ROBOTIC ARM, WRIST AND HAND USING PIC MICROCONTROLLER

Nurul Huzaimi Bin Mohamad Basir
Bachelor of Electrical Engineering (Hons)
Faculty of Electrical Engineering
Universiti Teknologi Mara
40450 Shah Alam

Abstract- **A mobile robotic arm that can be controlled using computer is proposed in this paper. For PC based controller, a graphical user interface (GUI) software is used to provide control capability on the mobile robot is developed using Visual Basic. The PC is connected to the mobile robot via serial port (RS232). The mobile robot also can be controlled using our portable control box. For hardware interfacing, Programmable Interface Controller (PIC) is used. The PIC receives the signal from the PC to provide motion control on the mobile robot. This PIC acts like analyzer that will analyze a serial signal from PC and produce a PWM (Pulse Width Modulation) combined with Electronic Speed Controller (ESC) that has high current spec to control high power motor. All of this combination will be used to control motor speed. It has been shown that the designed robot can grip up to 500 grams load. The robotic arm can be used in multi-purpose applications such as in industrial manufacturing and servicing robot.**

Keyword: **Graphical user interface, programmable interface controller, pulse width modulation, and electronic speed controller.**

1.0 INTRODUCTION

Nowadays many interesting events have been created and developed for the benefit of human being. Robot technology is widely recognized as one of the key technologies for the 21st century. In order to reduce human works robot has been created and the computer act as a brain. These phenomena can be clearly observed in the industries that mostly use robots in production, monitoring and others.
Contrarily, human's arms are capable of adapting to various tasks due to a different mechanical structure. Thus, several works proposed human-like robotic manipulators for the service robots [7, 8].

First of all, the robotic arm is designed as a three degree-of-freedom mechanical structure. The robotic arm is composed of shoulder frame, a shoulder part, an elbow part, and a wrist part.

The direct current motor is one of the important parts in this project because the motor is used to control the speed and movement of the arm robot. It converts electrical energy into mechanical rotation [1]. A direct current motor is an electric motor that converts electrical energy (direct current) into mechanical motion. Direct current motor is the most often used in most application such as industrial, automobile and mobile robots since it is easy for speed control and torque regulation.

A serial interface controller will be used to control the robot arm power windows motor (direct current motor) by using Mosfet H-Bridges circuit via serial connection to a personal computer or directly from the onboard PIC microcontroller [2].

Mosfet H-Bridges circuit is used to control the speed of the motor. A high current is needed to control the motor continuously. GUI software is develop using VB in order to control the arm robot or a controller box can be use to control the arm robot. The Mosfet H-Bridges act as a speed controller that will accept pulse width modulation (PWM) from PIC. A high current Mosfet is needed to control the power window motor continuously. Heat is one factor that has to be face in building this circuit. Mostly Mosfet, especially N type produce heat as pulse width modulation (PWM) is applied on one of its input.

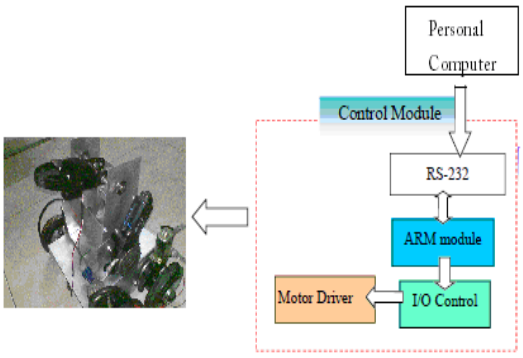


Fig. 1. System Architecture of Robotic Arm

Usually this H-bridge is a combination of two inverter of P and N channel. One act as source current circuit and one act as sinking current circuit. As sinking occur at N-type Mosfet, heat produce. This type of heat will damage the internal Mosfet circuitry. In order to overcome this problem several circuit that will act as a sinking circuit has to be added. The objective of this circuit is to protect the two N-type Mosfet from damaging itself. The new circuit consists of N-Channel Mosfet, resistor and n-p-n transistor.

The PIC is compiled with a program to control the movement of the robot. The PIC is program to react for any incoming input from GUI interface or controller box. The PIC will constantly update the motor with required speed to drive the arm robot or wheel to the position required. A few external components needed to be checked and test whether their values correspond to the desired ones, and finally to see whether all the connections are done right.

