Motor Controller (Requirement)

1. Objective

In this project, you get to design a motor controller to drive a DC motor or a stepper motor. The controller will base on a microcontroller, in this project, an 8051 MCU will be applied.

2. Introduction

2.1 Basic knowledge of hardware

2.1.1 Motor

To start, you need to read some materials about the motor, which you can find in https://en.wikipedia.org/wiki/Electric_motor.

Please focus on the part of the **DC motor** and **stepper motor** and then make sure you have understood them since you need to use them in your project. The figure 1 and figure 2 show the motors you will use in this project.

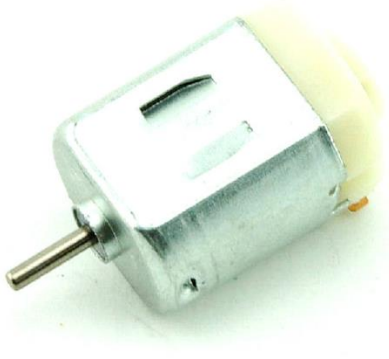

Figure1. DC Motor

Figure2. Stepper Motor

2.1.2 Microcontroller

A microcontroller is, in short, an economical computer-on-a-chip built for dealing with specific tasks, such as displaying or receiving information through LEDs or remote controlled devices.

Note: You can also watch the video to acquire the basic knowledge of microcontroller.

The most commonly used set of microcontrollers belong to **8051** family. You can use any kind of controller but 8051 family is recommended since it is cheap and easy to use. **8051 Microcontrollers** continue to remain a preferred choice for a vast community of hobbyists and professionals. Through **8051**, the world became witness to the most revolutionary set of microcontrollers. Figure 3 shows some chips belong to 8051 family.

Figure 3. 8051 Chips

8051 Family

Intel fabricated the original 8051 which is known as MCS-51. The other two members of the 8051 family are:

i. 8052 – This microcontroller has 256 bytes of RAM and 3 timers. In addition to the standard features of 8051, this microcontroller has an added 128 bytes of RAM and timer. It has 8K bytes of on chip program ROM. The programs written for projects using 8051 microcontroller can be used to run on the projects using 8052 microcontroller as 8051 is a subset of 8052.

ii. 8031 – This microcontroller has all the features of 8051 except for it to be ROM-less. An external ROM that can be as large as 64 K bytes should be programmed and added to this chip for execution. The disadvantage of adding external ROM is that 2 ports (out of the 4 ports) are used. Hence, only 2 ports are left for I/O(Input/Output) operations which can also be added externally if required for execution.

Features	**8051**	**8052**	**8031**
RAM(Bytes)	128	256	128
ROM	4K	8K	0K
Timers	2	3	2
Serial Ports	1	1	1
I/O ports	32	32	32
Interrupt sources	6	8	6

Table1. Comparison of 8051 family members

Features of 8051

The main features of 8051 microcontroller are:

i. RAM – 128 Bytes (Data memory)
ii. ROM – 4Kbytes (ROM signify the on – chip program space)
iii. Serial_port– Using UART makes it simpler to interface for serial communication.
iv. Two 16 bit Timer/ Counter
v. Input/output Pins – 4 Ports of 8 bits each on a single chip.
vi. 6 interrupt Sources
vii. 8 – bit ALU(Arithmetic Logic Unit)

viii. Harvard_architecture– It has 16 bit Address_bus(each of RAM and ROM) and 8 bit Data Bus
ix. 8051 can execute 1 million one-cycle instructions per second with a clock frequency of 12MHz.

This microcontroller is also called as "System on a chip" because it has all the features on a single chip. The Block Diagram of 8051 Microcontroller is as shown in Figure 4.

