DC MOTOR CONTROLLER USING LINEAR QUADRATIC REGULATOR (LQR) ALGORITHM IMPLEMENTION ON PIC

NUR IZNIE AFRAH BINTI MOHD ISA

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2008

ABSTRACT

Linear Quadratic Regulator (LQR) algorithm is one of the controller methods to control a system. In this project, the LQR was implemented on the PIC microcontroller to control the dc motor. The main objective of this controller is to minimize the deviation of the speed of dc motor. Dc motor speed is controlled by its driving voltage. The higher the voltage, the higher the motor speed. The speed of the motor is specifying that will be the input voltage of the motor and the output will be compare with the input. As the result, the output must be the same as or approximately the same as the input voltage. In this project, the LQR algorithm was implemented on the PIC microcontroller so the result can be shown. Before the implementation on the PIC, the dc motor state-space has to be derived. Then, from the state-space, we can design the LQR controller by using the MATLAB software. The stable system is got by tuning the Q and R value that can be seen by the simulation.

CHAPTER 1

INTRODUCTION

1.1 Background

DC motor can be controlled either by software or directly by hardware. Software controlling needs computers which are bulky and common man cannot afford for it, so hardware controls are in use. Even in hardware if it is programmable device then it is preferred because it can be modeled according to the requirements of the user.

This project needs to develop the DC motor controller which is brushless servomotor by using Linear Quadratic Regulator (LQR) that is implemented on Programmable Integrated Circuit (PIC). To make the motor operate, first we must make the program on PIC. On the program, the input has to be set up. Then the input voltage is converted from analog signal into digital signal by ADC in the PIC. The PWM (Pulse Width Modulation) function of PIC is used for the electric current control to drive a motor.

The function of Linear Quadratic Regulator (LQR) is to minimize the deviation of the speed of the motor. The speed of the motor is specifying that will be the input voltage of the motor and the output will be compare with the input. The output must be the same as or approximately the same as the input voltage.

The advantages of used LQR are it is easy to design and increases the accuracy of the state variables by estimating the state. The nice feature of the LQR control as compared to pole placement is that instead of having to specify where n eigenvalues should be placed a set of performance weighting are specified that could have more intuitive appeal. The result is a control that is guaranteed to be stable.

1.2 Problem Statement

The speed of the DC motor may change due to disturbance present surrounding it. This will make the desired speed sometimes change and will be not maintain. By using LQR control algorithm, the deviation of the speed can be minimized.

1.3 Objectives

The objectives of this project are;

i. To implement Linear Quadratic Regulator (LQR) controller in Programmable Integrated Circuit (PIC) which can minimized the error of the speed of dc servomotor. The PIC that has been used in this project was PIC16F84A.

ii. To develop dc motor controller by using LQR algorithm.

1.4 Scope

This project actually concentrates on derivation of the mathematical model of dc servomotor and gets the value of K in LQR algorithm. K is the gain of the close loop system. To get the value of K, the state-space of the servomotor must be define first. So, LQR algorithm was used by means to minimize the deviation of the dc motor speed. The LQR is used to tune the value of Q and R. The value of Q and R is tuned the get the stable system.

CHAPTER 2

LITERATURE REVIEW

2.1 DC Motor

A DC motor is devised to convert electrical power into mechanical power. In DC motor, electrical energy is converted into mechanical energy through the interaction of two magnetic fields. One field is produced by permanent magnet assembly (on the start) and the other field is produced by an electrical current flowing in the motor winding (on the rotor). These two fields result in a torque that tends to rotate the rotor. As the rotor turns, the current in the windings is commutated to produce a continuous torque output.[1]

DC motor speed is controlled by controlling its driving voltage. The higher the voltage, the higher the motor speed. In many applications, a simple voltage regulation would cause lots of power loss in the control circuit, so a PWM method is used in many DC motor-controlling applications. In basic PWM method, the operating power to the motors is turned on and off to modulate the current to the motor. The ration of on time to off time is what determines the speed of the motor.

