Embedded Lab

Embedded Systems tutorials, projects, and more ...

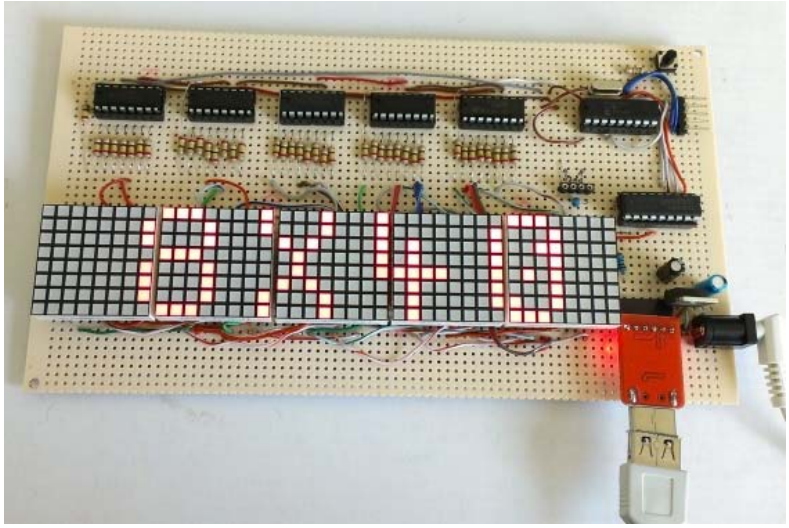
Quickturn PCB & Assembly
Your one-stop source for all types of PCBs and Full Turnkey PCB Assembly.
OPEN

M E USING SHIFT R

comments |

CB & Assembly
rce for all types of PCBs and Full
embly.
OPEN

LE display text, graphics, animations, and video, and therefore, they have become a popular mean of displaying information these days. You can see them at gas stations displaying the gas prices, or in the public places displaying information, and alongside highways displaying advertisements on large dot matrix panels. This project is about constructing a mono-color LED matrix display board that consists 320 LEDs arranged in 8 rows and 40 columns. The heart of this project is PIC16F1847 microcontroller which receives data from a PC through a serial port (or USB using an USB-UART interface), and display on the LED matrix with the help of five 74HC595 shift registers.



8x40 LED Matrix Display

Theory

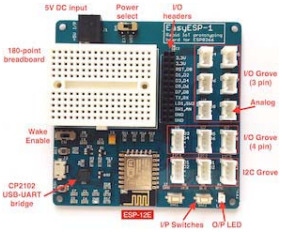
If you are not familiar with LED matrices at all, I strongly suggest to read my two experimental tutorials that were posted earlier: Basics of LED matrix display and Scrolling text message on an LED matrix. The first tutorial describes the basic structure of LED matrices and the multiplexing technique of driving them to display static characters. The second tutorial is focussed more on creating animation and demonstrates the concept of displaying a scrolling message on a matrix of 16×8 LEDs. This project is basically a continuation of those two tutorials

Protoype PCB Assembly

Search...

ESP8266 LEARNING MADE EASY

EasyESP-1 is a rapid IoT prototyping board for ESP8266



Ships worldwide
Get one for yourself

Circuit diagrams

Five 8×8 square LED display modules are arranged in a linear fashion to construct an 8-row X 40-column display matrix. The LED display modules I used in this project are NFM-12883AS from Futurlec. They are common-row-cathode type, which means the cathodes of all the LEDs in each individual row are interconnected, and therefore each row pin is supposed to sink currents from eight LEDs in that row. The similar row pins of all 5 modules are further connected together and are fed to the the output sinks of an ULN2803 IC, which consists of 8 Darlington arrays. Now each output pin of ULN2803 is a sink for 40 (=8×5) LEDs in that particular row. Since ULN2803 can sink current up to 500 mA per pin, the current per LED is limited to 500/40 = 12.5 mA.

The anode terminals of LEDs are accessible through column pins. Since we are going to apply row scanning technique (read Scrolling text message on an LED matrix), each column pin needs to source current for only one LED, as only one row is active in that column at a time. The columns are driven by the outputs of a 40-bit serial-in-parallel-out arrangement constructed by cascading five 74HC595 ICs. The 74HC595 device is an 8-bit serial-input and parallel/serial output shift register. The serial output feature allows cascading of multiple 74HC595 devices. From engineering point of view, 74HC595 may not be a good choice as column drivers because they are not meant for sourcing current for LEDs. But several experiments done by people (including me) have shown that they can drive LEDs reasonably bright, and therefore can be used as column drivers in an LED matrix project like this.