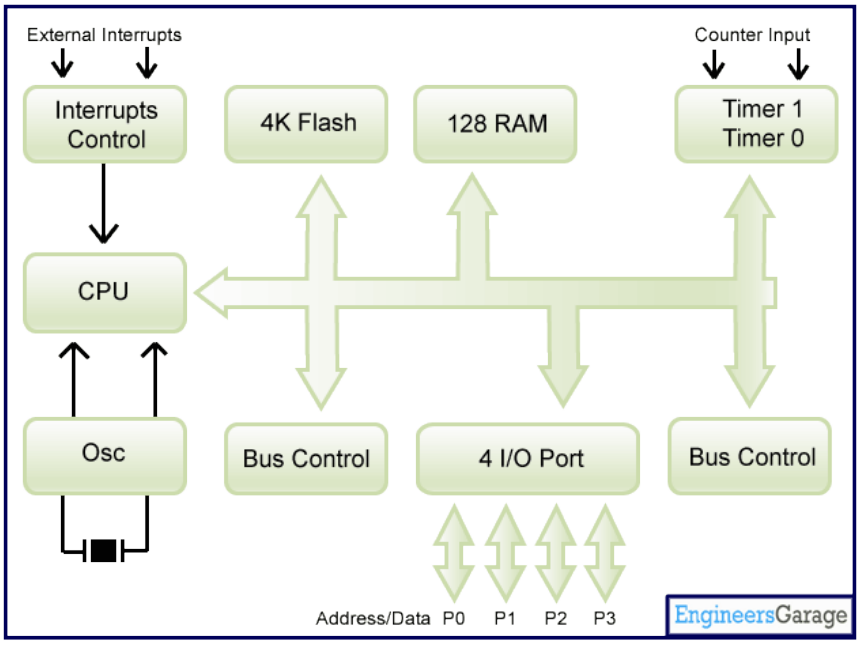

Figure 4. Block Diagram of 8051 Microcontroller

Applications

The 8051 has been in use in a wide number of devices, mainly because it is easy to integrate into a project or build a device around. The following are the main areas of focus:

i. Energy Management: Efficient metering systems help in controlling energy usage in homes and industrial applications. These metering systems are made capable by incorporating microcontrollers.
ii. Touch screens: A high number of microcontroller providers incorporate touch-sensing capabilities in their designs. Portable electronics such as cell phones, media players and gaming devices are examples of microcontroller-based touch screens.
iii. Automobiles: The 8051 finds wide acceptance in providing automobile solutions. They are widely used in hybrid vehicles to manage engine variants. Additionally, functions such as cruise control and anti-brake system have been made more efficient with the use of microcontrollers.
iv. Medical Devices:Portable medical devices such as blood pressure and glucose monitors use microcontrollers will to display data, thus providing higher reliability in providing medical results.

2.2 Basic knowledge of software

Programming environment and programmer

Formerly, programmers used machine language for coding. A machine language is a program that consists of 0s and 1s which was very dreary for the humans to any computer.

In due course of time, assembly language was developed in order to speed up the programming and make it error-free. Assembly language is a low level language which uses an assembler to translate the program into machine code. The high level programming languages such as BASIC, Pascal, Forth, C, C++, and Java are available to code the program for 8051. These high level languages make use of a Compiler to translate into machine code. For example, when a program is written in C, the program needs to be translated into machine language using C compiler. Usually, Assembly and C language is widely used for 8051 programs as compared to the other high level languages.

The 8051 provides a total of four ports for I/O operations. 8051 has 40 pins, of which 32 pins are set aside for the four ports. P0, P1, P2, and P3 each have 8 pins and can be used for either input or output. The remaining pins are designated as VCC, GND, XTAL1, XTAL2, RST, EA, ALE/PROG and PSEN.(**You can get information about the definition of these pins in the datasheet.**)

8051 allows you to manipulate one or all of the bits of a port, thus providing programmers with a unique and powerful feature. 8051 provides the programmer with the ability to read, write and modify each port to customize applications as much as possible.

