Chapter 3

HARDWARE AND SOFTWARE DETAILS OF THE EMBEDDED WEB CAMERA CONTROL

ABSTRACT

Microcontrollers are becoming an integral part of Engineering design process known as Embedded system, the design includes major portion of the Hardware and entire Software incorporated on a single chip.

The present work describes the design of an Embedded system for the control of web camera. Software is developed in Assembly language. The software in the form of Hex file is dumped on the microcontroller AT89C51 using the PC. This chapter also describes the development of hardware for the above work and the results are discussed at the end.
INTRODUCTION

ATMEL Microcontrollers play a useful role in the scenario of embedded information processing. Keeping this in viewpoint the present work is centered around ATMEL89C51 microcontroller. An inexpensive web camera, namely, IBM PC camera is used in the present work. It can be connected to the PC via USB port. The hardware and software descriptions of the embedded camera control is as follows.

3.1. Hardware description

The web camera used in the present work is shown in Figure 3.1 [1]. As can be seen from the figure, the web camera has to be just placed in one place after connecting it to the USB port of PC. If the picture situated in other places has to be captured the web camera has to be rotated manually. This is an ugly situation during the programs such as video conferencing.

To avoid such a situation, in the present work, hardware is developed to rotate the web camera through 360° in both X and Y directions. These rotations are accomplished using two stepper motors. One of the stepper motor control the movement of the web camera in X-direction whereas the other will help the web camera to rotate in Y-direction.

A detailed discussion on different stepper motors is given in section 2.3 of Chapter 2. In the present work, two stepper motors removed from discarded 5 1/4 inch floppy drive mechanisms are used such stepper motors are also available commercially. The mechanical jig is shown in Figure 3.2. To obtain smooth
Fig. 3.1. Photograph of the web camera (IBM)
Fig. 3.2. Photograph of the mechanical jig arrangement
movement of the web camera the plastic cover on the web camera’s connector is removed neatly. This will help the movement of the camera without loosing the steps.

As a matter of fact, the stepper motors can be controlled by the PC itself but the burden on PC becomes more besides capturing of the image. To release the PC from such a burden a dedicated embedded system is developed which controls the rotation of the web camera. The rotation is accomplished by a group of four switches which are loosely coupled to the hardware. This will facilitate the user to control the rotation of the web camera from a distance.

The block diagram of the present work is shown in Figure 3.3. It essentially consists of the Atmel’s microcontroller 89C51(U1), two numbers of Darlington driver ICs ULN2003 (U2 and U3), web camera mounted on two stepper motors (SM1 and SM2) and PC. The pin details and the description of AT89C51 are already explained in Chapter 2. As the output port of the microcontroller can’t drive the stepper motors directly, Darlington arrays in the form of ICs (ULN 2003) are used, one each for stepper motor. Software to rotate the stepper motor using 8051 microcontroller was already developed in the laboratory by Thimmaiah, et al. [2], Laxmaiah et al. [3] in both Assembly and C languages. These papers are stripped to the dissertation. The data sheets of both the microcontroller as well as driver IC are given in Appendix A. As a simple example exercise, the circuit diagram to interface the stepper motor with microcontroller kit is shown in Figure 3.4.

The Hardware of the complete embedded web camera controller circuit developed in the present work and its photograph are shown in Figure 3.5. It consists of two Darlington arrays in the form of ICs (ULN 2003) connected between output
Fig. 3.3. Block diagram of the Embedded web camera control

- Web cam
- Computer
- SM1, SM2 - Stepper motors
- AT89C51 microcontroller
- U1, U2, U3 = ULN 2003
- SM1, SM2 - Stepper motors
A,B,C,D – Stepper motor windings

Figure 3.4. Interfacing the stepper motor with microcontroller trainer kit
Fig. 3.5 (a). Embedded control hardware circuit

Component List

- U1
- U2, U3
- C1, C3
- C2
- C4
- R1, R2,
- R3, R4
- R5
- A, B, C, D
- Push Buttons
- SM1, SM2
- Stepper Motors

+5V
AT89C51
ULN2003
10 nF
30 pF
10 pF
10 KΩ
SK2
Push Buttons
Stepper Motors

U1
U2, U3
C1, C3
C2
C4
R1, R2,
R3, R4
R5
A, B, C, D
SM1, SM2
Stepper Motors

+5V

Fig. 3.5 (a). Embedded control hardware circuit
Fig. 3.5 (b). Photograph of the hardware circuit used in the present work.
port of the microcontroller and stepper motors (SM1 and SM2). Before the insertion of the microcontroller in the embedded hardware, it has to be programmed. This programming is accomplished by using one of the commercially available programmers.

3.2 Software description

The flow chart to control the rotation of the web camera using the designed hardware is shown in Figure 3.6. Program is developed using the Assembler ASM51. The program listing is given in Table 3.1. The developed program is loaded on to the 89C51 using the programmer, whose description is given in the following section.

3.3 Programming the microcontroller AT89C51

The programmer for ATMEL 89C51/52/55 available from the website (kswichit@kmitl.ac.th) is directly downloaded on a floppy and is used in the present work. The hardware required to program the AT89C51 used in the present work is shown in Figure 3.7. The same hardware is connected to one of the COM PORTs of the PC (COM 1). The assembly language program is developed using DOS editor and converted to Hex file using the EZ Command software. Finally the Hex file is loaded on to the microcontroller AT89C51 through the serial port (COM 1) using following commands.