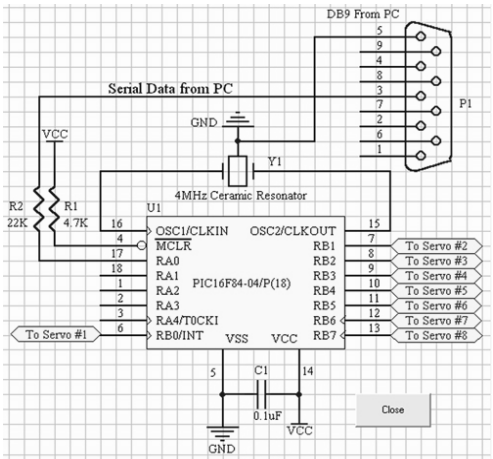


Fig. 2. Serial Interface Controller (The PIC16F84A is replaced with PIC16F877A)

The objective of this project is to develop a teleoperated robot using Microsoft Visual Basic. A program is created specially designed to control motor speed from computer. Most likely the true functions of this robot are to create human virtual interface software as a controller and for this project the robot will depend fully on human to give a signal from the computer. The signal from the computer then will be analyze by PIC Microcontroller. It will generate a Pulse Width Modulation (PWM) to control motor speed. There are several steps to complete the objective requirement and the primary goal is to design a simple and economy speed controller direct current motor using computer. The direct current motor will be connected to the computer through a serial port.

2.0 METHODOLOGY

The main objective is to build a PIC microcontroller circuit as in Fig. 2 to listen for any incoming serial communication from the host computer, and then set (servos to the position received / Mosfet H-Bridges speed required). The PIC is programmed to constantly update the power windows with their positional speed required so that the power windows will hold their speed. These operations explain how remote control ESC work and how they are controlled will make a new understanding and gallant my knowledge. The R/C servo is geared, (power windows is direct from ESC), direct current (DC) motor with a built-in positional feedback control circuit. This makes it ideal for use with small robot (use servos) but ideal for medium robot if uses high power motor such as power windows.

The PIC 16F877A software for the serial interface controller operates by receiving information in a serial format from a host computer (teleoperated robot). The PicBasic Pro compilers, by microEngineering Labs, include an instruction called *serin* command to received data. *Serin* must receive the qualifier bytes in exact order before receiving any data items. If any of the bytes received do not match the bytes in the qualifier sequences, the qualification process starts over.

Another feature of the *serin* command is the ability to set a timeout and label. The timeout qualifier allows the program to continue execution at the point specified by label if a character is not received within a certain amount of time. Timeout is specified in 1 ms units. This is ideal because it gives the program the ability

to constantly update the servos with positional information if nothing is happening on the communication port.

For the program, the ASCII value of 255 is used as the qualifier to identify that data are being sent from the host computer. Once the ASCII value of 255 has been received, the next byte of information will be stored in a variable called slider. This variable determines which servo will be updated. The next byte of information is stored in variable called control and contains the position to which the servo will be set. The line code is shown below:

Serin com_in,baud,5,set_pos,[255],slider,control

The serial data are received on pin com_in (PORTC.0) at a baud rate of baud (2400). If nothing is received on pin 0 of port A within 5 ms, then program execution will continue at set_pos, where the servos are set. When the serial data that determine which servo and its position are received, the PIC will then transmit the data back to the computer pin com_out (PORTC.1). The serial servo control program running on the computer will use this information to verify that the PIC has received the correct information and will then send the next set of servo and position information. The program execution will remain in a tight loop af receiving servo position data and then setting the servos with the data received [4].

This robot is design with 3 degree-of-freedom mechanical structure as shown in Fig. 3.

Fig. 3. Robotic Arm Mechanical Structure

To communicate with the robot via serial interface circuit board, MSComm 6.0 control will be added. The MSComm control was one of the first custom controls designed for Visual Basic and is the original means of communications for my application by allowing the transmission and reception of data through the serial ports. The MSComm control provides serial communications for my application by allowing the transmissions and reception of data through the serial ports. The MSComm control provides two ways to handle communications. The first is event-driven communications, a very powerful method for handling serial port interactions. The second method of handling communications is by polling for events and errors by checking the value of the CommEvent property after each critical function of your program [6].