A PWM circuit can be implemented by using discrete components. However, this approach cannot provide the desired flexibility and controllability is expensive. A better implementation method for PWM circuitry is to use the PWM functions available in many microcontrollers today. Most of the PIC 18 devices have PWM functions.[1]

Controlling the speed of the motor is an important area to be considered. The speed of motor is directly proportional to the DC voltage applied across its terminals. Hence, if we control the voltage applied across its terminal we actually control its speed.

A PWM (Pulse Width Modulation) wave can be used to control the speed of the motor. Here the average voltage given or the average current flowing through the motor will change depending on the ON and OFF time of the pulses controlling the speed of the motor. The duty cycle of the wave controls its speed. [2]

For this project, Clifton Precision Servo Motor Model JDH-2250-HF-2C-E is used. The specifications of this DC motor are;

- Torque Constant: 15.76 oz-in. / A
- Back EMF: 11.65 VDC / KRPM
- Peak Torque: 125 oz-in.
- Cont. Torque: 16.5 oz-in.
- Encoder: 250 counts / rev.
- Channels A, B in quadrature, 5 VDC input (no index)
- Body Dimensions: 2.25" dia. x 4.35" L (includes encoder)
- Shaft Dimensions: 8 mm x 1.0" L w/flat

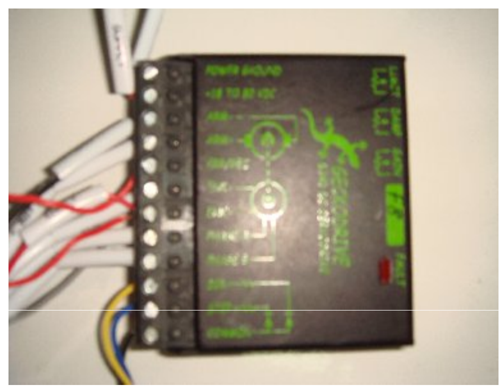

Figure 2.1 Clifton Precision Servo Motor Model JDH-2250-HF-2C-E with its encoder

2.2 Linear Quadratic Regulator (LQR) Algorithm

In layman's terms, Linear Quadratic Regulator (LQR) means the settings of a (regulating) controller governing either a machine or process (like an airplane or chemical reactor) are found by using a mathematical algorithm that minimizes a cost function with weighting factors supplied by a human (engineer). The "cost" (function) is often defined as a sum of the deviations of key measurements from their desired values. In effect this algorithm therefore finds those controller settings that minimize the undesired deviations, like deviations from desired altitude or process temperature. Often the magnitude of the control action itself is included in this sum as to keep the energy expended by the control action itself limited.[5]

In the particular case of a quadratic performance index combining the square of the error and square of the actuation, the solution to the optimal control problem is a feedback control where the measurements used for the feedback are all of the state variables. In this feedback control, each of the state variables is multiplied by a gain and the results are summed to get a single actuation value. The result of the LQR formulation is the set of gains, based on the relative weighting of the error and actuation in the performance index. [6]

Identification techniques based on minimization of equation error give good results, but it is essential that their sensitivity to high frequency noise and sample rate be compensated. This fact has generally been overlooked in previous literature on identification of flexible structures. Alternative approaches based on minimization of output error perform poorly, probably due to the existence of local minima in the performance index. [5]

Suppose that the space model is

$$\dot{x} = Ax + Bu, \tag{2.1}$$

with x(t) € R n , u(t) € R m. The initial condition is x (0).[6]

Suppose that we have sensors to measure the entire state and that we use a controller (regulator)

$$u = -K x \tag{2.2}$$

that seeks to drive the state to zero. You could use pole placement via Ackerman's formula. Here, we use the LQR methodology to specify the gain K. For this, let

$$J = \int_0^\infty (x^t Q x + u^T R u) \, d t \tag{2.3}$$

where Q = Q ≥ 0 and R = R > 0.The term "linear-quadratic" refers to the linear system dynamics and the quadratic cost function and we seek to find the gain vector K to minimize this "cost function."[7]

