DUAL MODE MOBILE ROBOT: APPLICATION USING MATLAB GUI

SAIFULLAH BIN AHMAD

A thesis submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Electrical Engineering (Electronic)

Faculty of Electrical & Electronic Engineering
Universiti Malaysia Pahang

NOVEMBER 2008
ABSTRACT

Dual mode mobile robot has hardware and software part. For the software part, we use MATLAB GUI and for the hardware, it consists PIC 16F877A, sensor, LCD, relay and others. There are two mode to control the mobile using PIC 16F877A by interfacing with MATLAB GUI. The two modes are manual and autonomous mode. For the manual we control the mobile with GUI. For the autonomous mode, distance sensor is use to control the mobile and stop the mobile with certain distance that we have set up. The distance will display on LCD at hardware. Rs232 cable is use to connect hardware with software (PC). This project is focus on interfacing MATLAB software and hardware using rs232 cable. A graphical user interface (GUI) is a graphical display that contains devices, or components, that enable a user to perform interactive tasks. To perform these tasks, the user of the GUI does not have to create a script or type commands at the command line. Often, the user does not have to know the details of the task at hand. A PIC Microcontroller chip combines the function of microprocessor, ROM program memory, some RAM memory and input-output interface in one single package which is economical and easy to use.
# TABLES OF CONTENT

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Scope of Project</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>1.5</td>
<td>Thesis Organization</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>LITERATURE REVIEW</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Graphical User Interface (GUI)</td>
<td>6</td>
</tr>
<tr>
<td>2.1.1</td>
<td>GUI definition</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2</td>
<td>MATLAB GUI</td>
<td>7</td>
</tr>
<tr>
<td>2.1.3</td>
<td>MATLAB GUIDE</td>
<td>8</td>
</tr>
<tr>
<td>2.1.4</td>
<td>GUI operation</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>PIC Microcontroller</td>
<td>10</td>
</tr>
<tr>
<td>2.2.1</td>
<td>History of PIC</td>
<td>10</td>
</tr>
<tr>
<td>2.2.2</td>
<td>The PIC Microcontroller</td>
<td>11</td>
</tr>
<tr>
<td>2.2.3</td>
<td>PIC Basic Pro Compiler</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>DC Motor</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1</td>
<td>History of DC Motor</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2</td>
<td>DC Motor Operation</td>
<td>14</td>
</tr>
</tbody>
</table>
2.3.3 Advantage of DC Motor 15

2.4 Sensor 16

2.4.1 Definition of Sensor and the Application 16

2.4.2 SHARP GP2Y0A21YK0F 16

2.5 Relay 18

2.6 LCD 19

2.7 MAX 233 20

2.8 Darlington Transistor C1815 20

3 METHODOLOGY 21

3.1 Introduction 21

3.2 Software Development 22

3.2.1 Development MATLAB GUI Using MATLAB GUIDE 24

3.2.2 Build MATLAB Programming 29

3.2.3 Build PIC programming 33

3.3 Hardware Development 35

4 RESULT AND DISCUSSION 41

4.1 Introduction 41

4.2 Main Menu of GUI 41

4.3 Interface MATLAB GUI Software 45

4.4 User information GUI 48

4.5 Result 48

4.5.1 DC motor 48

4.5.2 Sensor (GP2Y0A21YK0F) 49

4.5.3 LCD Display 50

5 CONCLUSION AND RECOMMENDATION 52

5.1 Conclusion 52
5.2 Future Recommendation 53
5.3 Costing and Commercialization 53

REFERENCES 55
Appendices 57-72
CHAPTER 1

INTRODUCTION

1.1 Overview

This project is focus on designing the Graphical User Interface (GUI) through MATLAB to control the DC motor using PIC. The PIC is a programmable interface devices or controller between PC (MATLAB GUI) and the DC motor. The main contribution of this project is the interfacing of the MATLAB with PIC and Graphical User Interface (GUI).

The Peripheral Interface Controller (PIC) use in this project is as controller device between Personal Computer, analog distance sensor and the DC motor. The PIC is use because of wide availability and economical. Beside that PIC is a free development tools and can perform many function without needed extra circuitry. PIC also have analog to digital converter that will be use to connect with analog distance sensor.
The purpose using MATLAB in creating the GUI is because it already has Graphical User Interface Development Environment (GUIDE) that provides a set of tools for creating GUI. These tools simplify the process of lay out and programming GUIs. The GUI create in MATLAB with appropriate coding will control the DC motor via serial port that interface with the PIC.