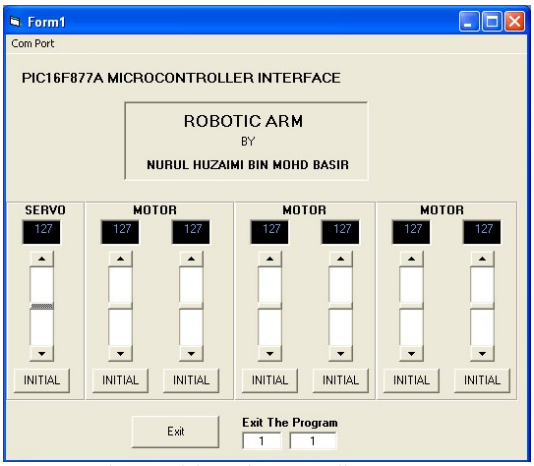


Fig. 4. Serial Interface Controller Program

The PIC16F877A microcontroller has a built in PWM command that is used to send a pulse width modulation signal on one of its output port. The code to send PWM signal through one of the output port is:

PWM Pin, Duty, Cycle

Outputs a pulse width modulated pulse train on Pin. Each cycle of PWM consists of 256 steps. The Duty cycle for each PWM cycle ranges from 0 (0%) to 255 (100%). This PWM cycle is repeated Cycle times. Pin may be a constant, 0 - 15, or a variable that contains a number 0 – 15 (e.g. B0) or a pin name (e.g. PORTA.0).

The Cycle time of PWM is dependent upon the oscillator frequency. If a 4MHz oscillator is used, each Cycle is about 5ms long. If a 20MHzoscillator is used, each Cycle is about 1ms in length. Defining an OSC value has no effect on PWM. The Cycle time always changes with the actual oscillator speed [1]-[4].

The H-Bridges as in Fig. 5 uses MOSFETs for one main reason – to improve the efficiency of the bridge. When BJT transistors (normal

transistors) were used, they had a saturation voltage of approximately 1V across the collector emitter junction when turned on. The power supply voltage was 10V and consuming 2V across the two transistor required to control the direction of the motor. 20% power was eaten up by the transistor.

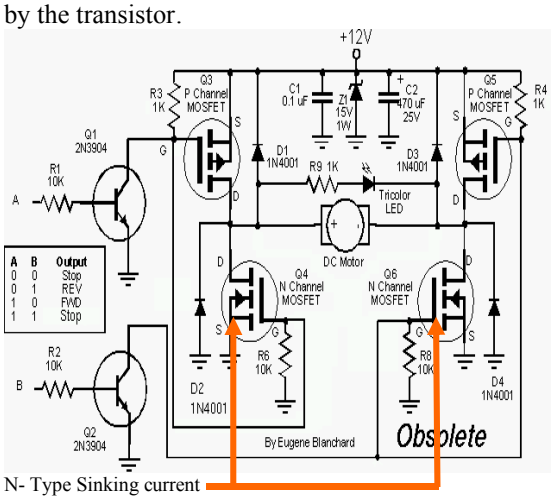


Fig. 5. Mosfet H- Bridge P-type and N-type sinking current.

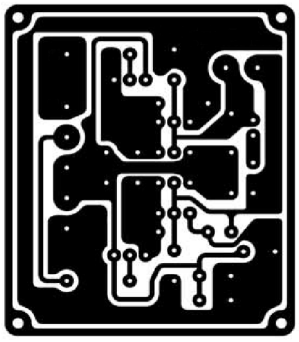


Fig. 6. PCB layout for Mosfet H-Bridge

MOSFETs were chosen because when they turn on they have an ON resistance called RDS (on). This is the resistance between the Drain and Sources when turned on. It is quite easy to buy MOSFETs that have very low RDS (on) ratings of less than 0.1 ohm. At 4 amps, this would mean that the voltage drop would be 0.4V per MOSFET, a definite improvement. The MOSFETs chosen had (on) rating of 0.04 ohm which greatly improved the efficiency. Now, when MOSFETs has a low RDS (on) rating, it usually has quite a high current rating typically in the 10s of amps.

Naturally, the lower the RDS (on) rating, the more expensive the MOSFET Low RDS (on) P channel MOSFETs are more difficult to find than N channel. MOSFETs work by applying a voltage to the Gate. They called this transconductance. When a positive voltage greater than the Gate threshold voltage is applied, the MOSFET turns on (Q4 & Q6 – N channel only). The P channel works in reverse (See Q3 & Q5).

Relays circuit are used in this project. It is basically as same function as H-Bridge, which is to move the motor. But the difference is that relays cannot control the speed of the motor like H-Bridge circuits.