2.3 Programmable Integrated Controller (PIC) Microcontroller

A PIC is a **P**rogrammable **I**ntegrated **C**ircuit microcontroller, a 'computer-on-a-chip'. They have a processor and memory to run a program responding to inputs and controlling outputs, so they can easily achieve complex functions which would require several conventional ICs.[4]

The input voltage to PIC is converted by A/D converter. Changed voltage is used for the PWM function of the CCP to control the motor drive. At the circuit this time, a small motor is used as the generator to detect the number of rotations of the motor. The input voltage (the control voltage) to PIC is changed by the fluctuation of the number of rotations of the motor. The PWM (Pulse Width Modulation) function of PIC is used for the electric current control to drive a motor.[8]

Many methods evolved to control the revolution of a motor. DC motors can be controlled either by software or directly by hardware. Software controlling needs computers which are bulky and common man cannot afford for it, so hardware controls are in use. Even in hardware if it is programmable device then it is preferred because it can be modeled according to the requirements of the user.[2]

Advantages of using PIC over other controlling devices for controlling the DC motor are given below:

- **Speed:** The execution of an instruction in PIC IC is very fast (in micro seconds) and can be changed by changing the oscillator frequency. One instruction generally takes 0.2 microseconds.
- **Compact:** The PIC IC will make the hardware circuitry compact.
- **RISC processor:** The instruction set consists only 35 instructions.
- **EPROM program memory:** Program can be modified and rewritten very easily.
- **Inbuilt hardware support:** Since PIC IC has inbuilt programmable timers, ports an interrupts, no extra hardware is needed.
- **Powerful output pin control:** Output pins can be driven to high state, using a single instruction. The output pin can drive a load up to 25mA.
- **Inbuilt I/O ports expansions:** This reduces the extra IC's which are needed for port expansion and port can be expanded very easily.
- **Integration of operational features**: Power on reset and brown/out protection ensures that the chip operates only when the supply voltage is within specification. A watchdog timer resets PIC if the chip ever malfunctions and deviates from its normal operation.

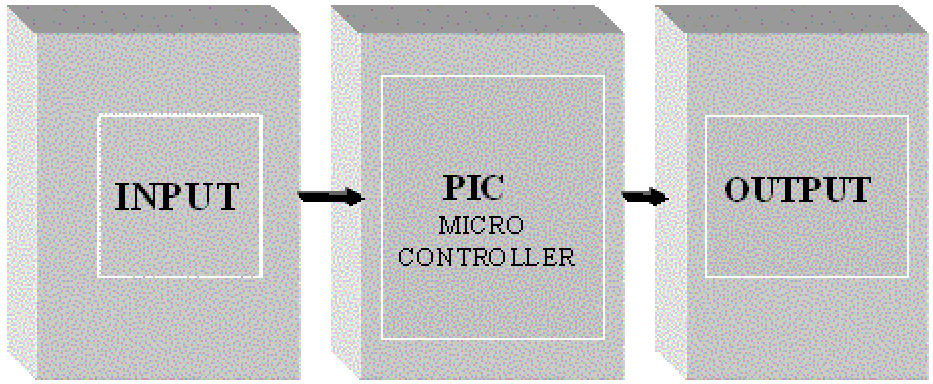

Figure 2.2 DC Motor Controlling System

The block diagram of the circuit is shown above. This circuit controls the speed and direction of the motor. The PWM (Pulse Width modulation) output from the four port pins is given to the H-Bridge circuit which drives the motor. On changing the duty cycle (ON time), we can change the speed. By interchanging output ports, it will effectively change direction of the motor. [9]

The PIC microcontroller is the brain of the circuit controlling all actions to be done. Inputs are given to control the speed and direction of the motor. The PIC output controls the DC motor.

