1. INTRODUCTION

Embedded systems are getting more and more popular with their integration in large number of applications. Broadly speaking today every application has got at least one embedded component integrated inside it, e.g., consumer electronics, automobiles, industrial automation, networking, etc. Nowadays they are also being devised as web application servers. The major benefits of embedding the web server are improved reliability, performance, efficiency, scalability and low cost. Such systems can be found very useful in remote monitoring applications where direct human intervention is not possible or at least not necessary.

Many such applications are being developed, for example Can Filibeli et al. [1] have described use of embedded web server for controlling home appliances remotely. Elkeelany et al. [2] have worked on the design of low cost, low power embedded device which eliminates the requirement of OS level processing of network protocol stack.

Manfred Bathelt et al. [3] have extended the design of web server and implemented device control using web browser.

Takayuki Fujita et al. [4] have reported work on an environmental sensing system in which monolithically integrated multi sensor chip with on chip peripheral have been used to store data in temporary RAM and to send it to TCP/IP protocol on Ethernet.

Farah Magrabi et al. [5] have described a web based approach for monitoring patients’ electrocardiogram. The server used in system provides database facility to store information of all patients and an access to the patient’s data and ECG has been provided for clients’ node.

This paper describes design of an embedded web server for weather parameter monitoring, which is devised around microcontroller P89V51RD2 interfaced with the Ethernet card RTL8019. In the proposed system the temperature and humidity sensors along with the signal conditioning circuitry form an input part. When powered the system initializes the TCP/IP link, and waits for the HTTP request. On arrival of request from the remote node the Embedded Web Server takes snapshot of current ambient parameters and puts it into the preformatted web-page. This web page is then sent to the requester node. All the TCP/IP binding is catered by the Ethernet controller under the control of firmware of the system.

2. SYSTEM BLOCK DIAGRAM

The system block diagram is shown in Fig. 1. It consists of temperature and humidity sensors, analog-to-digital converter, a microcontroller for controlling system’s activities, a LCD for local real-time display of parameters, and an Ethernet interface for network connectivity. The individual blocks are described below.

The sensor unit monitors temperature and atmospheric humidity with the help of LM35 semiconductor temperature sensor and SY-HS-220 humidity module, respectively. The LM35 precision temperature sensor produces calibrated and linear output in the range of –55 to +150°C. Similarly the calibrated humidity module SY-HS-220 minimizes the system complexity by reducing component count.
The internally compensated op-amp LM358 with low input offset voltage is used for temperature signal conditioning. The humidity sensor module does not require separate signal conditioning circuitry.

The A/D converter used in this system is an 8-bit, multi-channel, microprocessor compatible ADC0809. The system has been designed to monitor the temperature range of 0–50°C with the resolution of 0.5°C and the humidity range of 30–90% RH with 5% RH resolution. This reflects that the 8-bit resolution is sufficient for the system.

Philips P89V51RD2BN microcontroller is used due to its larger RAM and Flash-ROM capacity of 1 and 64 kB, respectively, which is adequate for small applications. It facilitates rapid application development with the on-chip In-System Programming facility.

The Ethernet module used in the system is REALTEK'S RTL8019. It supports full duplex Ethernet functionality and is compatible with the µIP, a freeware TCP/IP stack.

The system uses 16 × 2 LCD module for local real-time display. The module has on-board display controller which relieves the main microcontroller from manually generating dot-matrix character display. The detailed schematic diagram of the system is shown in Fig. 2.

3. SOFTWARE DESCRIPTION

Figure 3 shows the flow-chart of the system software. The system software is mainly composed of a TCP/IP stack, a BSP (Board Support Package) and application software. The programmed behavior of the system is as follows.

When the system is powered-up the initialization part of the system software configures various on-chip peripherals such as timers, interrupts etc. It also initializes the external interfaces viz. Ethernet, LCD, ADC, etc. Once the system hardware is initialized, the system loads the TCP/IP stack and enables the network communication via Ethernet. The HTTP server application is started and the web server can now be reached from the remote node. At this point the system keeps polling for the HTTP request.

When the HTTP request arrives from any of the remote hosts, the Embedded Web Server responds it with the acknowledgement, and checks for the authentication. The requests with proper authentication are accepted by the Embedded Web Server, in response of which it reads the ambient parameters, embed them in the preformatted web-page and sends the HTML web-page to the requestor node.