The microcontroller used in this project is PIC16F1847 from the enhanced mid-range 8-bit PIC family. It runs at 32 MHz using its internal clock source (with PLL enabled). PORTB provides ground path for individual rows (cathode) through the ULN2803 darlington pair array, whereas RA0, RA1, and RA2 pins are used to control the Data, SH_CP, and ST_CP lines of the cascaded shift registers. Pin 8 and 7 of PIC16F1847 are the default pins for UART communication. They go to the USB-UART module for serial communication with the PC. A serial terminal program running on the PC can be used to send character data to the microcontroller which saves the received information into its internal EEPROM.

ACCELERATE DEVOPS MOMENTUM
Learn More ›
Quest

SUBSCRIBE

Embedded Lab
18,562 likes
Like Page

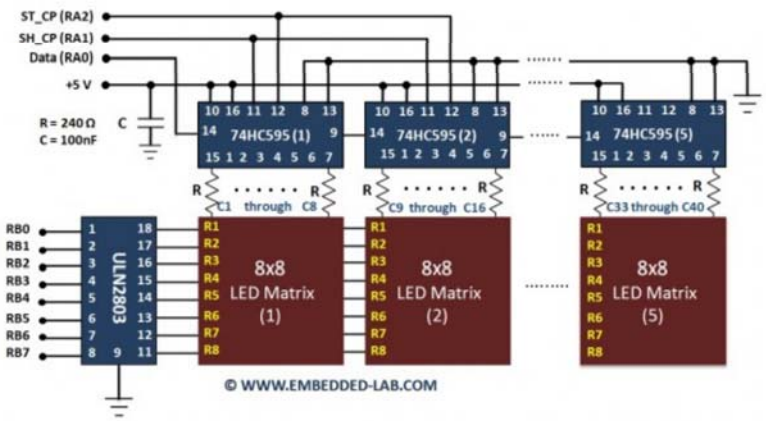
2 Layers $10 ea
4 Layers $25 ea
Order Now
RUSHPCB
ISO 9001:2008, ITAR, IL LISTED
Locates in Silicon Valley, San Jose, CA

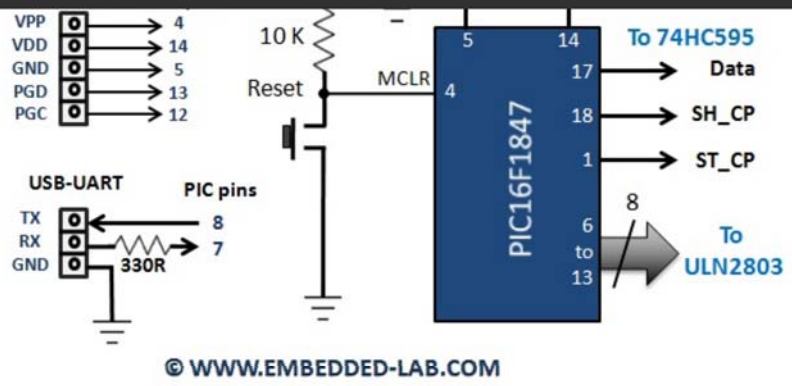
Follow @EmbeddedLab
rough email
Go to Top Contact Us Privacy Policy
© 201 Embedded Lab All Rights Reserved
Sign Up

I SELL ON TINDIE
Donate

Read Our Privacy Policy!

