supply is derived from a circuit integrated within Proteus, which in turn is connected to the microcontroller. The program slated for microcontroller integration is edited within the Proteus environment. It is then compiled and executed to identify and rectify potential errors. Following a successful compilation, the program is subsequently loaded into the microcontroller via a dumper mechanism.

3. Procedural Steps for Compilation, Simulation and Dumping:

In the context of the PIC microcontroller, the PIC C compiler is employed for the compilation process. The compilation procedure encompasses the following steps:

- Launch the PIC C compiler.
- Upon initiation, you'll be prompted to designate a name for the new project. To maintain organization, create a dedicated folder to house all project files. Select a suitable name for the project, then save your selection to proceed.

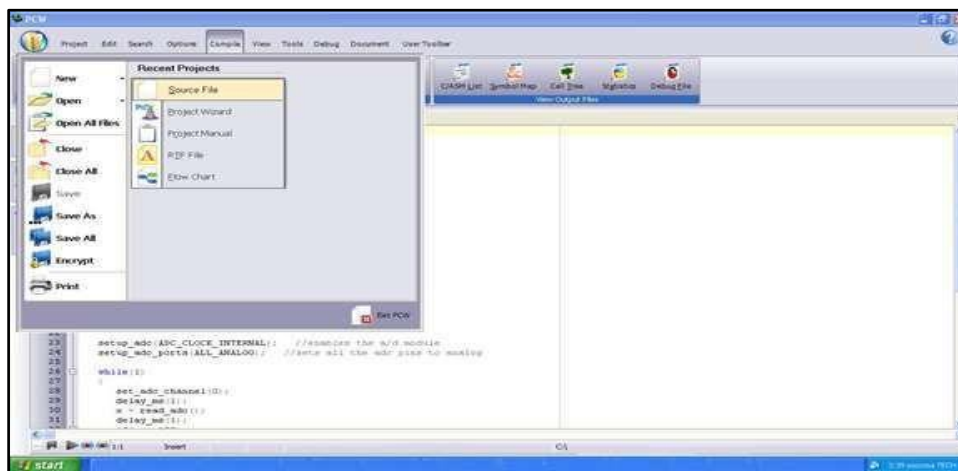


Figure 3: New File using the PIC C compiler

Click Project, New, and something in the box named 'Text1' is where your code should be written later.

Now you must click 'File, save as' and choose a file name for your source code ending with the letter '.c'. You can name as 'projects' for example and click save. Then you have to add this file to your project work.