In addition to this main functionality, a timer driven ISR runs separately in the background to read ambient parameters and refresh the local LCD display.

Some of the major software routines are described below.

3.1. HTTP.C

This file consists of routines which enables server to listen request, to respond and process the request, to format the web page and to send the web page to the node which made the request.
Fig. 2. Schematic of system. (IC$_1$) ADC0808D, (IC$_2$) P89c51RD2BN, (IC$_3$) IC7805, (IC$_4$) MAX3221EAE, (IC$_5$) LM358D, (D$_1$) U57X32, (D$_2$) 1N007.

![Flow chart diagram](image_url)

Fig. 3. Flow chart.
3.2. TCP_IP.C

This file consists of routines to initialize the TCP protocol layer of stack, to process the TCP packets. The subroutines provided allow the microcontroller to initialize various parameters of the TCP/IP protocol, establish TCP connection with remote IP and port number, check for the timeout of the connection, indicate the arrival of TCP packets, and receive or send the data over the active connection, process the IP packets, and perform checksum.

3.3. Ethernet

This is a driver for Ethernet controller chip which maintains communication between node and the network. It provides the routines to transmit IP frame onto the network or vice versa.

3.4. HumConversion() and TempConversion()

These routines read the environmental data with the help of ADC and send it onto the network by embedding into the web page.

4. OBSERVATIONS

The data of temperature recorded using the “Embedded Web server” is represented graphically in Fig. 4, which shows close agreement between temperature parameter monitored by the Embedded Web Server and the one recorded on the calibrated thermometer.

Figure 5 gives variation of humidity as a function of time recorded using the Embedded Web Server system and using calibrated hygrometer.

From the graphs of Humidity and temperature it is clear that there is very close agreement between the data collected by our system and that measured by already available and calibrated systems which validates the measurements made by our system.

REFERENCES

DESIGN OF MICROCONTROLLER BASED DC PROGRAMMABLE POWER SUPPLY

P.A. Kadarm¹, G.G. Tengade² and S.R. Sawant³

¹Department of Electronics, Shivaji University, Kolhapur
²Department of Electronics, D.Y. Patil College of Engineering, Kolhapur

ABSTRACT

Some applications require switching between several different levels of voltage applied to the instrument. The switching needs to be reliable and proper; otherwise the possibility of device malfunctioning can not be ruled out. For these needs digital circuitry is more suitable when number of switching levels is large. The present circuit is designed around National Semiconductor's 8-bit D-A converter DAC0808 and voltage regulator IC LM723. The microcontroller ATMEGA169C2051 is used to control the output voltage. The circuit worked efficiently up to 1A load current. The voltage and line regulations are 0.6% and 0.2% respectively.

INTRODUCTION

DC Power supply is required to provide stable source of power to the electronic circuits and systems. These Power supplies are mainly categorized as follows¹:

1. Constant Voltage Power Supply (unipolar or bipolar):
   a. Voltage reference type
   b. High voltage low current type
   c. Low voltage high current type

2. Constant Current Supply:
   a. Current reference type
   b. High current low voltage type
   c. Low current high voltage type

3. Others:
   a. Constant voltage/constant current type
   b. Bipolar type
   c. Programmable type, with analog or digital control
Programmable power supplies differ from the other categories in the following points:

a. Output voltage can be time controlled with wide variations ranging from few millivolts/min to a few volts/sec as per requirements.

b. Ability to supply rated current at all these voltages.

c. Better repeatability of the output voltage than the analog control.

Such a programmable power supply can easily be built using D-A converter followed by a linear series pass regulator. This supply finds the applications as calibrator, working voltage standard, system reference supply or as laboratory power supply.

The digital control makes it ideal for following applications:

1. **Automatic Semiconductor Device Testing**:  
The supply can be used to set the bias conditions for the device being tested. These conditions need to be changed frequently and accurately. The digital control is of great advantage in such situations.

2. **Waveform Generation (Unipolar)**:  
The supply can provide very well controlled arbitrary waveforms e.g., 10,000 voltage changes/sec. These waveforms are highly useful in the testing of different systems.