The GUI create in MATLAB with appropriate coding will control the DC motor via serial port that interface with the PIC. There are many advantage by using the DC motor, among that the DC motor has no adverse effect on power quality and the speed is proportional to the magnetic flux.

This project is to control the mobile robot by using GUI in MATLAB and PIC controller. There are two modes to control the mobile robot. The first mode is control mobile robot manually. For the second mode (autonomous), to control the mobile robot we use analog distance sensor to detect the distance between the robot and wall. Then it will stop automatically in certain distance after detect the block or wall.

Figure 1.1: Block Diagram of Project
1.2 Objective

These projects have two main objectives. The objective of this project is to:

i. To control the mobile robot using GUI in MATLAB
ii. Able to interface the MATLAB GUI with hardware using PIC

The important part of this project is to interface the MATLAB GUI with the PIC. Then, important part of this project is to receive a signal from sensor that will transmit to MATLAB GUI and interface using PIC. After that, the programming will send the signal to control the mobile robot automatically.

1.3 Scope of project

The scopes of this project are laying out the GUI in MATLAB GUIDE and create programming for the GUI’s. Secondly Prepare the PIC circuitry and serial 3 connections (DB9) circuit for interfacing part. For the third part is to build IR sensor circuit and interface with PIC. And the last part is creating program for PIC using PICBasic Pro Compiler to control the DC motor.

For this project, there are two scopes. The scope of project is dividing to software part and hardware part:

For the software part, we have:

i. MATLAB programming
ii. PIC programming

iii. PICBasic Pro Compiler

For the hardware part, we have:

i. 2 ways serial parallel port (transmit and received input or output)

ii. PIC 16F877A

iii. DC Motor and other components

iv. Distance sensor 6V relays

v. LCD

1.4 Problem statement

The main objective in this project is to interface the MATLAB GUI with the PIC. It is a difficult part to develop the program for MATLAB and the PIC simultaneously to make the interfacing part. By using the PicBasic Pro Compiler software to develop programming to control DC motor, it can reduces the difficulty by comprises a list of statements that written in a programming language like assembler, C, or PBASIC. With this opportunity, the men in charge do not have to take long time to written and troubleshoot the program.

To interface MATLAB GUI with PIC controller we use RS232. For my project it will use bidirectional communication to transmit and receive the data. Previously, not much has develop application using PIC via RS232 bidirectional communication in MATLAB GUI due to its difficulties. The development of this project is for research purpose in MATLAB communication using rs232.
1.5 Thesis organization

This thesis consists of five chapters including of the first chapter. The contents of each chapter will explain details about this project.

Chapter 2 contains a detailed description each part of project. It will explain about the MATLAB and MATLAB GUI, PIC, sensor and DC motor. The project methodology is in Chapter 3. This will explain how the project is organized and the flow of the process in completing this project.

Chapter 4 will presents the expected result of simulation runs using MATLAB GUI interface with PIC. It also will show how the sensor will function to send an input signal to GUI and will transmit back to PIC to control the DC motor. In this chapter also will explain how the sensor works as automatic controller. Finally the conclusions for this project are presented in Chapter 5.
CHAPTER 2

LITERATURE REVIEW

2.1 Graphical User Interface (GUI)

2.1.1 GUI definition

A graphical user interface (or GUI, often pronounced "gooey"), is a particular case of user interface for interacting with a computer which employs graphical images and widgets in addition to text to represent the information and actions available to the user. Usually the actions are performed through direct manipulation of the graphical elements [1].

The first graphical user interface was designed by Xerox Corporation's Palo Alto Research Center in the 1970s, but it was not until the 1980s and the emergence of the Apple Macintosh that graphical user interfaces became popular. One reason
for their slow acceptance was the fact that they require considerable CPU power and a high-quality monitor, which until recently were prohibitively expensive [2].

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth [3]. A true GUI includes standard formats for representing text and graphics [3]. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. For example, when a mouse click occurs on a pushbutton, the GUI should initiate the action described on the label of the button [4].