C:\atp> asm51 web_cam.asm
C:\atp> oh51 web_cam.obj
C:\atp> ez311

The command windows are shown in Figure 3.8. After successful programming the microcontroller is removed from the ZIF socket and placed in the embedded hardware circuit.
Initialize P1, P2 as output ports and P3 as input port

Initialize the register bank and other registers

Input the status of the keyboard from Port 3

Is Push button A pressed?
  No
  Out data to Port 1 to rotate SM1 in C.W. direction
  Call the DBDLY and Store the Status

Yes
  Is Push button B pressed?
    No
    Out data to Port 1 to rotate SM1 in A.C.W. direction
    Call the DBDLY and Store the Status
    Is Push button C pressed?
      No
      Out data to Port 2 to rotate SM1 in C.W. direction
      Call the DBDLY and Store the Status
      Is Push button D pressed?
        No
        Out data to Port 2 to rotate SM1 in A.C.W. direction
        Call the DBDLY and Store the Status
        Yes

Figure 3.6. Flowchart of the program
ORG 0000H
JMP MainLine

ORG 0100H

MainLine: MOV R1,#88H ; Load step sequence data in R1 for SM1
MOV R2,#88H ; Load step sequence data in R2 for SM2
LOOP_1: MOV A,#OFFH ; Configure PORT3 as input port
MOV P3,A
MOV A,P3
RRC A ; Rotate the contents of ACCUMULATOR
MOV R0,A ; Save the contents of ACCUMULATOR
JNC CLK_SM1 ; Check whether key A is pressed

KEY_B: MOV A,R0
RRC A ; Rotate the contents of ACCUMULATOR
MOV R0,A
JNC ACLK_SM1 ; Check whether key B is pressed

KEY_C: MOV A,R0
RRC A ; Rotate the contents of ACCUMULATOR
MOV R0,A ; Save contents of ACCUMULATOR
JNC CLK_SM2 ; Check whether key C is pressed

KEY_D: MOV A,R0
RRC A ; Rotate the contents of ACCUMULATOR
MOV R0,A
JNC ACLK_SM2 ; Check whether key D is pressed

LOOP_2: AJMP LOOP_1 ; Go to Check the pressing of a key

CLK_SM1: MOV A,R1 ; Transfer SM1 step sequence to ACCUMULATOR
RR A ; Rotate the contents of ACCUMULATOR right
MOV P1,A ; Output the data to PORT 1
MOV R1,A ; Save the sequence
ACALL DELAY ; Call delay subroutine
JMP KEY_B ; Loop back to check key B

ACLK_SM1: MOV A,R1 ; Transfer SM1 step sequence to ACCUMULATOR
RL A ; Rotate the ACCUMULATOR contents left
MOV P1,A ; Output data to PORT 1
MOV R1,A
ACALL DELAY ; Call delay subroutine
JMP KEY_C ; Loop back to check key C

CLK_SM2: MOV A,R2 ; Transfer SM2 step sequence to ACCUMULATOR
RR A ; Rotate ACCUMULATOR contents right
MOV P2,A ; Output data to PORT 2
MOV R2,A ; Save the sequence
ACALL DELAY ; Call delay subroutine
JMP KEY_D ; Loop back to check key D
ACLK_SM2: MOV A, R2 ; Transfer SM2 step sequence to ACCUMULATOR
RL A ; Rotate ACCUMULATOR contents left
MOV P2, A ; Output data to PORT 2
MOV R2, A
ACALL DELAY
JMP LOOP_2 ; Loop back to the end of the program

DELAY: MOV R3, #50H ; Load R3 with delay counter
BACK: DJNZ R3, BACK ; Decrement R3 till it attains zero value
RET ; Return from the subroutine
END ; End of mainline
Figure 3.7. Circuit diagram of Easy-Downloader
Figure 3.8. Command windows during 89C51 programming
3.4 Testing

The circuit shown in Figure 3.5 is rigged up on a general purpose PCB and the appropriate ICs are plugged in their respective sockets. The IC U1 is programmed by the data (Table 3.1) before insertion of the IC in the socket. The IC is programmed using the programmer, as described in the earlier section.

The embedded control board is powered up with a 5V power supply constructed in the laboratory (Figure 3.9).

The web camera is positioned conveniently and the driver software is run on the PC (web cam). By using the keys A, B, C, D the web cam is rotated at different angles, in different directions and the images are observed successfully. The photographs of embedded hardware, web camera, programmer and PC are shown in Figure 3.10.

3.5 Results and Discussion

The images scanned by the web camera inside the lab are photographed, and are shown in Figure 3.11.

In the present work the software is developed using ASM51. At present there are a good number of C compilers available on the market which supports software development for microcontrollers e.g., Reads51 from Rigel Corporation (www.rigelcorp.com), Keil software (Keil µvision 2). Work is in progress to develop the program in C and to control the rotation of the camera using IR remote controller.
Figure 3.9. 5V power supply to the system
Fig. 3.10. Photograph of the web camera connected to the mechanical jig and interface to the PC through USB port
Fig. 3.11. Images scanned by the web camera
References

1. IBM PC web camera.