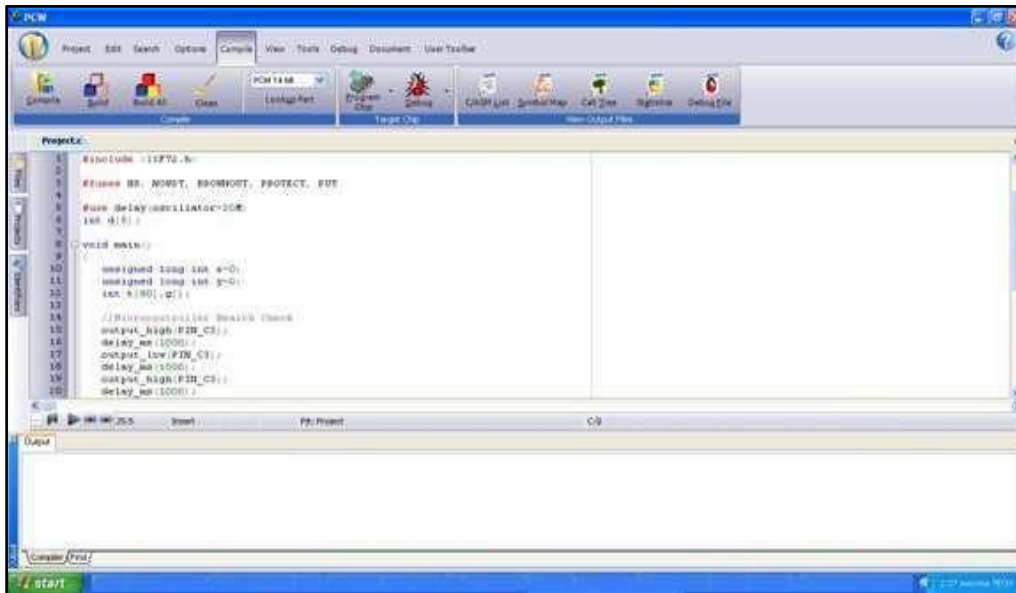


Figure 4: Picture of Compiling a New File Using the PIC C Compiler



Figure 5: Window Image of Checking Errors and Warnings using the PIC C Compiler

You can then start to write the source code in the window titled 'projects' then before testing your source code; you have to compile your source code and correct eventual syntax errors.

- **Dumping steps:** The steps outlining the procedure for transferring the edited program from Proteus 7 to the microcontroller: Prior to connecting the program dumper to the microcontroller kit, the initial window will be displayed as depicted:

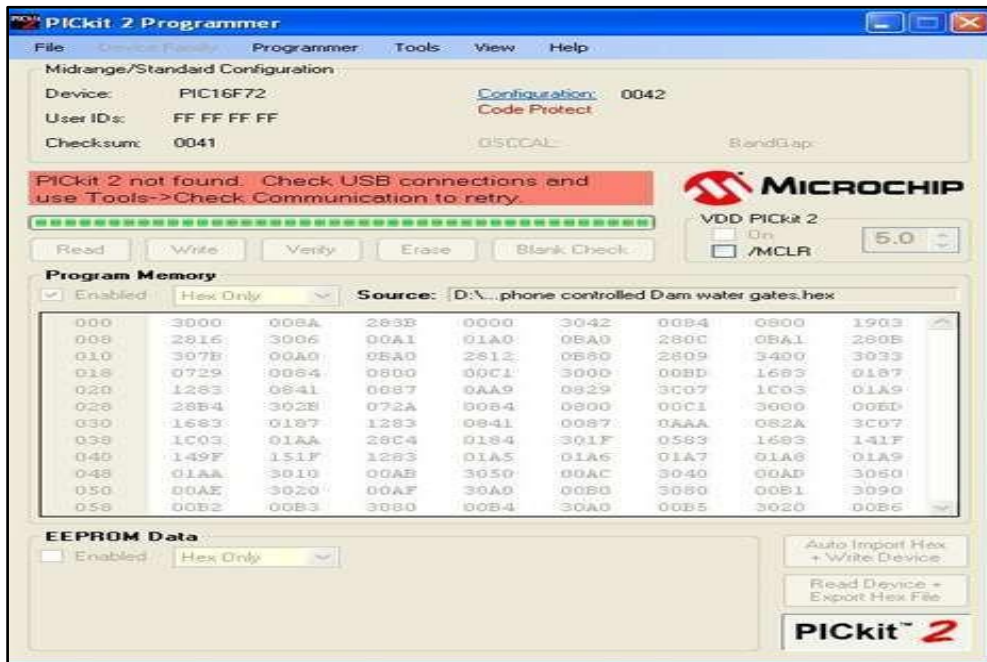


Figure 6: Window Image of Program Dumper Window

select the Tools option and click on Check Communication for establishing a connection as shown in the below window