MOST POPULAR PAGES
ESP8266 projects
Tiva C tutorials
AVR XMEGA tutorials

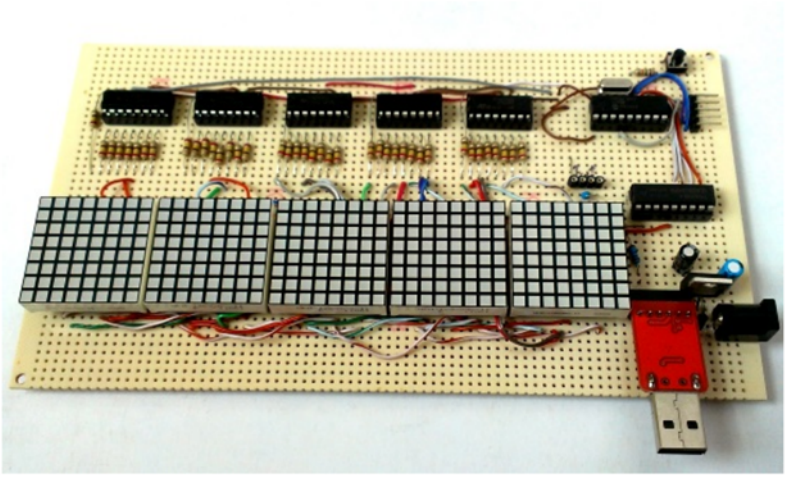




Microcontroller circuit

I constructed the above circuit on a general purpose perforated prototyping board. You definitely need a bigger one to fit all 5 LED display modules along with other components. I used VECTOR ELECTRONICS 3-hole solder pad circuit board from Newark which is a very good quality 4.5" x 8.08" perfboard. An LM7805 IC is used to derive regulated +5V power supply. The power supply circuit is not shown above because it is a very common circuit and I assumed you know it.

The following picture shows the finished LED display board with an USB-UART module plugged in. [A crystal oscillator is also seen in the board but it is not really used. The microcontroller runs at 32MHz using its internal clock source. I first thought I would use an external clock, but later changed my mind but I was lazy to take the crystal off the board.]



Completed 8x40 LED matrix display board

Software

The firmware for this project is developed using mikroC Pro for PIC compiler. The program uses an 8×5 byte (=320 bits) array (Buffer[8][5]) to store the display pixel information. Each bit corresponds to one LED. The user can send message from PC to the display board using a serial terminal program. I am using Termite 2.9 program on Windows. The message sent is saved onto the PIC's internal EEPROM. The first two EEPROM locations are reserved for a data input flag and number of characters in the message. The following sequence describes in detail how the firmware of this project works.

1. Power is turned ON. The PIC MCU initializes the ports, and listen to its serial port. It prints a message 'ARE YOU READY FOR YOUR INPUT? Y/N' to serial port and the message is displayed on PC serial terminal application window. The user gets 3 seconds to send character 'Y'. Otherwise, the MCU

Product Reviews
PIC Projects
Arduino Projects
PIC Tutorials

CATEGORIES

Categories
Select Category

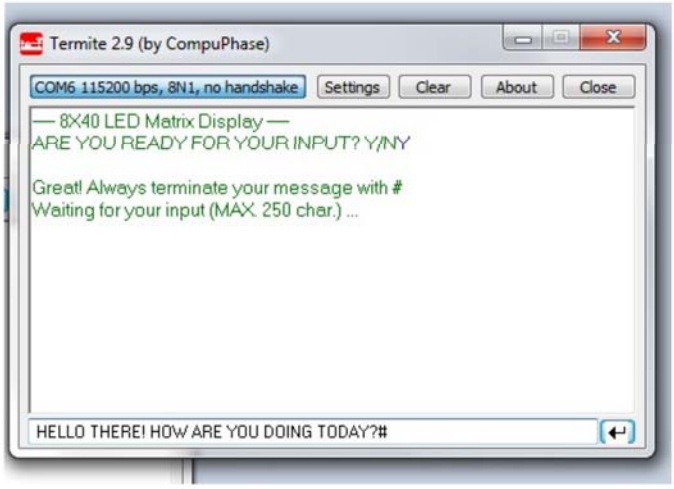
sent from the PC. The MCU then writes 1 to EEPROM location 0x00, and saves the number of bytes received at EEPROM location 0x01. The received characters are sequentially saved into sequential EEPROM locations starting from 0x02.

3. Next, the MCU loads the character data from EEPROM into RAM and display it on the LED matrix scrolling from right to left. Each character is mapped to its graphic font which is saved in to the program memory as static array. Please read Scrolling text message on an LED matrix tutorial for more detail on LED matrix font.