3. **Process control**:  
The circuit can be interfaced to central process controller to control the peripheral devices such as motors, valves, etc. The power supply being digitally controlled is highly suited to this application.

**DESIGN PROCESS**:  
The major blocks of the system are shown in the fig. 1. Microcontroller feeds DAC with appropriate digital count to control the output voltage. The DAC converts this digital count into the equivalent analog voltage. The output of DAC forms the reference to the series pass voltage regulator, which drives the load.

![Fig. 1. System Block Diagram](image-url)
The microcontroller is chosen from the MC3S1 family because of ease of availability and low cost. The 8-bit DAC gives 256 different voltage levels. The series pass regulator with transistor current booster will provide load current up to 1A.

**Circuit description:**

The microcontroller AT89C2051 adds the intelligence to the system. Two switches have been employed one for step-up and other for step-down operations respectively. The inputs from the two switches after ANDing are applied to the interrupt input of the microcontroller. The DAC is interfaced to the port 1 of the microcontroller to receive digital count via port 1. The step voltage is selected as per the requirements. The DAC is current driven. It takes 5 volts as Vref through 2 KΩ resistance. The same resistance is used in the 1-to-V converter. Thus the output of 1-to-V ranges from 0 to 5 volts. (Actually the higher side is 1 LSB less than 5 volt). For the systems where the small variations in output voltage due to power supply and temperature variation are tolerable this setup is fine. For more stable design the reference current for DAC must be stable and be provided through the current source e.g. as shown in the following circuit which may also be derived from unregulated power supply.

![Fig. Stable Current source](image)

This circuit has the advantage of stable current operation, as the temperature coefficient of LED is close to that the emitter-base junction of the transistor. The additional benefit is that the sourcing current can easily be controlled varying R2.

The DAC output is applied to the ‘+in’ terminal of the internal comparator of voltage regulator IC LM723. The ‘-in’ terminal is fed with the Vref. Thus the control can be strained over output voltage change due to load current or fine voltage change. IC LM723 is capable of supplying maximum 150 mA output current. The current is further amplified with the help
Note: Tolerance for the resistances R1, R2, R3, R10 and R11 is 1% Max; for others 5% is permissible.
of NPN Power Transistor TIP3055. The IC also provides the Current Limit Protection with the help of current limiting resistance \( R_{\text{so}} \).

**Software**

Two routines have been designed and tested. One with 1 LSB change at the output voltage \( V_{\text{cm}} \), WHILE the other with 8*LSB change at the same. Listing 1 gives the program with 1 LSB change while listing 2 provides 8*LSB change from 00h to FFh.

**Listing 1:** Program for 1 LSB change at the output voltage.

```assembly
ORG 0
LJMP START
ORG 0050H

START:
  MOV PI, 0
  ORL TCON, #O1H
  MOV IE, #11H ; Enable INTO edge sensitive to read switches
  SJMP $5
  ORG 0003H ; ISR to read switches
  MOV B, P3
  JNB 0FH, DOWN
  JNB 0F1H, UP
  RETI

DOWN:
  MOV A, PI ; Voltage 1 step down
  CJNE A, #0, DECR
  RETI

DECR:
  DECA
  MOV PLA
  RETI

UP:
  MOV A, PI ; Voltage 1 step up
  CJNE A, #FFH, INC
  RETI
```

**Listing 2:** Program for 8*LSB change at the output voltage.

```assembly
ORG 0
LJMP START
ORG 0050H

START:
  MOV PI, 0
  ORL TCON, #O1H
  MOV IE, #11H ; Enable INTO edge sensitive to read switches
  SJMP $5
  ORG 0003H ; ISR to read switches
  MOV B, P3
  JNB 0FH, DOWN
  JNB 0F1H, UP
  RETI

DOWN:
  MOV A, PI ; Voltage 1 step down
  CJNE A, #0, DECR
  RETI

DECR:
  DECA
  MOV PLA
  RETI

UP:
  MOV A, PI ; Voltage 1 step up
  CJNE A, #FFH, INC
  RETI
```
INCR:
  INC A
  MOV PI, A
  RETI

Listing 2: Program for 6*LSB change at the output voltage.