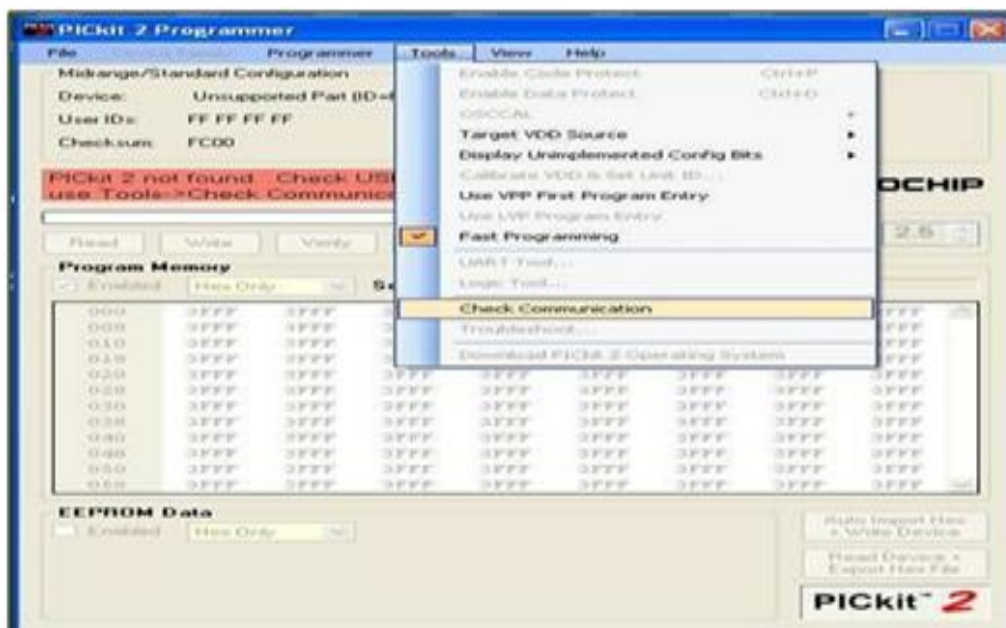


Figure 7: Window Image of Checking Communications Before Dumping the Program into Themicro Controller

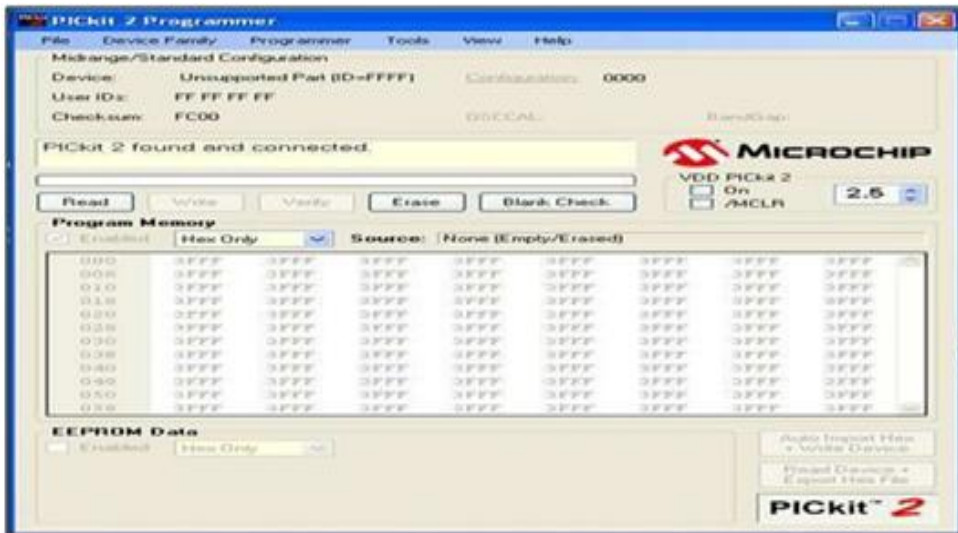


Figure 8: Window Image of after Connecting the Dumper to Themicro Controller

After connecting the dumper properly to the microcontroller kit the window appeared as shown below.

Again, by selecting the Tools option and clicking on Check Communication, the microcontroller gets recognized by the dumper and hence the window is as shown below.