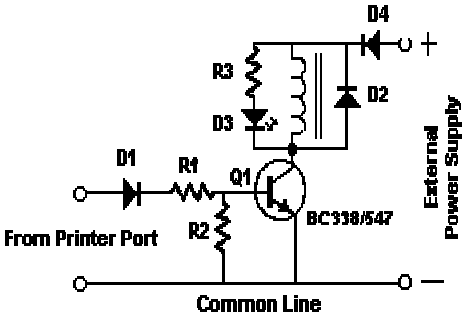


Fig.7. Relay Switching

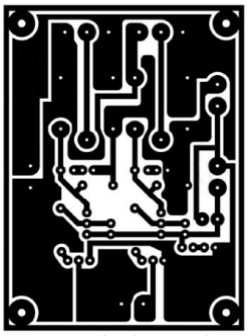


Fig. 8. PCB Layout for the 8x Relay switching

When one of the lines is taken HIGH (by a command from PIC output) 5 volts appears at the base of the transistor Q1. This switches on the transistor allowing a larger current to flow through the coil of relay RL1, turning it on. Taking PIC port line LOW will turns the relay off again Diode D1 prevents any 'reverse' potential from entering PIC and causing damage.

3.0 RESULTS AND DISCUSSION

A test was conducted by attaching the motors for mobility and movement. For the gripping system, a motor is needed for arm motion and one servo is needed for gripping.
The mobility system consists of 4 DC motors. Power window motors were chose for the mobility purpose. Each of the power windows

controls one tire or one sprocket that produced enormous torque. The reasons why the power windows were proposed in this project are:

1. The mobile robot is slightly heavy. As a result it requires high torque for movement.
2. The environmental structure that unnoticed because of the earth environment such as slope that required high torque.
3. The application that this robot handle consists of different structure and different environments

An experiment on burden measurement is conducted to measure how much weight the gripper can hold or raised.

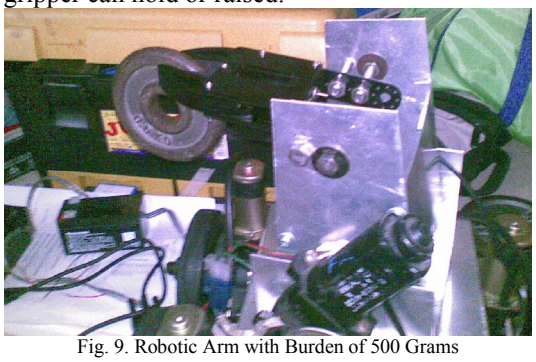


Fig. 9. Robotic Arm with Burden of 500 Grams

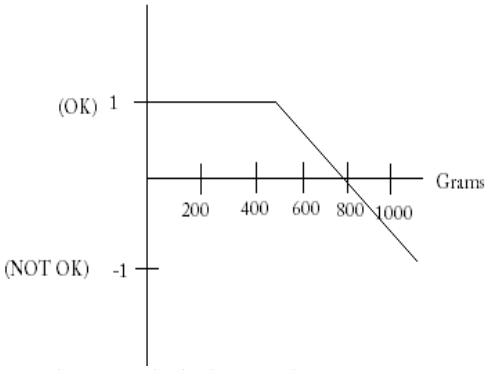


Fig. 10. Graph of Gripper Burden Measurement

The graph in Fig. 10 explains that the gripper can hold the amount of burden for about 500 grams. The amount exceed than that will cause the gripper loose the strength to hold.

This is because of the servo that is used with the gripper. The servo produces low torque which results in low power to produce tight grip. The only solutions is that the servo is replace with direct current motor to have high output power thus have strong grip.

The gripper can hold thing which is not exceed 40mm of diameter. The gripper can open about 40mm only as shown in Fig. 10.

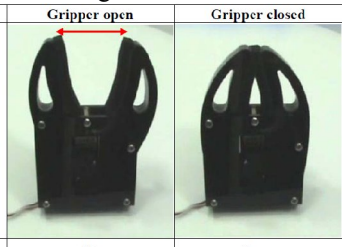
	Gripper open	Gripper closed
Gripper		
Opening Distance	~40mm	0mm

Fig. 11. Robot Gripper

Gripper is used to pick up dangerous or suspicious items for the robot to carry, some can turn doorknobs, and others are designed to carry only very specific things like beer cans.

Closing too tightly on an object and crushing it is a major problem with this gripper. There must be some way to tell how hard is enough to hold the object without dropping it or crushing it.

For these reasons, gripper design requires as much knowledge as possible of the range of items the gripper will be expected to handle. There are several basic types of gripper geometries. The most basic type which is used in this project has two simple jaws geared together so that turning the base of one turns the other. This pulls the two jaws together.