  COR 0
  LJMP START

  COR 0050H

START:
  MOV PI, 0
  ORL TCON, #0H
  MOV IE, #01H ; Enable INTO edge sensitive
  SJMP $

  COR 0000H ; ISR to read switches

  MOV B, P3
  INB OF0H, DOWN
  INB OF1H, UP
  RETI

DOWN ; voltage 3 steps down
  MOV A, PI
  CINE A, #0, DECR
  RETI

DECR:
  SUBB A, #08H
  MOV PI, A
  RETI

UP ; voltage 3 steps up
  MOV A, PI
  CINE A, #0F8H, INCR
  RETI
INCR:
ADD A, #01H
MOV Pi, A
RETI

As in listing-1, the program enables the INT0 interrupt of the microcontroller. The ISR is written to read the switches in real time. Two switches have been provided, one for increasing the voltage \( V_{an} \) and the other for decreasing it. In this routine the digital count is increased by 1 LSB. Thus there are 256 steps to be followed from 00h to FFh. For safety the initial digital count is set to 00H with equivalent \( V_{an} \) (0 Volts).

The only change made in listing-2 is the digital count steps in multiples of 8 of LSB. Thus the output voltage \( V_{an} \) changes accordingly with less number of steps.

Observations:

Load Regulation:

\[
V_{an} = 12V
\]

\[
V_{an} \text{ of LM723} = 4.98V
\]

<table>
<thead>
<tr>
<th>Obs No.</th>
<th>( V_{an} ) (in Volts)</th>
<th>Load Resistance (in Ω)</th>
<th>Load Current (in mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>4.98</td>
<td>50</td>
<td>94</td>
</tr>
<tr>
<td>2.</td>
<td>4.98</td>
<td>40</td>
<td>116</td>
</tr>
<tr>
<td>3.</td>
<td>4.98</td>
<td>30</td>
<td>132</td>
</tr>
<tr>
<td>4.</td>
<td>4.98</td>
<td>20</td>
<td>221</td>
</tr>
<tr>
<td>5.</td>
<td>4.97</td>
<td>10</td>
<td>470</td>
</tr>
<tr>
<td>6.</td>
<td>4.96</td>
<td>5</td>
<td>890</td>
</tr>
<tr>
<td>7.</td>
<td>4.95</td>
<td>4</td>
<td>1090</td>
</tr>
<tr>
<td>8.</td>
<td>4.88</td>
<td>3</td>
<td>1360</td>
</tr>
<tr>
<td>9.</td>
<td>3.43</td>
<td>2</td>
<td>1360</td>
</tr>
</tbody>
</table>

From table 1 it is observed that,

For Load currents ranging from 94 mA to 1.09 Amp, load regulation is 0.6% of \( V_{an} \).

Line Regulation:

\[
V_{an} \text{ of LM723} = 4.98V
\]

\[ R_L = 50 \, \Omega \]

\[ I_L = 96.5 \, mA \]
Table 2. Variations in output voltage with line variations

<table>
<thead>
<tr>
<th>$V_{in}$ volts</th>
<th>$V_{out}$ volts</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4.98</td>
</tr>
<tr>
<td>11</td>
<td>4.98</td>
</tr>
<tr>
<td>10</td>
<td>4.98</td>
</tr>
<tr>
<td>9</td>
<td>4.98</td>
</tr>
<tr>
<td>8</td>
<td>4.98</td>
</tr>
<tr>
<td>7</td>
<td>4.97</td>
</tr>
</tbody>
</table>

From table 2 it is observed that, for $V_{in} = 7$ Volts to 12 Volts, line regulation of 0.2% of $V_{in}$ can be achieved. The voltage regulator IC LM723 requires that the supply $V_{in}$ should be greater than or equal to 8 volts.

4. CONCLUSION

The designed circuit is very useful in the applications where a multiple level step change in power supply voltage is required. The circuit can be operated successfully at load currents of up to about 1 Amp. A precise step increase/decrease in output voltage can be achieved. The simplicity of operation makes it useful alternative of analog power supplies where generally two different potentiometers are employed: one for general voltage adjustment and one for precise voltage adjustment. The microcontroller embedded in the system can add several facilities to it, viz. Programming of output voltage $V_{out}$ from PC via serial communication link, 7 segment/LCD display of $V_{in}$ and other parameters, etc. The internal precision comparator of microcontroller IC AT89C2051 and additional transistors can replace IC LM723. But this IC is better choice for faster designs, as it is readily available with good specification data.