CHAPTER 3

METHODOLOGY

This chapter explains on getting the state-space model of the dc servomotor, LQR design, hardware configuration and the implementation of LQR controller on PIC.

3.1 Program Flow Chart

The flow chart in Figure 4 shows the flow of this project. The mathematical modeling is doing to find the mathematical model for the DC motor where we will get the state-space model. This followed by getting the LQR controller from the state-space model. The MATLAB software is used to get the result. This mathematical modeling had to be done so the result that was get can be compare with the result from the experiment. The hardware design is needed to implement the DC motor with the PIC. The software that used to make the programming of the PIC is MicroCode Studio. Then, the integration between the software and hardware will run the motor.

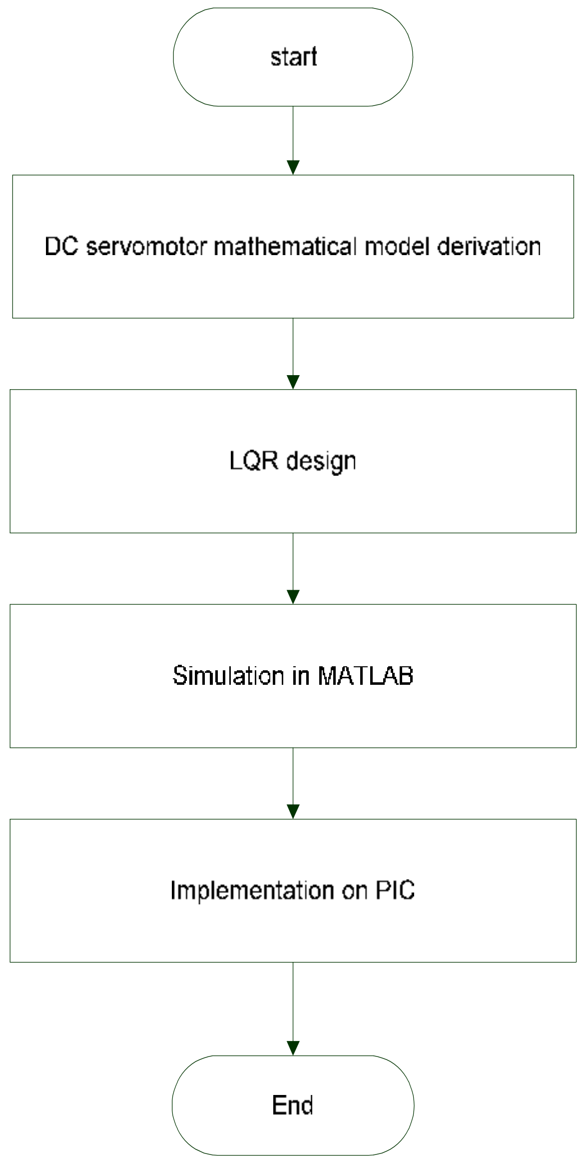

Figure 3.1 Flow chart of the project

3.2 Mathematical Model of DC Motor

The equivalent electrical circuit of a DC motor is illustrated in Figure 3.2. It can be represented by a voltage source (V_a) across the coil of the armature. The electrical equivalent of the armature coil can be described by an inductance (L) in series with a resistance (R) in series with an induced voltage (V_c) which opposes the voltage source. The induced voltage is generated by the rotation of the electrical coil through the fixed flux lines of the permanent magnets. This voltage is often referred to as the back emf (electromotive force).