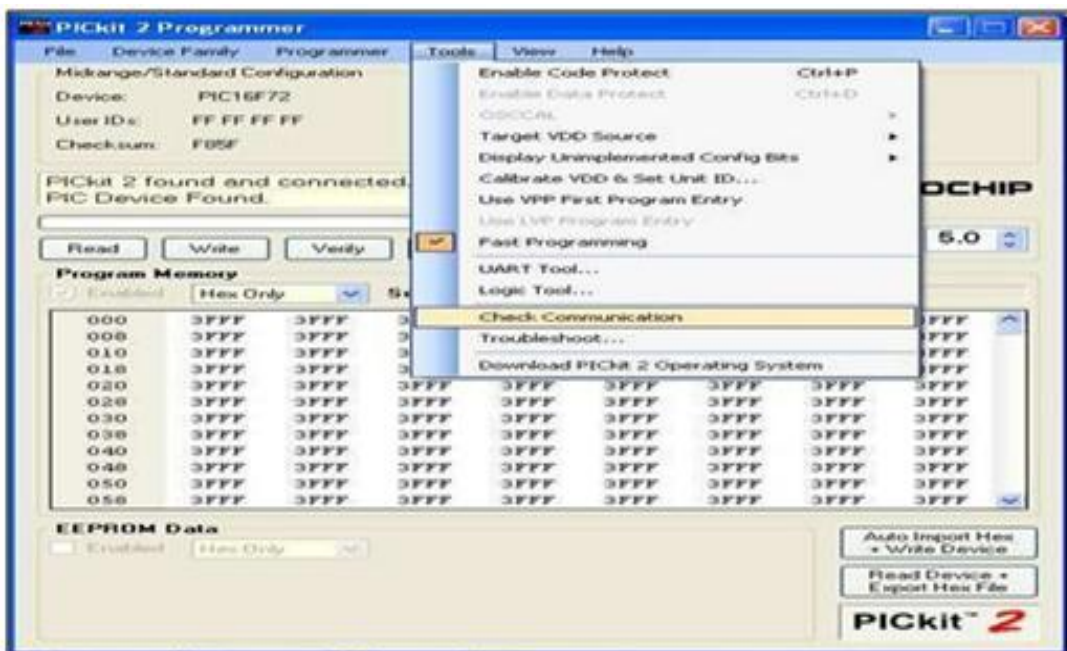


Figure 9: Window Image of Dumper Recognition to the Microcontroller

Import the program which is the '. hex' file from the saved location by selecting the File option and clicking on 'Import Hex' as shown below the window.

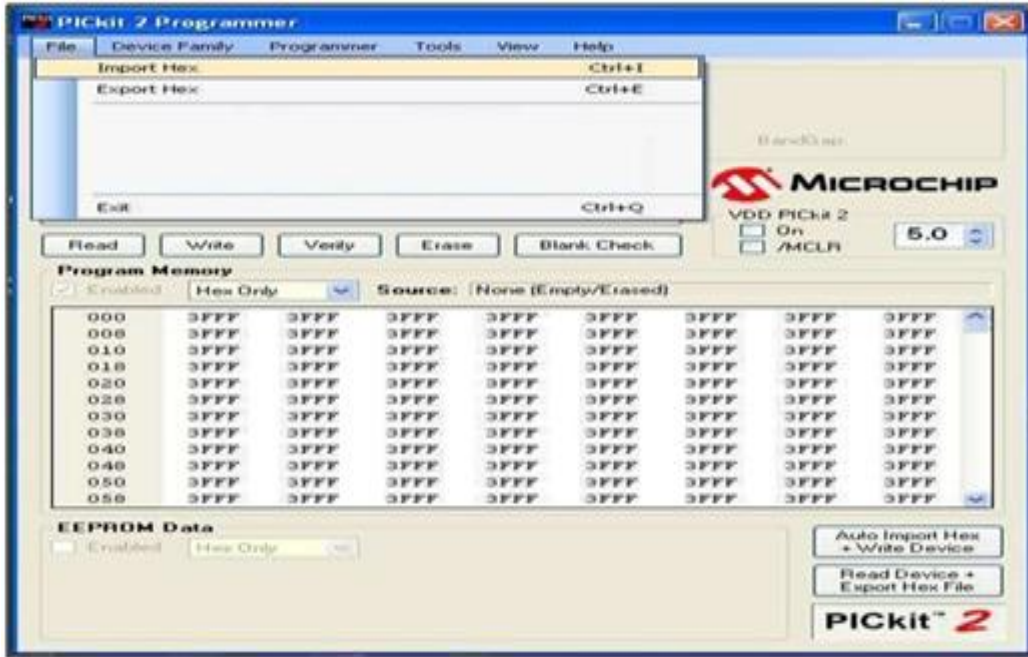


Figure 10: Window Image of Program Importing into the Microcontroller

Upon selecting the 'Import Hex' option, the subsequent step involves browsing to locate the program's file location. Once the 'prog hex' is identified, clicking on 'open' initiates the program's transfer into the microcontroller. Upon the successful completion of the program transfer, the interface assumes the appearance illustrated below:

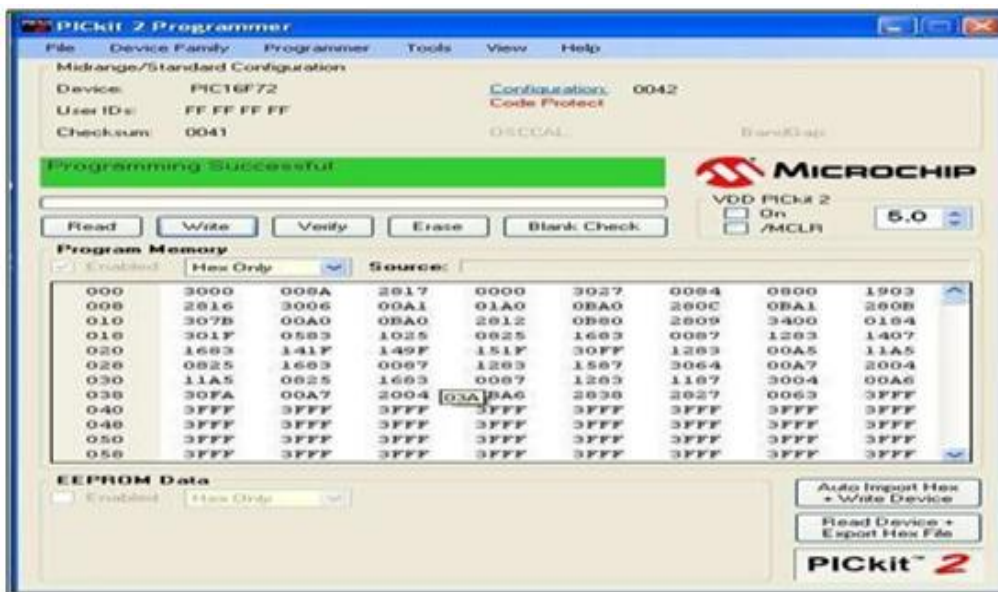


Figure 11: Window Image of after Program Dumped into the Microcontroller

IV. HARDWARE TESTING

- 1. Continuity Test:** In the realm of electronics, a continuity test is employed to verify whether an electric circuit allows the flow of current, indicating its completeness. This test involves introducing a minor voltage (linked in series with an LED or noise-producing element like a piezoelectric speaker) across the designated route. If the flow of electrons encounters hindrances such as severed conductors, impaired components, or excessive resistance, the circuit is deemed "open." Conducting continuity tests can be executed through instruments like multimeters, which gauge current, and specialized continuity testers, simpler devices often featuring a basic light bulb that illuminates when current traverses the circuit. A noteworthy application involves employing continuity tests to identify the endpoints of specific wires within a bundle. Negligible resistance is observed between the appropriate endpoints.

Typically conducted following the completion of hardware soldering and configuration, this test aims to uncover any electrical interruptions in the circuit post-soldering. Breaks in electrical continuity often stem from subpar soldering practices, mishandling of the printed circuit board (PCB), improper utilization of soldering equipment, component malfunctions, and discrepancies in the circuit diagram. A multimeter is employed for this test, configured in buzzer mode. The multimeter's ground terminal is connected to the ground, while both terminals are connected across the path under scrutiny. If continuity exists, a distinct beep sound becomes audible.

- 2. Power on Test:** This test is conducted to verify the adherence of voltage at various terminals to specified criteria. The procedure entails using a multimeter to assess the voltages. Begin by examining the voltage across the terminals to determine the presence or absence of power transmission.