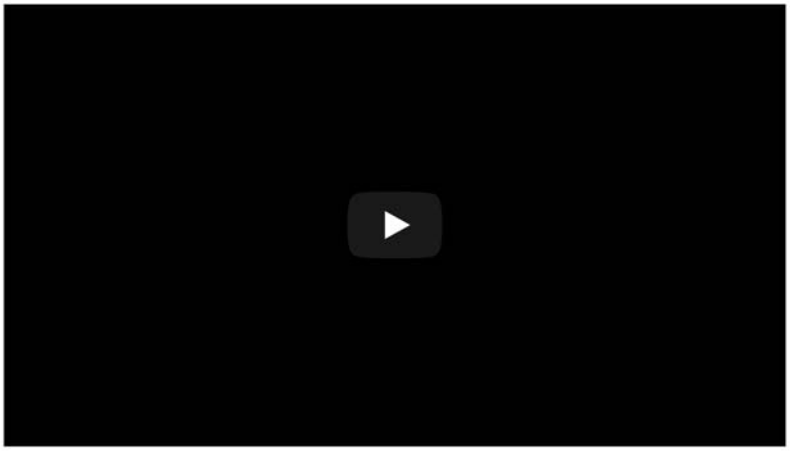
Share

Download the complete source code and compiled HEX file

Output



Data input from a serial terminal application window running on PC



JLCPCB: Prototype 10 PCBs for $2 (2-layer,100*100mm)

China's Largest PCB Prototype Manufacturer, 290,000+ Customers & 8,000+ Online Orders Per Day

PCB Pricing: https://jlcpcb.com/quote

10 Best Practices for
Creating Effective Dashboards
GET THE WHITEPAPER
+ableau

Share 535 Tweet 11 Email 5 Share 2462
Like 211

Related Posts

Share

Free 10PCS 2Layer PCB

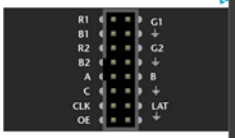
Ad firstpcb

16x32 RGB Matrix Panel Driver Shield Revision 1

embedded-lab.com

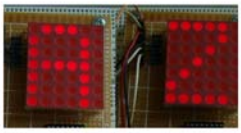
Another ESP8266 internet clock using LED matrix panels

embedded-lab.com



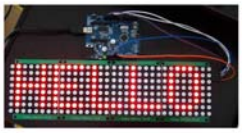
Hookup guide for 16x32 RGB LED pane - Part 2

embedded-lab.com

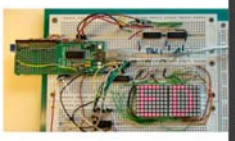


Lab 12: Basics of LED dot matrix display

Hookup guide for 16x32 RGB LED panel - Part 1



EasyMatrix2



Lab 15: Scrolling text message on an LED dot-matrix display

Tagged display board, led matrix, mikroC, PIC16F1847

Embedded Lab Projects | PIC Projects

139 COMMENTS

Shashwat Rakesh October 12, 2017 1:20 am

Sir
I am trying the same project with 8051 . can you send me code along with proteus simulation.

Reply

Oladipo, Mumin Olatunji April 26, 2017 7:25 pm

i cant get it to work on proteus. kindly assist me with the simulation file

Reply

S.k.joy April 20, 2017 2:21 am

Sir, I am very much interested with your project. Could you send me the code for Pic16f873A? God bless you.

Hi..
I want to do this project in my acedmic , so please suggasted me about that . can i use pic 18f4550 microcontroller.

Reply

BobA December 3, 2016 10:39 am

Oh, please make it and code it for me and send it to me by Christmas. Oh, and one for my uncle too... 🙂

I just explored your project looking for ideas for my own Arduino project. Not sure that shift registers will be fast enough for me on a 7×10 bargraph display. I might need to latch data. I'll figure it out, thanks for posting your own work! I'm afraid to post mine, considering all the people who want someone to do everything for them!