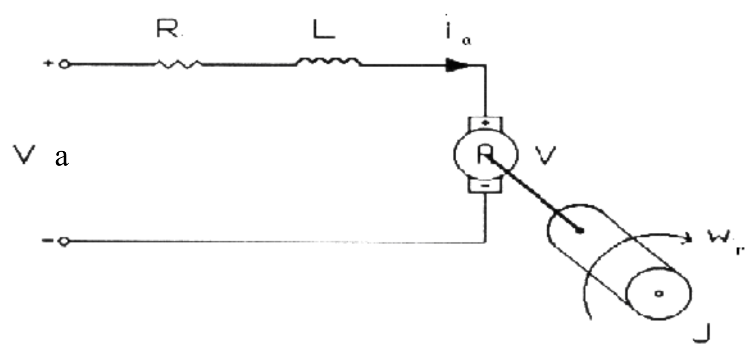

Figure 3.2 Electrical Circuit Representation Of A Dc Motor.

The physical parameters have been set up as followed;

Parameters	Values
Electric Resistance, R	2.7 Ω
Electric Inductance, L	0.004 H
Electromotive Force Constant, K	0.105 Nm A^{-1}
Moment of Inertia of the Rotor, J	0.0001 Kg m2
Damping Ratio of the Inertia of the Rotor, B	0.0000093 Nms rad^{-1}

3.2.1 Electrical characteristic

From the Figure 3.2, the following equations based on Newton's Law and Kirchoff's Law has been had.

$$\frac{di_a}{dt} = \frac{R}{L} i_a - \frac{K}{L}\omega_r + \frac{1}{L}V_a \qquad (3.1)$$

$$\frac{d\omega_r}{dt} = \frac{K}{J} i_a - \frac{B}{J}\omega_r \qquad (3.2)$$

3.2.2 Mechanical characteristic

The motor torque, T is related to the armature current, i_a by a constant factor, K and can be written as;

$$T = Ki_a \qquad (3.3)$$

3.3.3 State-space Representation

In the state-space form, the equations above can be expressed by choosing the rotating speed and electrical current as the states variables and the voltage as an input. The output is chosen to be the rotating speed.

$$\begin{bmatrix} \frac{di_a}{dt} \\ \frac{d\omega_r}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K}{L} \\ \frac{K}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i_a \\ \omega_r \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_a \tag{3.4}$$

$$= \mathrm{A}x + \mathrm{B}u$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_r \end{bmatrix} + [0] V_a \tag{3.5}$$

$$= \mathrm{C}x + \mathrm{D}u$$

By substituting the parameters on the state-space model, the value of A, B, C and D can be getting.

$$\begin{bmatrix} \frac{di_a}{dt} \\ \frac{d\omega_r}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{2.7}{0.004} & -\frac{0.105}{0.004} \\ \frac{0.105}{0.0001} & -\frac{0.0000093}{0.0001} \end{bmatrix} \begin{bmatrix} i_a \\ \omega_r \end{bmatrix} + \begin{bmatrix} \frac{1}{0.004} \\ 0 \end{bmatrix} V_a$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_r \end{bmatrix} + [0] V_a$$

$$A = \begin{bmatrix} -675 & -26.25 \\ 1050 & -0.093 \end{bmatrix} \qquad B = \begin{bmatrix} 250 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

3.3 LQR Design

To meet the desired specification, the controller is designed using LQR methodology. Let the cost function be defined as

$$J=\int_0^{\infty} (x^t Q x + u^T R u) dt \qquad (3.6)$$

Where Q is weighting factors of states (positive semidefinite matrix) and R is weighting factors of control variables (positive definite matrix). To design the LQR controller, the first step is to select the weighting matrices Q and R. The value R weight inputs more than the states while the value of Q weight the state more than the inputs. Then the feedback K can be computed and the closed loop system responses can be found by simulation.

