FIRMWARE ASSIGNMENT

1)The PC transmits 1000 bytes of text to the 8051 microcontroller.

2)The 8051 microcontroller receives the data and sends it back to the PC, displaying the real-time data transmission speed on the LCD.

#include <reg51.h>

#include <stdio.h>

#define LCD\_DATA P2 // 8-bit data bus connected to P2

#define RS P3.0 // Register select

#define RW P3.1 // Read/Write

#define EN P3.2 // Enable

#define BAUD\_RATE 2400 // Baud rate for UART

// Given text data (1000 bytes)

const char text\_data[] =

"Finance Minister Arun Jaitley Tuesday hit out at former RBI governor Raghuram Rajan for predicting that the next banking crisis would be triggered by MSME lending, saying postmortem is easier than taking action when it was required. Rajan, who had as the chief economist at IMF warned of impending financial crisis of 2008, in a note to a parliamentary committee warned against ambitious credit targets and loan waivers, saying that they could be the sources of next banking crisis. Government should focus on sources of the next crisis, not just the last one";

#define BUFFER\_SIZE 1000

char data\_buffer[BUFFER\_SIZE];

unsigned int buffer\_index = 0;

unsigned long timer\_counter = 0;

void Timer0\_ISR(void) interrupt 1

{

timer\_counter++; // Increment timer\_counter every 1 ms

}

void delay(unsigned int count)

{

while (count--);

}

void LCD\_Command(char cmd)

{

LCD\_DATA = cmd; // Send command to LCD

RS = 0; // Register select -> Command Register

RW= 0; // Write mode

EN = 1; // Enable

delay(100);

EN = 0; // Disable

}

void LCD\_Char(char data)

{

LCD\_DATA = data; // Send data to LCD

RS = 1; // Register select -> Data Register

RW = 0; // Write mode

EN = 1; // Enable

delay(100);

EN = 0; // Disable

}

void LCD\_String(char \*str)

{

while (\*str) {

LCD\_Char(\*str); // Display each character

str++;

}

// UART initialization for transmitting and receiving data

void UART\_Init()

{

TMOD = 0x20; // Timer 1, 8-bit auto reload mode

TH1 = 0xFD; // Load value for 2400 baud rate

TL1 = 0xFD; // Load value for 2400 baud rate

SCON = 0x50; // Mode 1: 8-bit data, 1 stop bit, reception enabled

TR1 = 1; // Start Timer 1

}

// Function to receive data from UART (PC to MCU)

void UART\_ReceiveData()

{

while (RI == 0); // Wait until data is received

RI = 0; // Clear the receive interrupt flag

data\_buffer[buffer\_index] = SBUF; // Store the received byte in the buffer

buffer\_index++; // Move to the next position in the buffer

if (buffer\_index >= BUFFER\_SIZE) {

buffer\_index = 0; // Reset buffer if full (optional)

}

}

// Function to transmit data via UART (MCU to PC)

void UART\_Transmit(char data)

{

SBUF = data; // Load data into UART buffer for transmission

while (TI == 0); // Wait until the transmission is complete

TI = 0; // Clear the transmit interrupt flag

}

// Function to send data back to PC (send received data)

void UART\_SendBackData()

{

unsigned int i;

for (i = 0; i < buffer\_index; i++) {

UART\_Transmit(data\_buffer[i]); // Send each byte from the buffer

}

}

// Function to calculate transmission speed in real-time

void Calculate\_Transmission\_Speed(unsigned int bits\_received, unsigned long time\_elapsed)

{

unsigned long speed = 0;

// Calculate speed (bits per second)

if (time\_elapsed > 0) {

speed = (bits\_received \* 1000) / time\_elapsed; // Convert ms to seconds

}

char speed\_str[10];

sprintf(speed\_str, "Speed: %ld bps", speed);

LCD\_Command(0x01); // Clear screen

LCD\_String(speed\_str); // Display speed on LCD

}

// Main Program

void main()

{

char received\_data;

unsigned int bits\_received = 0;

unsigned long start\_time, current\_time;

unsigned int i;

TMOD = 0x01; // Timer0 in mode 1 (16-bit timer mode)

TH0 = 0xFC; // Load for 1ms interval

TL0 = 0x18; // Load for 1ms interval

ET0 = 1; // Enable Timer0 interrupt

EA = 1; // Enable global interrupt

TR0 = 1; // Start Timer0

UART\_Init(); // Initialize UART

LCD\_Command(0x38); // Initialize LCD (8-bit mode, 2 lines)

LCD\_Command(0x0C); // Display ON, Cursor OFF

LCD\_Command(0x01); // Clear screen

LCD\_Command(0x06); // Move cursor to next position after each character

LCD\_String("Waiting for data..."); // Display message on LCD

start\_time = timer\_counter; // Capture the start time (in ms)

while (1)

{

// Receive data from PC and store it in the buffer

UART\_ReceiveData();

// After receiving the data, send the received data back to the PC

UART\_SendBackData();

// Calculate transmission speed (in bits per second)

current\_time = timer\_counter; // Update with actual timer value (time in ms)

unsigned long elapsed\_time = current\_time - start\_time;

bits\_received += 8; // Each character has 8 bits

Calculate\_Transmission\_Speed(bits\_received, elapsed\_time);

}

}