Many DOS programs include some features of GUIs, such as menus, but are not graphics based. Such interfaces are sometimes called graphical character-based user interfaces to distinguish them from true GUIs [4].

### 2.1.2 MATLAB GUI

A graphical user interface (GUI) is a graphical display that contains devices, or components, that enable a user to perform interactive tasks. To perform these tasks, the user of the GUI does not have to create a script or type commands at the command line. Often, the user does not have to know the details of the task at hand [5].
The GUI components can be menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. In MATLAB, a GUI can also display data in tabular form or as plots, and can group related components [5].

There are two basic tasks in process to implement a GUI. The two basic tasks in process of implementing a GUI are:

i. Laying out a GUI where MATLAB implement GUIs as figure windows containing various styles of uicontrol (User Interface) objects.

ii. Programming the GUI, where each object must be program to perform the intended action when activated by the user of GUI [5].

2.1.3 MATLAB GUIDE

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs [6].

i. GUIDE is primarily a set of layout tools

ii. GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI

The M-file will provide a framework for the implementation of the callbacks, the functions that execute when users activate a component in the GUI [6].
2.1.4 GUI operation

The GUI is already associated with one or more user written routines known as callbacks. The execution of each callback is triggered by a particular user action such as, mouse click, pushbuttons, toggle buttons, lists, menus, text boxes, selection of a menu item, or the cursor passing over a component and so forth [7].

By clicking the button triggers the execution of a callback. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a callback [7].

This kind of programming is often referred to as event-driven programming. The event in the example is a button click. In event-driven programming, callback execution is asynchronous, controlled by events external to the software. In the case of MATLAB GUIs, these events usually take the form of user interactions with the GUI [7].

The writer of a callback has no control over the sequence of events that leads to its execution or, when the callback does execute, what other callbacks might be running simultaneously [7].

Callbacks:

i. Routine that executes whenever you activate the uicontrol object.

ii. Define this routine as a string that is a valid MATLAB expression or the name of an M-file.

iii. The expression executes in the MATLAB workspace.
2.2 PIC Microcontroller

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division. PICs are popular with developers due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and reprogramming with flash memory) capability[8].

2.2.1 History of PIC

The original PIC was built to be used with GI's new 16-bit CPU, the CP1600. While generally a good CPU, the CP1600 had poor I/O performance, and the 8-bit PIC was developed in 1975 to improve performance of the overall system by offloading I/O tasks from the CPU [8].

The PIC used simple microcode stored in ROM to perform its tasks, and although the term wasn't used at the time, it is a RISC design that runs one instruction per cycle (4 oscillator cycles) [8].

In 1985 General Instruments spun off their microelectronics division, and the new ownership cancelled almost everything which by this time was mostly out-of-date. The PIC, however, was upgraded with EPROM to produce a programmable channel controller, and today a huge variety of PICs are available with various on-
board peripherals (serial communication modules, UARTs, motor control kernels, etc.) and program memory from 512 words to 32k words and more[8].

2.2.2 The PIC16F877A Microcontroller

A PIC PIC16F877A Microcontroller chip combines the function of microprocessor, ROM program memory, some RAM memory and input-output interface in one single package which is economical and easy to use. The PIC devices generally feature is sleep mode (power saving), watchdog timer and various crystal or RC oscillator configuration, or an external clock [9].

Logicator system is designed to be used to program a range of 8, 18, 28 pin reprogrammable PIC microcontroller which provide a variety of input output, digital input and analogue input options to suit students project uses [9]. Reprogrammable “FLASH Memory” chips have been selected as the most economical for student use. If a student needs to amend to control system as the project is evaluated and developed, the chip can simply be taken out of the product and reprogrammed with an edited version of the floe sheet [9].

The features that available in PIC 16F877A:

i. General purpose i/o pins
ii. Internal clock oscillators
iii. 8/16 Bit Timers
iv. Internal EEPROM Memory
v. Synchronous/Asynchronous Serial Interface USART
vi. MSSP Peripheral for I²C and SPI Communications
vii. Capture/Compare and PWM modules
viii. Analog-to-digital converters
ix. USB, Ethernet, CAN interfacing support
x. External memory interface
xi. Integrated analog RF front ends (PIC16F639, and rfPIC)
xii. KEELOQ Rolling code encryption peripheral (encode/decode)

![Figure 2.1: IC Configuration](image)

2.2.3 PIC Basic Pro Compiler

The PicBasic Pro Compiler (or PBP) makes it even quicker and easier to program Microchip Technology’s powerful PICmicro microcontrollers (MCUs). The
English-like BASIC language is much easier to read and write than the quirky Microchip assembly language [10].