The LQR controller is given by

$$u = - Kx \qquad (3.7)$$

where K is the constant feedback gain obtained from the solution of the discrete algebraic Riccati equation. The gain matrix K which solve the LQR problem is

$$K = R^{-1}B^TP^* \qquad (3.8)$$

Where P^* is unique, positive semidefinite solution to the Riccati equation;

$$A^TP + PA – PBR^{-1}B^TP + Q = 0 \qquad (3.9)$$

Using the equation (3.9), the value of P is;

[0.0051 0.0036; 0.0036 0.0035]

Then, the value of K can be compute by using equation (3.8) which is

[1.2814 0.9002]

3.4 Coding in MATLAB

To create the open-loop response, the following command is entered in MATLAB;

```
r = 2.7;
L = 0.004;
J = 0.0001;
k = 0.105;
b = 0.0000093;
A = [-r/L -k/L; k/J -b/J];
B = [1/L; 0];
C = [0 1];
D = [0];
step (A, B, C, D)
```

The state feedback is designed by using nominal values of parameters and the matrices Q and R is choose in such a way that both the state response of displacements x_1, x_2 unit initial conditions in both and to a delayed unit step exhibit nearly critical damping.

The K is found to minimize the J involves solving the Riccati equation. The MATLAB ‘lqr’ command is used to directly solve the gain vector K given A, B, Q and R.

The coding for getting the simulation of the controller was as followed;

```
r = 2.7;
L = 0.004;
J = 0.0001;
k = 0.105;
b = 0.0000093;
A = [-r/L -k/L; k/J -b/J];
B = [1/L; 0];
C = [0 1];
D = [0];
Q = diag([1 1]);
R = 1;
[K,S,e] = lqr(A,B,Q,R);
Ac = A - B*K;
step(Ac,B,C,D,1,t)
```

In order to improve the design, the various Q matrices and R values were tried. For Q matrices, [1 0; 0 10] and [1 0; 100] were used and for R, 0.1 and 0.01 were applied.

3.5 Hardware Design

3.5.1 PIC Microcontroller

It is used to control the rotation of the motor. This senses the input and process it using the program burned in it and gives the required PWM output on the required port pins. To control the speed we need to control the duty cycle (‘ON’TIME / PERIOD).

The output from the port A is given to the H-bridge. It is the arrangement of four transistors as shown below. Here four power transistors 2N3055 are used (C1, C2, C3, C4). At any time either C1, C3 or C2, C4 are made ‘ON’, hence current flows from the source (V_{dd}) to ground (Vss) through the motor. The direction of the motor is dependent on the polarity of the current. Hence by changing the ‘ON’ transistor pairs we can control the direction of the motor. To ‘ON’ the transistor we need to give high to its base. On controlling output coming out of port we can achieve the control over the motor rotation direction.