This system can be modified for bipolar operation so that the output will be bipolar and also the current will be in both directions for either polarity of output.

REFERENCES

MOUSE SIGNAL INTERPRETER USING MICROCONTROLLER 89C2051

Department of Electronics, Shivaji University, Kolhapur - 416004

ABSTRACT

The mouse, a pointing device, is generally connected to the standard serial (COM1, COM2, etc.) port of PC. It is used to move the cursor around a CRT screen to make drawings or execute commands by selecting commands from menus.

During the mouse manufacture process the last and important stage is mouse testing. For this purpose a low cost embedded system has been designed around ATMEL 89c2051 microcontroller and tested. This system can be used to test all the functions of standard serial 3-button mouse. The system is found to be very useful in testing serial mouse.

1. INTRODUCTION

When a computer is used for the purpose of mouse testing, it needs to be turned off before connecting or disconnecting a mouse to it. Each time the computer needs to be rebooted to test the mouse using software programs. The PC will then be turned off before disconnecting the mouse. This process need be repeated for each mouse to be tested. This sequence in time consuming as every time the PC needs to be rebooted before executing mouse test software. An embedded system dedicated for the objective can come to the rescue here. Such an embedded system can be designed using Atmel’s AT89C2051 microcontroller and IC MAX232. The AT89C2051 is an economical and cost-effective member of Atmel’s growing family of microcontrollers. It contains 2K bytes of flash program memory and on-chip full duplex serial port. Thus it is a good solution for the development of small embedded-systems, as the one proposed in this work.

2. THEORETICAL

PC mouse information

Typical PC mouse controller system has following major components:

- Sensors
- Mouse Controller
- Communication Link
- Data Interface
- Driver
- Software

Opto-mechanical Sensors are generally used as movement detectors, which sense the mouse movement. The push-to-on switches are used to sense the mouse button status. Mouse
controller reads the state of sensors and takes an account of present mouse status. Mouse controller sends a packet of data to the computer for changes in information from its last status.

The mouse driver inside the computer receives data packets sent by the mouse, decodes the information and acts accordingly. A typical mouse driver keeps the information of present state of mouse (position and button status) and passes on this information to the operating system. The mouse driver calls mouse cursor movement routines when mouse is moved and sends messages to the software when buttons are pressed.

Serial Mouse Hardware

A serial mouse connects to an existing RS-232 serial port at the PC using a standard DB-9F (9-pin female) or DB-25F (25-pin female) connector. The pin out of these connectors is given in table 1.

<table>
<thead>
<tr>
<th>DB-9F 9 PIN</th>
<th>DB-25 25 PIN</th>
<th>WIRE NAME</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell</td>
<td>1</td>
<td>Protective Ground</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Receive data</td>
<td>Serial data from host to mouse</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Transmit data</td>
<td>Serial data from mouse to host (for power only)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>DSR</td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>Signal Ground</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>DTR</td>
<td>Data Terminal Ready</td>
</tr>
</tbody>
</table>

Serial port of the computer uses bipolar signaling which favours long cabling with minimum noise. A positive voltage between +3Vdc to +15Vdc represents a logic 0 (space) condition. A negative voltage between −3Vdc and −15Vdc represent logic 1 (mark) condition. These voltages are shown in figure 1.

<table>
<thead>
<tr>
<th>Logic 0</th>
<th>Start</th>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>+3...+15V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logic 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>−3...−15V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1: RS-232 frame format
Hardware Implementation

A PC serial mouse uses typically DTR and RTS lines for generating +5V power for internal microcontroller circuitry. Typical opto-mechanical mouse needs some power for driving LEDs used in the opto-coupler movement detectors. Diodes connected in the circuit take current from DTR and RTS lines and then feed this current through resistors to all of the (infrared) LEDs used in the movement detectors. All the four LED's are connected in series with approximately 5V drop across these LEDs (typical to LEDs used in mouse). This arrangement provides adequate power for low power mouse microcontroller. The serial data transmitting circuit consists of simple discrete transistor circuit, which consumes very little power.

The negative supply for mouse transmitter is taken from TD pin. Standard serial mouse take about 10 mA current and operates at voltage range of 6-15V. The data is sent using standard asynchronous RS-232C serial format.