Reply

bright October 7, 2016 4:31 pm

With due respect sir, I need a complete coding of a matrix board using a remote control, to control it THANK

Reply

martins July 15, 2016 2:01 pm

Goog day sir, i want to build your 8&40 led scroling text massage, but the ploblem is that i want to replace pic16f1847 to pic18f2550 i dont kown if it can work,bet if it possible plese can you write the source code and the hex file, circuite diagram and i want the led to display somethine like this, text will display and end, the next step will start by flying up and down and pus for some seconds and start nomal. thanks am waiting for your reply thanks

Reply

manik July 3, 2016 3:05 am

hello sir, I am making a 4 segment counter using 4 shift register 4094 by cascading.so how can I will sent data serially. I have also followed your matrix display shifting c program.plz help me by given data sending mikroc c program.plz plz plz sir.
tm.manik.3e@gmail.com

Reply

James F January 5, 2016 2:49 pm

Hi sir,
I'm working on a 108×24 LED matrix scrolling display.
I used shift registers for the columns and decade counter for the rows.

now I'm having problems with the code, I'm not an expert on coding,
so I'm asking you to help me make it work.

it supposes to work via Bluetooth communication.

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char
0x03, 0xFF, 0xFF, 0x61, 0xFF, 0xFF, 0xF1, 0xFF, 0xFF, 0x61, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char !
0x06, 0xFE, 0x01, 0x00, 0xFE, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFE, 0x01, 0x00, 0xFE, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char "
0x0D, 0x00, 0xC3, 0x00, 0x00, 0xC3, 0x00, 0x00, 0xC3, 0x7F, 0x00, 0xFF, 0x7F,
0xFE, 0xFF, 0x00, 0xFE, 0xC3, 0x00, 0x00, 0xC3, 0x00, 0x00, 0xC3, 0x7F, 0x00, 0xFF,
0x7F, 0xFE, 0xFF, 0x00, 0xFE, 0xC3, 0x00, 0x00, 0xC3, 0x00, 0x00, 0xC3, 0x00,
0x00, 0x00, 0x00, // Code for char #
0x0D, 0xC0, 0x03, 0x0C, 0x00, 0xE0, 0x07, 0x1C, 0x00, 0xF0, 0x0F, 0x38, 0x00,
0xF0, 0x1F, 0x38, 0x00, 0xF8, 0x1F, 0xFE, 0x01, 0x38, 0x3E, 0xFE, 0x01, 0x38, 0x3C,
0xFE, 0x01, 0xFF, 0x78, 0x38, 0x00, 0xFF, 0xF8, 0x3F, 0x00, 0xFF, 0xF0, 0x1F, 0x00,
0x38, 0xF0, 0x1F, 0x00, 0x70, 0xE0, 0x0F, 0x00, 0x60, 0x80, 0x07, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char $
0x0E, 0xF8, 0x1F, 0xF0, 0x1C, 0x38, 0xF0, 0x0C, 0x30, 0x3C, 0x1C, 0x38, 0x0F,
0xF8, 0x9F, 0x0F, 0xF0, 0xFF, 0x03, 0x00, 0xFC, 0x01, 0x00, 0x7F, 0x00, 0x00, 0xFF,
0x07, 0xC0, 0xF3, 0x1F, 0xF0, 0x18, 0x38, 0x7C, 0x0C, 0x30, 0x1F, 0x18, 0x18,
0x0F, 0xF0, 0x0F, // Code for char %
0x0E, 0x00, 0x80, 0x0F, 0xC0, 0xE1, 0x1F, 0xE0, 0xF3, 0x3F, 0xF8, 0xFF, 0xF8, 0xFC,
0x7F, 0xE0, 0x3E, 0x3F, 0xC0, 0x1E, 0x3E, 0xE0, 0x4E, 0x3E, 0xF0, 0x1E, 0x7F, 0x78,
0xFC, 0xFB, 0x3F, 0xF8, 0xE3, 0x1F, 0xF0, 0x81, 0x3F, 0x00, 0xF0, 0xFF, 0x00, 0xF8,
0xF0, // Code for char &
0x06, 0x00, 0x01, 0x00, 0xF0, 0x01, 0x00, 0xFC, 0x01, 0x00, 0xFC, 0x00, 0x00,
0x3C, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char '
0x06, 0x80, 0xFF, 0x01, 0xE0, 0xFF, 0x07, 0xFC, 0xFF, 0x3F, 0x7E, 0x00, 0x3C, 0x0F,
0x00, 0xF0, 0x03, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char (
0x06, 0x03, 0x00, 0xC0, 0x0F, 0x00, 0xF0, 0xFE, 0x00, 0x7E, 0xFC, 0xFF, 0x3F, 0xE0,
0xFF, 0x07, 0x80, 0xFF, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char )
0x0B, 0x80, 0x00, 0x00, 0xC8, 0x09, 0x00, 0xD8, 0x1D, 0x00, 0xBC, 0x1E, 0x00,
0xFC, 0x07, 0x00, 0xF0, 0x07, 0x00, 0xFC, 0x07, 0x00, 0xBC, 0x1E, 0x00, 0xD8,
0x1D, 0x00, 0xC8, 0x09, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char *
0x0C, 0x00, 0x38, 0x00, 0x00, 0x38, 0x00, 0x00, 0x38, 0x00, 0x00, 0x38, 0x00,
0xF8, 0xFF, 0x1F, 0xF8, 0xFF, 0x1F, 0xF8, 0xFF, 0x1F, 0xF8, 0xFF, 0x1F, 0x00, 0x38,
0x00, 0x00, 0x38, 0x00, 0x00, 0x38, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char +
0x06, 0x00, 0x00, 0x70, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x7C,
0x00, 0x00, 0x3C, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char ,
0x0C, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00,
0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00,
0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char –
0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x00, 0x78, 0x00, 0x00, 0xFC,
0x00, 0x00, 0xFC, 0x00, 0x00, 0x78, 0x00, 0x00, 0x30, 0x00, 0x00, 0x00, 0x00,
```

Share