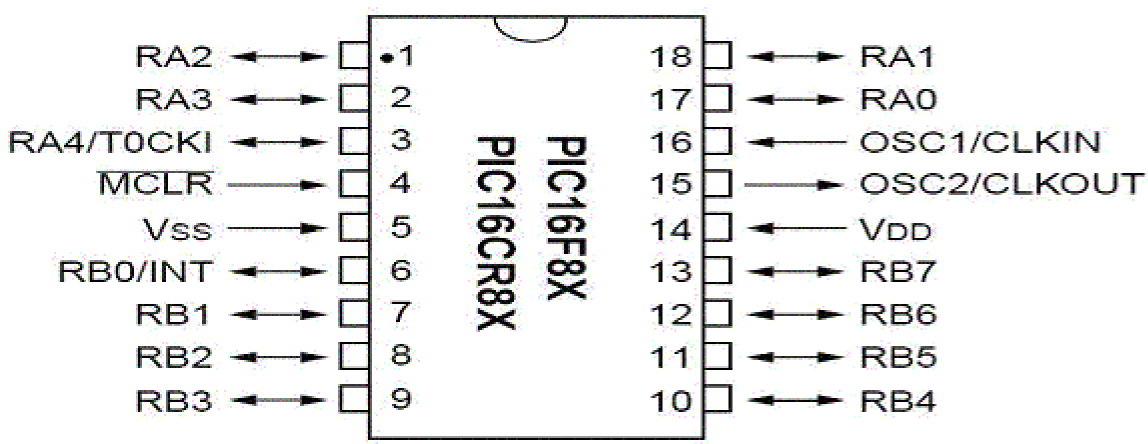

Figure 3.3 PIC16F84A IC Pin Diagram

3.5.2 Power Supply Circuit

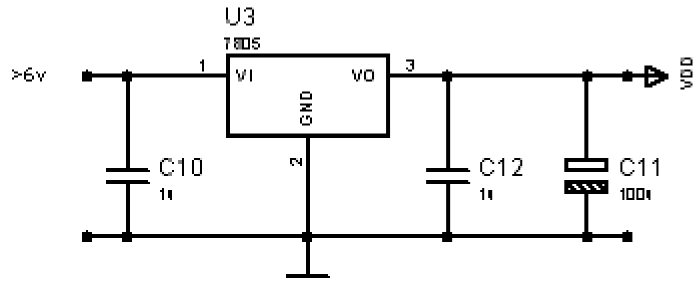

Figure 3.4 Power Supply Circuit

This circuit is very important as it was supply the power to the all component. Three terminal regulator is used to get the operating voltage for PIC. The left pin is for input voltage, the middle pin second is connected to ground while the right pin is 5V output voltage which connected to power the PIC.