For proper operation both the RTS and DTR lines must be held positive. The lines DTR-DSR and RTS-CTS must not be shorted. The RTS toggle function is implemented by setting the RTS line negative and positive again. The negative pulse width should be at least 100 ms. After a cold boot, the RTS line is usually set to a negative level. In this case, setting the RTS line to a positive level is also considered an RTS toggle.

Serial data parameters

1200bps, 1 start bit, 8 data bits, 1 stop-bit

Data packet format

Mouse data packet consists of 3-bytes. It is sent to the computer every time mouse state changes (mouse moves or keys are pressed/released).

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>1</td>
<td>LB</td>
<td>RB</td>
<td>Y7</td>
<td>Y6</td>
<td>X7</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>0</td>
<td>X5</td>
<td>X4</td>
<td>X3</td>
<td>X2</td>
<td>X1</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>0</td>
<td>Y5</td>
<td>Y4</td>
<td>Y3</td>
<td>Y2</td>
<td>Y1</td>
</tr>
</tbody>
</table>

The bits marked with x are 0 if the mouse uses 7 data bits and 2 stop bits format. It is also possible to use 8 data bits and 1 stop bit format for receiving. In this case x gets value 1.

The byte with D6=1 is sent first followed by the others. The bit D6 in the first byte is used for synchronizing the software to mouse packets if it goes out of sync.

LB is the state of the left button (1 = pressed down), RB is the state of the right button (1 = pressed down). The bits X7-X0 indicate movement in X direction since last packet (signed byte). Y7-Y0 indicates movement in Y direction. Fig. 2 gives the graphical description of how data packet is formatted.
3. EXPERIMENTAL

The system block diagram is shown in fig. 3. Serial mouse operates using the standard RS-232C signals. It needs to be interfaced to the microcontroller through the RS-232C line driver/receiver IC MAX232. Mouse electronics normally uses +/-5V, which is drawn from the IC MAX232, used in the system.

Fig. 3: Block Diagram of Mouse test system

Fig. 4 shows system circuit diagram. In this system the microcontroller AT89C2051 is used as an interpreter of mouse signals. The ease of availability and low unit cost of the microcontroller makes it a perfect choice for the system. It has a full duplex UART serial channel that supports different modes of communication. The 'Mode 1' of serial communication i.e. standard UART mode is used for receiving data packets from serial mouse. Received data is formatted as a start bit (which is always low), eight data bits (LSB first), and a stop bit (always high). The microcontroller can be programmed to receive/transmit data at any baud rate desired.
Fig. 4: Mouse 2nd System
Standard serial mouse transmits data at the baud rate of 1200 bps, thus the microcontroller is programmed to receive data packets at the same baud rate. This baud rate is one of the standard baud rates supported by IBM PC's COM port. The microcontroller clock is driven by the crystal of frequency 11.0592 MHz, as this frequency supports accurate baud rate calculation.

The mouse transmits unique data packets regarding its present state. The software loaded on to the microcontroller uses this information to indicate the present mouse status.

The display unit shown in fig. 5 consists of LED panel. It indicates the present status of mouse position and mouse buttons and thus provides the indication of any fault if present in the mouse.

![Diagram of mouse circuit](image)

**Fig. 5:** Display Unit

The proposed system is powered by a TTL compatible +5V @ 500mA regulated power supply. IC MAX232 is used to convert microcontroller's TTL compatible serial data levels into the RS232 standard data levels for mouse and back to TTL again. This IC also provides bipolar voltages required for supplying power to mouse circuitry.
The MCS51 family assembly language (ALP) is used to develop the software for the system. The software listing is given in Listing 1. Evaluation version of Pinemula's IDE software with 2KB limit is good enough for the small embedded-system design such as the one presented in this work. ALP is best suitable for small systems development where code size is limited. The major advantages of ALP are better memory/speed efficiency and better control over the program flow. Pinemula's IDE provide a complete development environment with built-in assembler and simulator facilities.

The program code will then be downloaded into the microcontroller's flash memory with the device programmer. The microcontroller chip will then added to the system to make it functional.