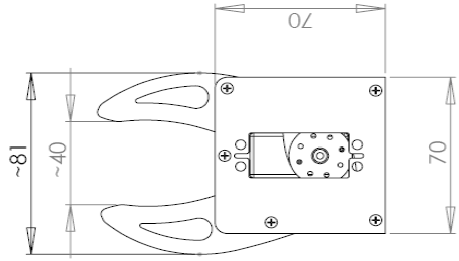


Fig. 12. Simple Direct Drive Swinging Jaw

In order for the gripper to open larger than before, a simple driven through a right angle drive gear motor could be used as shown in Fig. 13 which places the drive motor further out of the way of the gripper. This and similar designs have the drawback that the jaws are always at an angle to each other which tends to push the thing being grabbed out of the jaws[9].

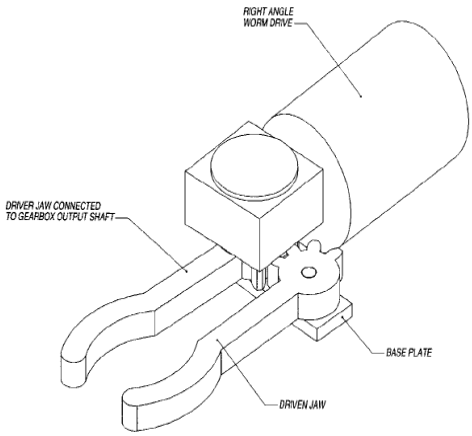


Fig. 13. Simple direct drive through right angle worm drive gearmotor

4.0 CONCLUSION

In this project, an economic mobile robot that can be controlled using computer via serial interface controller has been designed and developed. Basically the project can be divided into two parts, which is hardware and mechanical development for mobile robot and software for designing an interfacing for controlling the mobile robot. For interfacing controlling method, Graphical User Interface (GUI) software was developed using Visual Basic. The instructions from the computer will be sent through the serial port communication to provide control capability on the mobile robot. The PC is connected to the mobile robot via serial port (RS232). For hardware interfacing, programmable interface controller (PIC) is used. The PIC receives the signal from the PC to provide motion control on the mobile robot. The fundamental of creating a serial communications project using Visual Basic was covered. It gives the ability to craft the robot control software and the ability to harness the computing power of the personal computer.

6.0 FUTURE DEVELOPEMENT

Upgrading the robot is the main objective for the future works. Therefore I hope that this robot will have low power consumption and the motor that are currently used will be substitute with a better in term of weight and power usage. The circuitry also will be upgraded with a wireless device that will give the robot a freedom to mobilize anywhere within the radius of the device. Full hardware utilization will be used to upgrade the robot in order to make it even better. A control box can be design to replace for the Visual Basic as a software interface and designs a standalone system. I hope that the design will be robust and less heavy for future work.

7.0 ACKNOWLEDGEMENTS

I would like to extend my appreciation to all of the people who have helped me in the completion of this project. I wish to thank the following people for their supports and contributions from the beginning of this project. Without them maybe this project could not be successfully complete. To Dr. Fuziah Sulaiman for turning my ideas into reality, to all my friends, I am grateful with their helps and also to my family for their support and give me the inspirations to settle this project and thank to the various components manufacturers for providing pertinent data sheet schematics and photographs. Finally, I would like to thank to the following reviewers for their constructive comments on early manuscript. Thank you.

8.0 REFERENCES

[1] K. Williams. Build Your Own Humanoid Robot, The McGraw-Hill Companies, Publication 2004, ISBN: 0-07-142274-9.
[2] K. Williams. Amphibionics, Build Your Own Reptilian Robot, The McGraw-Hill Companies, Publication 2003, ISBN: 0-07-142921-2.
[3] G. McComb. Constructing Robot Bases, The McGraw-Hill Companies, Publication 2003, ISBN: 0-07-140852-5.
[4] R. Stevens. Serial PIC'n : PIC Microcontroller Serial Communications, Square 1 Electronics, Publication May 3, 1999, ISBN-10: 0965416224
[5] D. R. Shircliff. Build A Remote-Controlled Robot, The McGraw-Hill Companies, March 25, 2002, ISBN: 0-07-138543-6
[6] R. Petrusha. Visual Basic 2005: The Complete Reference, McGraw-Hill, July 27, 2006, ISBN-10: 0072260335
[7] J.M. Lee, B.S. Park, Y.S. Lee, J.S. Ahn, S.H. Lee, S.J. Lim, C.S. Han, "The development of the robot manipulator for an intelligent service robot," International Joint Conference on SICE-ICASE, pp. 282-287, 2006.
[8] T. Taipalus, K. Kazuhiro, "Development of service robot for fetching objects in home environment," IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 451-456 2005.
[9] P. E. Sandin, Robot Mechanisms and Mechanical Devices Illustrated, Publisher: McGraw-Hill, 2003 ISBN: 0-07-142928-X