```
0x00, 0x00, 0x00, // Code for char /
0x0D, 0xE0, 0xFF, 0x03, 0xF0, 0xFF, 0x07, 0xFC, 0xFF, 0x1F, 0x7E, 0x00, 0x7E, 0x1F,
0x00, 0xF8, 0x0F, 0x00, 0xF0, 0x07, 0x00, 0xE0, 0x0F, 0x00, 0xF0, 0x1F, 0x00, 0xF8,
0x7E, 0x00, 0x7E, 0xFC, 0xFF, 0x1F, 0xF0, 0xFF, 0x07, 0xE0, 0xFF, 0x03, 0x00, 0x00,
0x00, // Code for char 0
0x09, 0x60, 0x00, 0x00, 0x78, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x0C, 0x00, 0x00,
0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char 1
0x0E, 0x00, 0x00, 0xE0, 0x1C, 0x00, 0xF0, 0x1E, 0x00, 0xF8, 0x1F, 0x00, 0xFC, 0x0F,
0x00, 0xFE, 0x07, 0x00, 0xFF, 0x07, 0x80, 0xFF, 0x07, 0xE0, 0xEF, 0x0F, 0xF0, 0xE7,
0x1F, 0xFC, 0xE3, 0xFF, 0xFF, 0xE0, 0xFE, 0x7F, 0xE0, 0xFC, 0x3F, 0xE0, 0xF0, 0x0F,
0xE0, // Code for char 2
0x0E, 0x1C, 0x00, 0x38, 0x1E, 0x00, 0x78, 0x1F, 0x00, 0xF8, 0x1F, 0x00, 0xF8, 0x0F,
0x18, 0xF0, 0x0F, 0x18, 0xF0, 0x0F, 0x18, 0xF0, 0x0F, 0x3C, 0xF0, 0x1F, 0x7E, 0xF8,
0xFF, 0xFF, 0xFF, 0xFE, 0xFF, 0x7F, 0xFC, 0xE7, 0x3F, 0xF8, 0xC3, 0x1F, 0xF0, 0x81,
0x0F, // Code for char 3
0x0E, 0x00, 0xF8, 0x00, 0x00, 0xFE, 0x00, 0x80, 0xEF, 0x00, 0xC0, 0xE3, 0x00,
0xF0, 0xE0, 0x00, 0x38, 0xE0, 0x00, 0x1C, 0xE0, 0x00, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0xE0, 0x00, 0x00, 0xE0, 0x00, 0x00,
0xE0, 0x00, // Code for char 4
0x0C, 0xFF, 0x03, 0xFC, 0xFF, 0x03, 0xFC, 0xFF, 0x03, 0xFC, 0xFF, 0x03, 0xE0, 0x07,
0x07, 0xE0, 0x07, 0x07, 0xF0, 0x07, 0x07, 0xF8, 0x07, 0x0E, 0x7E, 0x07, 0xFE, 0x3F,
0x07, 0xFC, 0x1F, 0x07, 0xF8, 0x0F, 0x07, 0xE0, 0x07, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char 5
0x0C, 0x00, 0xFF, 0x0F, 0xC0, 0xFF, 0x1F, 0xE0, 0xFF, 0x7F, 0xF0, 0x3F, 0xFC, 0xFC,
0x1F, 0xF8, 0x7E, 0x0E, 0xF0, 0x3F, 0x06, 0xE0, 0x0F, 0x0E, 0xF0, 0x07, 0x1E, 0x78,
0x07, 0xFC, 0x3F, 0x07, 0xFC, 0x1F, 0x07, 0xF8, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char 6