4. RESULTS AND DISCUSSIONS

The system is implemented and tested successfully. All of the serial mouse functions were successfully tested and displayed on LED display panel. The mouse readings taken by the microcontroller are observed on the LCD display used for testing purpose only. The observations are shown in Table 2.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Pressed</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Switch</td>
<td>DBH 80H 80H</td>
<td>CBH 80H 80H</td>
</tr>
<tr>
<td>Middle Switch</td>
<td>CBH 80H 80H A0H</td>
<td>CBH 80H 80H 80H</td>
</tr>
<tr>
<td>Left Switch</td>
<td>EBH 80H 80H</td>
<td>CBH 80H 80H</td>
</tr>
<tr>
<td>Wheel L-R</td>
<td>C3H BEH 80H</td>
<td>C3H BEH 80H</td>
</tr>
<tr>
<td>Wheel R-L</td>
<td>CBH 81H 80H</td>
<td>CBH 82H CBH</td>
</tr>
<tr>
<td>WHEEL B-T</td>
<td>CCH 80H BEH</td>
<td>CCH 80H BEH</td>
</tr>
<tr>
<td>WHEEL T-B</td>
<td>CBH 80H 81H</td>
<td>CBH 80H 82H</td>
</tr>
</tbody>
</table>

Note: The observations in Table 2 are taken with the help of 16*2 LCD display panel connected to the microcontroller system.

A simple embedded system can easily be designed to support Q/C tasks in different industries. The proposed test system for mouse is found to be effective solution for the problem. The system is portable as it can fit at matchbox size, excluding the power supply or in case of battery operated system. Three AAA size battery cells can be used to supply power to the system. The cost of the system is the added advantage. In case of large productions the system will cost below Rs. 100/-.

One of the limitations of this system is that it can not display the displacement in X and Y direction, as simple LED panel is used. But this can be easily replaced with 16*2 LCD display panel to indicate mouse displacement in either direction, with little increment to the costs.

The system can also be extended to accommodate other mouse types such as PS/2 mouse, scroll mouse, optical mouse, etc. The proposed system is designed and operated successfully.
Listing 1: Software for serial mouse test

1 0005: ;-------------------------------------------------------------
2 0009: ; SERIAL MOUSE TEST
3 000D: ;
4 000F: ; THIS PROGRAM TESTS THE SERIAL MOUSE
5 0013: ; AND DISPLAYS THE STATUS USING LED's
6 0017: ;
7 0019: ; MOUSE BAUD RATE: 1200
8 001D: ; STATUS LED's: PORT P1
9 0021: ;-------------------------------------------------------------
10 0025: ;STATUS LED CONNECTION
11 0029: ;
12 002D: ; R1 EQU P1.0
13 0031: ; DWN EQU P1.1
14 0035: ; UP EQU P1.2
15 0039: ; LO EQU P1.3
16 003D: ; LB EQU P1.4
17 0041: ; RB EQU P1.5
18 0045: ; RB EQU P1.6
19 0049: ;-------------------------------------------------------------
20 004D: ;MAIN PROCEDURE
21 0051: ;
22 0055: ORG 0000H
23 0059: MOV R0, #05H
24 005D: CLR TI
25 005F: ANL PCON, #7FH
26 0063: ANL TMOD, #03H
27 0067: ORL TMOD, #20H
28 006B: MOV TH1, #083H
29 006F: SETB TI
30 0073: MOV SCON, #0H
31 0077: SETB RI
32 007B: ACALL DEMO
33 007F: ;LED TEST
34 0083: 0019 DEMO:
35 0087: ACALL DEMO
36 008B: DUNH RC, REREO
37 008F: 010D AGAIN:
38 0093: JB C R1, REREO
39 0097: Sjmp AGAIN
40 009B: RC:
41 009F: MOV A, SBUF
42 00A3: CMND A, #30H, CHK MID ;IGNORp FIRST BYTE
43 00A7: JMP AGAIN
44 00AB: CHK MID:
45 00AF: CMND A, #0AH, FIRST BYTE
46 00B3: SETB 06H ;IF MIDDLE BUTTON PRESSED
47 00B7: LCALL OUTPUT
48 00B1: 0315 JMP AGAIN:
49 00B5: FIRST BYTE:
50 00B9: CMND A, #0BH
51 00BB: CLR 06H
52 00BC: MOV 2OH, A
53 00BD: JNE CML, AGAIN ;IF NOT FIRST BYTE GET EXT DATA
54 00BE: LCALL OUTPUT ;PROCEDURE TO DISPLAY MOUSE STATUS
55 00C1: 030D JMP AGAIN:
56 00C5: ;
57 00C9: 030F OUTPUT: ;PROCEDURE TO DISPLAY MOUSE STATUS
58 00CF: 7D D6 D5 D4 D3 D2 D1 D0
59 00D4: 03F ;1 1 0 0 0 0 0 0
60 00D9: 03F ;
61 00DA: 03F OUTPUT:
62 00DE: JB 05H, LASET ;LEFT BUTTON PRESSED
63 00EF: Sjmp LB_RESET ;LEFT BUTTON RELEASED
64 00F0: LASET:
65 00F1: SETB LB
66 00F5: Sjmp NXT1
67 00F7: LB_RESET:
68 00F8: CLR LB
69 00FA: NXT1:
70 00FB: JB 04H, RB_SET ;RIGHT BUTTON PRESSED
71 00FD: Sjmp RB_RESET ;RIGHT BUTTON RELEASED
72 00FE: RB_SET:
73 00FF: SETB RB
74 0100: Sjmp NXT2
Mouse Signal Interpreter using Microcontroller 89C2051