The PicBasic Pro Compiler is “BASIC Stamp II like” and has most of the libraries and functions of both the BASIC Stamp I and II. Being a true compiler, programs execute much faster and may be longer than their Stamp equivalents. PBP is not quite as compatible with the BASIC Stamps as our original [10].

The PicBasic Pro Compiler produces code that may be programmed into a wide variety of PICmicro microcontrollers having from 8 to 84 pins and various on-chip features including A/D converters, hardware timers and serial ports [10].

### 2.3 DC Motor

#### 2.3.1 History of DC Motor

At the most basic level, electric motors exist to convert electrical energy into mechanical energy. This is done by way of two interacting magnetic fields one stationary, and another attached to a part that can move. A number of types of electric motors exist, but most BEAMbots use DC motors in some form or another. DC motors have the potential for very high torque capabilities (although this is generally a function of the physical size of the motor), are easy to miniaturize, and can be "throttled" via adjusting their supply voltage. DC motors are also not only the simplest, but the oldest electric motors [11].
The basic principles of electromagnetic induction were discovered in the early 1800's by Oersted, Gauss, and Faraday. By 1820, Hans Christian Oersted and Andre Marie Ampere had discovered that an electric current produces a magnetic field. The next 15 years saw a flurry of cross-Atlantic experimentation and innovation, leading finally to a simple DC rotary motor. A number of men were involved in the work, so proper credit for the first DC motor is really a function of just how broadly you choose to define the word "motor"[11].

2.3.2 DC Motor Operation

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion [12].

Figure 2.2: Part of DC Motor
Every DC motor has six basic parts axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets [12]. The stator is the stationary part of the motor this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout with the rotor inside the stator (field) magnets [13].

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating[13].

2.3.3 Advantage of DC Motor

The greatest advantage of DC motors may be speed control. Since speed is directly proportional to armature voltage and inversely proportional to the magnetic flux produced by the poles, adjusting the armature voltage and/or the field current will change the rotor speed [13].
2.4 Sensor

2.4.1 Definition of Sensor and the application

A sensor is a device which measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. For example, a mercury thermometer converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. A thermocouple converts temperature to an output voltage which can be read by a voltmeter. For accuracy, all sensors need to be calibrated against known standards [14].

Sensors are used in everyday objects such as touch-sensitive elevator buttons and lamps which dim or brighten by touching the base. There are also innumerable applications for sensors of which most people are never aware. Applications include automobiles, machines, aerospace, medicine, industry, and robotics [14].

A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes. For instance, if the mercury in a thermometer moves 1cm when the temperature changes by 1°, the sensitivity is 1cm/1°. Sensors that measure very small changes must have very high sensitivities [14].

2.4.2 SHARP GP2Y0A21YK0F
GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor [14].

![Figure 2.3: Analog distance sensor](image)

The features that available in this sensor type:

i. Distance measuring range : 10 to 80 cm
ii. Analog output type
iii. Package size : 29.5×13×13.5 mm
iv. Consumption current : Typ. 30 mA
v. Supply voltage : 4.5 to 5.5 V
2.5 Relay

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches. Relays allow one circuit to switch a second circuit which can be completely separate from the first [15].

Figure 2.4: Coil and switch contact

Figure 2.5: 6V relays
In this project relay use to supply the voltage to DC motor. 6V relays need more than 6V voltage to energized the contact. The output voltage from PIC is use to energized the contact of relay [15].

2.6 LCD

A liquid crystal display (LCD) is an electro-optical amplitude modulator realized as a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power [16].

![Figure 2.6: Liquid crystal display](image)

LCDs with a small number of segments, such as those used in digital watches and pocket calculators, have individual electrical contacts for each segment. An external dedicated circuit supplies an electric charge to control each segment. This display structure is unwieldy for more than a few display elements [16].