0x0C, 0x07, 0x00, 0xF8, 0x07, 0x80, 0xFF, 0x07, 0xE0, 0xFF, 0x07, 0xF8, 0x1F, 0x07,
0xFF, 0x0F, 0xC7, 0xFF, 0x03, 0xF7, 0x7F, 0x00, 0xFF, 0x1F, 0x00, 0xFF, 0x07, 0x00,
0xFF, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char 7
0x0B, 0xF0, 0xC1, 0x0F, 0xFC, 0xE1, 0x3F, 0xFE, 0xF3, 0x7F, 0x1E, 0x7F, 0x7C, 0x0F,
0x3E, 0xF8, 0x07, 0x1C, 0xF0, 0x0F, 0x3E, 0xF8, 0x1E, 0x7F, 0x7C, 0xFE, 0xF3, 0x7F,
0xFC, 0xE1, 0x3F, 0xF0, 0xC0, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char 8
0x0C, 0x00, 0x00, 0xC0, 0xF0, 0x1F, 0xE0, 0xFC, 0x3F, 0xE0, 0xFC, 0x3F, 0xF0, 0x1F,
0x78, 0xF8, 0x0F, 0x70, 0x78, 0x07, 0x60, 0x3C, 0x0F, 0x70, 0x1E, 0x1F, 0xF8, 0x1F,
0xFC, 0xFF, 0x0F, 0xFC, 0xFF, 0x03, 0xF0, 0xFF, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char 9
0x06, 0xC0, 0x00, 0x03, 0xE0, 0x81, 0x07, 0xE0, 0x81, 0x07, 0xE0, 0x81, 0x07,
0xE0, 0x81, 0x07, 0xC0, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, // Code for char :
0x07, 0x00, 0x00, 0x70, 0xC0, 0x00, 0x7F, 0xE0, 0x81, 0x7F, 0xE0, 0x81, 0x7F, 0xE0,
0x81, 0x1F, 0xE0, 0x81, 0x07, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char ;
0x0E, 0x00, 0x18, 0x00, 0x00, 0x18, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3E, 0x00,
0x00, 0x77, 0x00, 0x00, 0x77, 0x00, 0x80, 0xE3, 0x00, 0x80, 0xE3, 0x00, 0xC0,
0xC1, 0x01, 0xC0, 0xC1, 0x01, 0xE0, 0x80, 0x03, 0xE0, 0x80, 0x03, 0x70, 0x00,
0x07, 0x70, 0x00, 0x07, // Code for char
0x0A, 0x0C, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0F, 0xE0, 0x38, 0x07, 0xFC, 0x7C,
0x07, 0xFE, 0x7C, 0x0F, 0xFF, 0x38, 0xFF, 0x1F, 0x00, 0xFE, 0x0F, 0x00, 0xFC, 0x07,
0x00, 0xF0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char ?
0x0E, 0xC0, 0xFF, 0x0F, 0xE0, 0xFF, 0x1F, 0x30, 0x00, 0x38, 0x30, 0x00, 0x30, 0x18,
0x7F, 0x30, 0x98, 0xC3, 0x30, 0x98, 0x81, 0x31, 0x98, 0x81, 0x31, 0x98, 0x81,
```

Share