3.5.3 Complete Circuit

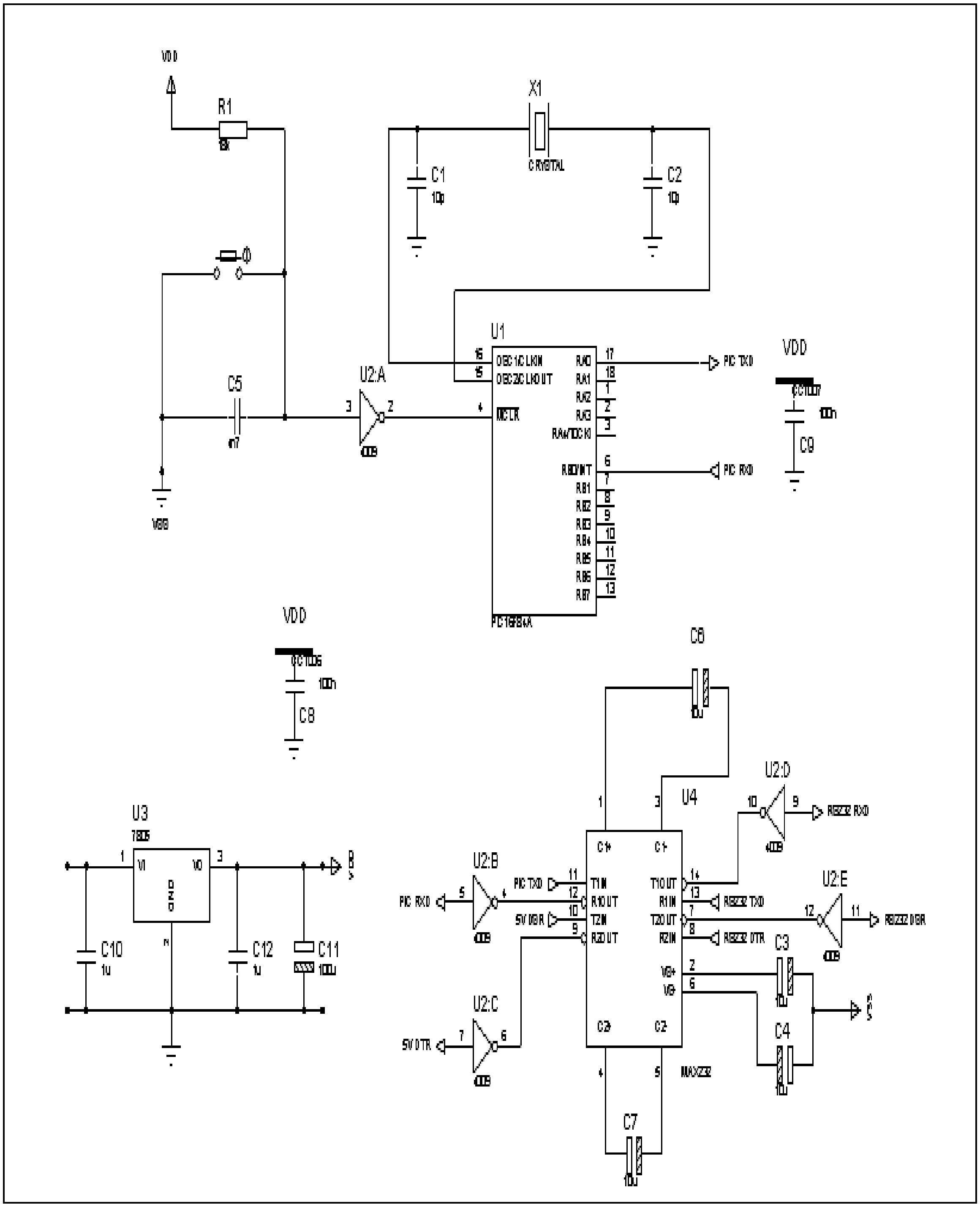

Figure 3.5 Complete Circuit Diagram

The circuit is built around the PIC IC 16F84A. The circuit is built as shown in Figure 3.5. The power supply of 5V and ground is given to appropriate pin of the PIC IC and to the H Bridge circuit. A 4M Hz crystal oscillator is connected as shown in the figure. Other type of oscillators can also be used. The crystal oscillator is more stable compare to other types. Oscillator acts as a clock source for the PIC IC operation.

The port B's four pins are given to the H-Bridge as shown in the figure. These pins are the control lines (PWM output lines) given to H-Bridge to rotate in the desired speed.

3.6 Software Development

3.6.1 Inside PIC16F84A

3.6.1.1 Flash Program Memory

Flash memory is used to store the program. One word is 14 bits long and 1024 words (1k words) can be stored. Even if power is switched off the contents of the flash memory will not be lost. Flash memory can be written to using the writer, but the number of times it be rewritten is limited to 1000 times.

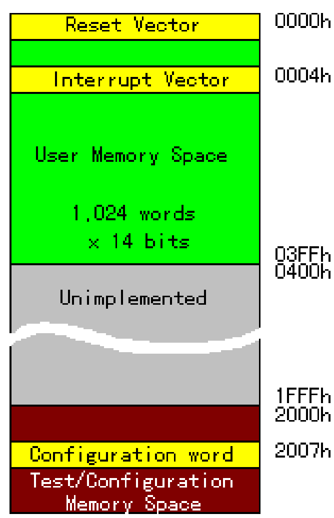

Figure 3.6 Program Flash Memory

The usage of some program memory addresses is already decided.

i. Reset Vector (0000h)

When a reset is executed, either by turning power on, by the WDT (Watchdog Timer) or any other factor, the program will start from this address.

ii. Peripheral Interrupt Vector (0004h)

When there is a time-out interruption from the timer (TMR0) or an outside interrupt, the program will start from this address.

iii. Configuration word (2007h)

The basic operation of the PIC is specified at this memory location. The enable bits of the Power-up timer, and the Watch-dog timer as well as the oscillator selection bits are set here. This area is behind the usual program area and can not be accessed by the program. These parameters must be specified using the burner when burning the program into flash memory.

3.6.1.2 SFR

16 different SFR (Special Function Registers) can be specified by the bank switching technique. The figure below shows the RAM File Registers. The memory capacity is only 160 bytes. The contents of the registers with the left pointing arrow are the same on both banks. The other registers of the SFR are accessible through bank switching, and the gray colored registers are not used.