78 0553 RD_RESET:
79 0553  CLR RB
80 0555 NXT2:
81 0555  JB 03H, UPD  ;MOVE TO UP DIRECTION
82 055B  CMP BOTTOM  ;MOVE TO BOTTOM DIRECTION
83 055A UPD:
84 055A  SETB UP
85 055C  CLR DOWN
86 055A  CMP NXT3
87 0560 BOTTOM:
88 0560  CLR UP
89 056A  SETB DOWN
90 0568 NXT3:
91 0564  JB 01H, L_R  ;MOVE FROM LEFT TO RIGHT
92 0567  SJMP R_L  ;MOVE FROM RIGHT TO LEFT
93 0569 L_R:
94 0569  SETB RD
95 0569  CLR LD
96 0568  SJMP NXT4
97 056F R_L:
98 056F  SETB LD
99 0571  CLR RD
100 0573 NXT4:
101 0573  JB 08H, MB_SET  ;MIDDLE BUTTON PRESSED
102 0576  SJMP MB_RESET  ;MIDDLE BUTTON RELEASED
103 0573 MB_SET:
104 0578  SETB MB
105 057A  CMP END
106 057C MB_RESET:
107 057C  CLR MB
108 057F  END:
109 057F  RET
110 057F  ;PROCEDURE FOR LED TEST
111 057F  :
112 057F  DEMO:
113 057F  MOV F1, #0FFH
114 0582  ACALL DLVO1
115 0584  MOV F1, #0
116 0587  ACALL DLVO1
117 0589  RET
118 058A DLVO1:
119 058A  MOV R7, #04
120 058C  MOV R6, #0
121 058E  MOV R5, #0
122 0590  HERR01:
123 0590  DJNZ R5, S
124 0592  DJNZ R6, HERR01
125 0594  DJNZ R7, HERR01
126 0596  RET
127 0597  ;PROCEDURE FOR LED TEST
128 0597  :
129 0597  DEMO1:
130 0597  MOV F1, #0
131 059A  ACALL DLVO1
132 059C  SETB RB
133 059E  ACALL DLVO1
134 05A0  SETB MB
135 05A2  ACALL DLVO1
136 05A4  SETB LB
137 05A6  ACALL DLVO1
138 05A8  SETB LB
139 05AA  ACALL DLVO1
140 05AC  SETB UP
141 05AE  SETB Down
142 05B0  ACALL DLVO1
143 05B2  SETB RD
144 05B4  ACALL DLVO1
145 05B6  RET
146 05B7  ;TRAN LOCATIN 20H = SMTP DATA
POSITIVE
POSITIVE
to test single mouse at a time. But the system can be extended with the help of multiplexer and little addition to the software, for multiple mice testing for industrial application.

5. CONCLUSION

This is an inexpensive, portable and novel design for mouse testing with flexibility of compatibility with other mouse technologies. Looking at these factors the above mentioned system will be preferred over the conventional PC based test system.

REFERENCES