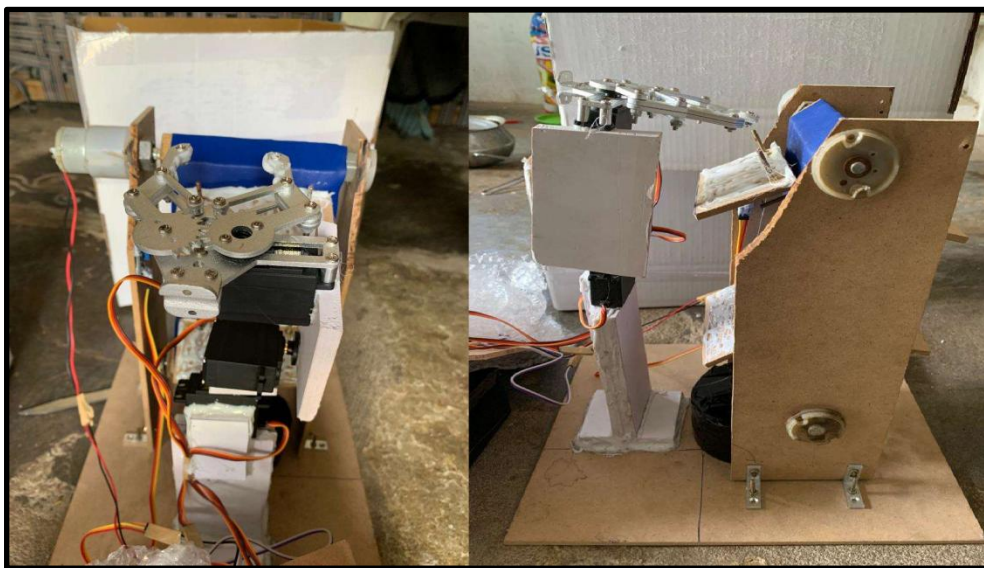


Figure 12: Fabricated Model of Servo-Controlled Pick and Place Robotic Arm

- 3. Subsequent Work:** This work has the potential for expansion through the integration of object sorting via color sensors, with future advancements focusing on enhancing the color sensor's efficiency. The sensor serves as a pivotal element of the project, facilitating precise object differentiation. Failure in its function could lead to incorrect material handling. As a result, it's imperative that the sensor exhibits exceptional sensitivity and the capability to discern between colors effectively. Subsequently, the image processing software system is harnessed to establish a statistical characterization of reflectance for each distinct information category. The Genetic Algorithm offers advantages such as extensive coding and decoding capabilities, enabling flexible conveyance of complex knowledge. Notably, the Genetic Algorithm excels in global optimization, particularly in scenarios with challenging objective functions, including those characterized by discontinuity or numerous local minima.

The MATLAB Genetic Algorithm toolbox simplifies usage, eliminating the need for lengthy code composition. This approach boasts swift execution times and produces visually interpretable results. The primary objective of this project revolves around achieving image classification through the utilization of MATLAB software.

Top of Form

MATLAB stands as a widely embraced software environment, particularly in the realm of robotics and automation research and teaching. This prominence can be attributed to its robust linear algebra capabilities, comprehensive toolbox collection that enhances its core functionality, and its interactive and open nature. The paper introduces a toolbox designed to provide direct access to tangible robotic and automation (R&A) hardware through the MATLAB interface. This innovation, when coupled with an existing robotics toolbox, significantly broadens its applications. Beyond the realms of robotic simulation and data analysis, users gain the ability to actively engage with the equipment in real-time. My personal experience with this tool underscores its value in both research endeavors and educational initiatives. For students, MATLAB offers the distinct advantage of requiring minimal training to commence its utilization, particularly when compared to other programming environments and languages such as Microsoft Visual C++ or Visual Basic

A novel strategy has been devised to achieve precise quality sorting of objects, exemplified by the sorting of apples within an agricultural factory. This innovative approach hinges on an image processing algorithm. The primary aim of this approach is twofold: firstly, to achieve meticulous object sorting based on their colors, and secondly, to proficiently identify any deviations in the color patterns surrounding the apples.

The effectiveness of this method has been assessed through experimentation, with obtained results compared to both human sorting procedures and color sensor sorting devices. In contrast to the conventional sorting method reliant on a combination of inductive, capacitive, and optical sensors for color differentiation, this novel approach integrates advanced mechatronics and image processing techniques.

Supported by Open CV, the image processing protocol functions by identifying circular objects within real-time webcam-captured images. Subsequently, it extracts pertinent color and position data. This data sequence serves as the basis for generating

sorting commands that direct a manipulator, responsible for executing the pick-and-place mechanism.