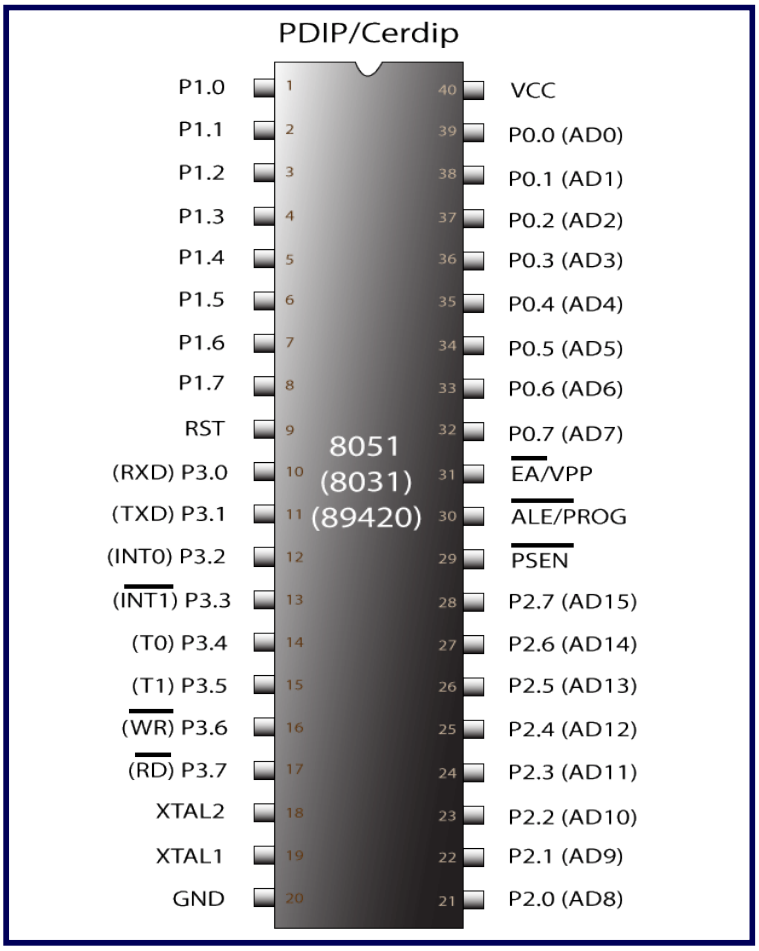

Figure 5. Pin Functions

C programming language is recommended in this project. Since all of you have used it, the basic knowledge of C programming language will not be covered in this guide.

In this project, the tool you need to work with is Keil. You can fill in your blanks and then get the link to download (**video for reference**). You can get the details of creating a new project and then compile it here and from the reference video.

Note: You can use either assembly language or C language to complete this project.

Once you have generated the .hex file, the next thing is to burn it into your microcontroller and restart your system to check whether your code is correct.

3. System block diagram (Minimum system)

The demo project is quite simple as you can see below:

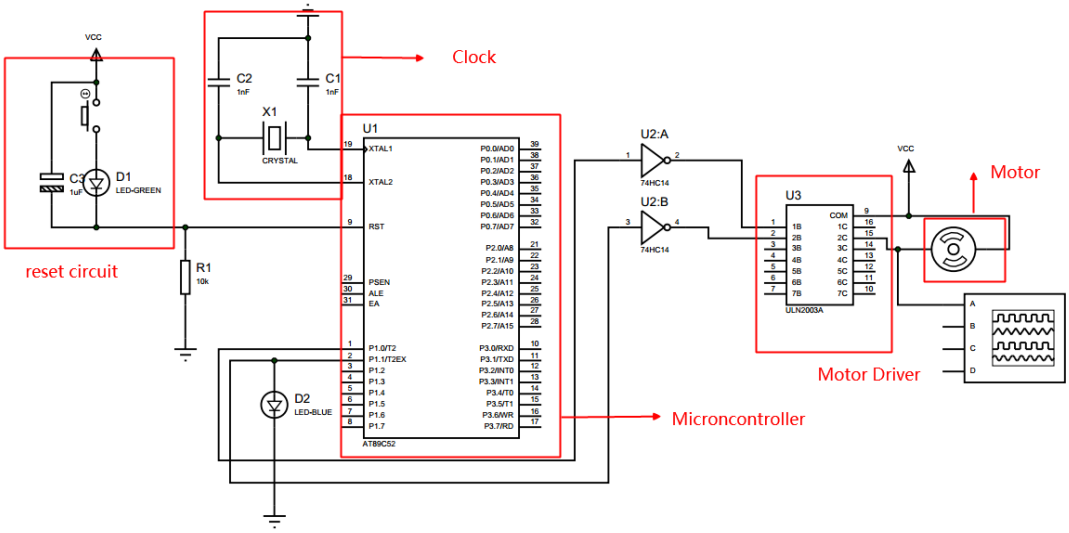

Figure 6. Demo minimum system

3.1 reset circuit

A reset circuit is the part of a circuit that restart the system. The detail of a reset part is shown below. The component in the red box is a button. If you push it, the voltage at RST is high and the system will be reset (Details in the datasheet).