Extensive testing validates the efficiency of this color-based object sorting system, demonstrating 100% accuracy under optimal conditions encompassing appropriate illumination and the circular shape and color of objects, including silver, red, and black. However, in less-than-ideal conditions such as unspecified color variations, accuracy diminishes to 80%.

V. RESULT AND CONCLUSION

- 1. Result:** The work titled "Fabrication and Development of a Servo Controlled Pick and Place Robotic Arm" has been effectively designed and realized. The implemented model demonstrates proficiency in grasping and relocating objects. Integral to this model is the incorporation of an IR obstacle sensor coupled with a comparator and auditory alert system, effectively detecting the presence of objects. Additionally, the system is equipped with an LDR sensor to discern container replacement, facilitating the strategic placement of objects outside the container.
- 2. Conclusion:** The development encompasses the seamless integration of functionalities from all utilized hardware components. Each module's inclusion has been meticulously justified and strategically positioned, thereby optimizing the unit's overall performance. Leveraging cutting-edge integrated circuits (ICs) aided by advancing technology, the project has been skillfully executed. In this manner, the design and testing of the project have culminated in a successful outcome.
- 3. Future Scope:** The primary objective of the project "Fabrication and Development of a Servo Controlled Pick and Place Robotic Arm" is to establish a system based on an IR obstacle sensor that detects objects and employs a robotic arm to retrieve them from a vertical axis conveyor belt. Operational details entail the microcontroller transmitting signals to the robotic arm's motors, enabling the arm to grasp and relocate the object to a predetermined position. Triggered by the detection of an object's presence, the robotic arm navigates to the designated location, releases the object, and returns to its initial position. The system's functionality also encompasses container replacement verification through an LDR sensor.

Object detection is executed using an IR obstacle sensor, which incorporates both an infrared sensor and a phototransistor. This arrangement generates a square wave output signal when the object obstructs the infrared light directed towards the phototransistor. This signal is then conveyed to the Microcontroller. The microcontroller's basic firmware is scripted in Embedded C language. A comparator circuit is employed for obstacle comparison, accompanied by alerts through a buzzer alarm system. The system is propelled by a DC motor responsible for conveyor belt movement, while a servomotor is responsible for object retrieval and placement within the container at a specific angle.

Expanding upon this foundation, further enhancements can be introduced. These include technologies that address line-of-sight limitations and extend operational

distances. Additionally, feedback regarding the conveyer belt's status absence can be integrated. Future prospects involve the incorporation of additional components for diverse applications.

REFERENCES

- [1] Tsalidis, S., S., Dentsoras, A., J., "Application of design parameters space search for belt conveyor design" *Journal of Plant Science*, Vol. 10, No. 6, pp. 617-629, 2010.
- [2] Huang, T, Wang, P.F., Mei, J.P., Zhao, X.M., "Time Minimum Trajectory Planning of a 2- DOF Translational Parallel Robot for Pick-and-place Operations", *IEEE Computer Magazine*, Vol. 56, No. 10, pp. 365-368, 2007.
- [3] Sahu, S., Lenka, P.; Kumari, S.; Sahu, K.B.; Mallick, B.; "Design a color sensor: Application to robot handling radiation work", *Industrial. Engineering*, Vol. 11, No. 3, pp. 77-78, 2010.
- [4] Khojastehnazhand, M., Omid, M., and Tabatabaefar, A., "Development of a lemon sorting system based on color and size", *Journal of Plant Science*, Vol. 4, No. 4, pp. 122-127, 2010.
- [5] Dogan Ibrahim "Microcontroller Based Applied Digital Control", *International Journal of Science*, Vol. 23, No. 5, pp.1000- 1010, 2007.
- [6] Raj Kamal –Microcontrollers Architecture, Programming, Interfacing and System Design.
- [7] Mazidi and Mazidi –Embedded Systems.
- [8] PCB Design Tutorial –David. Jones.
- [9] PIC Microcontroller Manual – Microchip.
- [10] Pyroelectric Sensor Module- Murata.