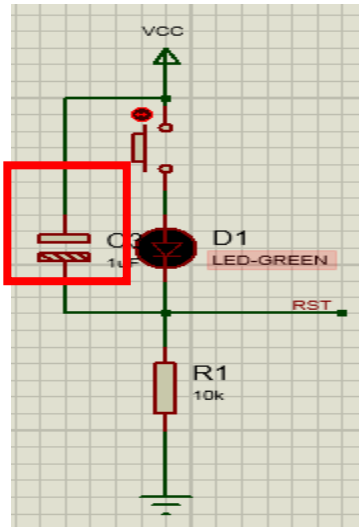

Figure 7. Reset circuit

3.2 clock circuit

The 8051 has on chip oscillator, pin XTAL1 and XTAL2 are provided for connecting a resonant network to from an oscillator crystal frequency ranges from 1 MHz to 24 MHz.

Ceramic resonators may be used as a low-cost alternative to crystal resonators but due to decrease in frequency stability and accuracy, ceramic resonators are not preferred for high-speed serial data

communication with other system.

The oscillator formed by the crystal, capacitors, and on chip inverter generates pulse train at the frequency of the crystal as shown in fig 8.

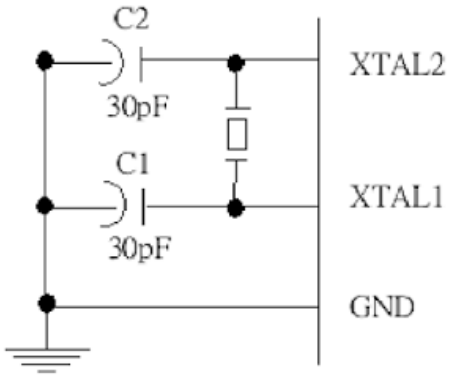

Figure 8. Clock circuit

3.3 Motor driver

Since the output current of microcontroller is not big enough to make the motor work, the motor driver is needed to provide a stronger current. You could not care about the reason. If you are interested in it, find some information in the Internet. ULN2003A is used as an example. **Of course, you can choose any other suitable chip to do the same thing**. Fig 9 is a simplified block diagram.

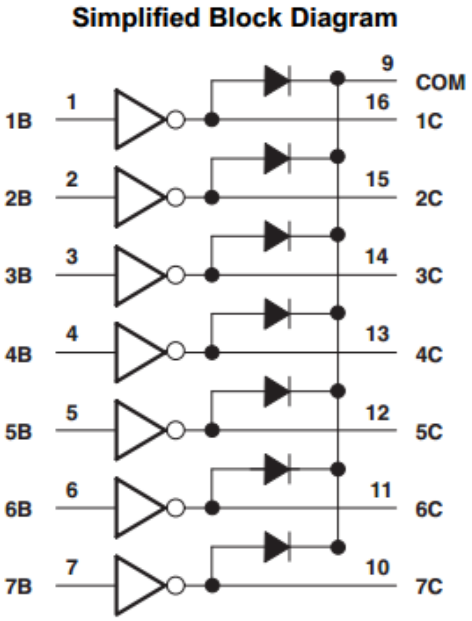

Figure 9. Simplified block diagram

3.4 Microcontroller & motor

The introduction to microcontroller and motor are in the beginning of this guide. In the demo system, the microcontroller is STC89C52RC or AT89C52RC and its frequency is 11.0592MHz.

3.5 Demo software

3.5.1 DC motor

You are supposed to have acquired the basic knowledge about timer and interrupt. In the demo project, both of them are applied to achieve our goal. The codes are included in the file DC_Motor.c. Of course, you could do the same thing in any way.

3.5.2 Stepper Motor

To do this part, you need to know how a stepper motor works. If you are not clear about this, go back to the first page and read the material carefully.

4. Build your circuit

4.1 First build your circuit and remember to be clean and neat.
4.2 Write the code and compile, link and debug to make sure that your codes work as which expected.
4.3 Burn the hex file by STC_ISP and then restart your system to check whether it works correctly.

5. Some notes

5.1 Minimum requirement

The **minimum requirement** is to drive a DC motor or a stepper motor. For DC motor, it must can speed up and then speed down. As for stepper motor, it must rotate either clockwise or anticlockwise. Finish this, you will get pass your project section in the course. If you want to get higher scores, you need to do some more work. Some options have been provided below.

5.2 Options

Any one of the followings is a plus:
(1) The revolving speed can be show with nixie tubes.
(2) The revolving speed can be show in a LCD.
(3) The DC motor could rotate clockwise or anticlockwise.
(4) You can assemble your motor and controller into a toy car. To do this, you need to buy a chassis.
(5) You can design your PCB if you want.
(6) Some other functions which could demonstrate your hard work.