```
0xFF, // Code for char A
0x0D, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
0x1C, 0xE0, 0x07, 0x1C, 0xE0, 0x07, 0x1C, 0xE0, 0x07, 0x3E, 0xF0, 0x0F, 0x7F,
0xF8, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0xFC, 0xF3, 0x3F, 0xF0, 0xC0, 0x0F, 0x00,
0x00, 0x00, // Code for char B
0x0D, 0xC0, 0xFF, 0x03, 0xF0, 0xFF, 0x0F, 0xF8, 0xFF, 0x1F, 0xFC, 0x00, 0x7F, 0x3E,
0x00, 0x7E, 0x1F, 0x00, 0xF8, 0x0F, 0x00, 0xF0, 0x07, 0x00, 0xE0, 0x07, 0x00, 0xE0,
0x07, 0x00, 0xE0, 0x07, 0x00, 0xE0, 0x07, 0x00, 0xE0, 0x07, 0x00, 0x60, 0x00,
0x00, 0x00, // Code for char C
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0xE0, 0x07,
0x00, 0xE0, 0x07, 0x00, 0xE0, 0x07, 0x00, 0xE0, 0x0F, 0x00, 0xF0, 0x3E, 0x00,
0x7C, 0x7C, 0x00, 0x7E, 0xFC, 0x81, 0x3F, 0xF8, 0xFF, 0x1F, 0xF0, 0xFF, 0x0F, 0xC0,
0xFF, 0x01, // Code for char D
0x0C, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F,
0x3C, 0xF0, 0x0F, 0x3C, 0xF0, 0x0F, 0x3C, 0xF0, 0x0F, 0x3C, 0xF0, 0x0F, 0x3C, 0xF0,
0x0F, 0x3C, 0xF0, 0x0F, 0x3C, 0xF0, 0x0F, 0x3C, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char E
0x0D, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F,
0x78, 0x00, 0x0F, 0x78, 0x00, 0x0F, 0x78, 0x00, 0x0F, 0x78, 0x00, 0x0F, 0x78, 0x00,
0x0F, 0x78, 0x00, 0x0F, 0x78, 0x00, 0x0F, 0x78, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char F
0x0E, 0xE0, 0xFF, 0x07, 0xF0, 0xFF, 0x1F, 0xF8, 0xFF, 0x3F, 0x7C, 0x00, 0x7E, 0x3E,
0x00, 0xFC, 0x1F, 0x00, 0xF8, 0x0F, 0x00, 0xF0, 0x07, 0x00, 0xE0, 0x07, 0x38, 0xE0,
0x07, 0x38, 0xF0, 0x07, 0x38, 0xF8, 0x07, 0xF8, 0xFF, 0x0F, 0xF8, 0x7F, 0x0E, 0xF0,
0x3F, // Code for char G
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C,
0x00, 0x00, 0x3C, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, // Code for char H
0x06, 0x03, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0x03, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char I
0x0E, 0x00, 0x00, 0x1F, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x7C, 0x00, 0x00, 0xF0,
0x07, 0x00, 0xF0, 0x07, 0x00, 0xF0, 0x07, 0x00, 0xF8, 0xFF, 0xFF, 0x7F, 0xFF, 0xFF,
0x3F, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0x07, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07,
0x00, 0x00, // Code for char J
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x80,
0x3F, 0x00, 0xC0, 0x7F, 0x00, 0xE0, 0xFB, 0x01, 0xF0, 0xF1, 0x07, 0xF8, 0xE1, 0x0F,
0xFC, 0xC0, 0xFF, 0x3F, 0x00, 0xFF, 0x1F, 0x00, 0xFE, 0x0F, 0x00, 0xF8, 0x07, 0x00,
0xE0, // Code for char K
0x0C, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00,
0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char L
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFC, 0x03, 0x00, 0xF0,
0x0F, 0x00, 0xC0, 0x1F, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF, 0x00, 0xC0, 0x1F, 0x00,
0xF0, 0x0F, 0x00, 0xFC, 0x03, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, // Code for char M
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0,
0x07, 0x00, 0xE0, 0x1F, 0x00, 0x80, 0x3F, 0x00, 0x00, 0xFE, 0x00, 0x00, 0xFC,
0x07, 0x00, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, // Code for char N
0x0D, 0xC0, 0xFF, 0x07, 0xF0, 0xFF, 0x1F, 0xF8, 0xFF, 0x3F, 0xFC, 0x81, 0x3F, 0x3E,
0x00, 0x7C, 0x1E, 0x00, 0x78, 0x1E, 0x00, 0x78, 0x1E, 0x00, 0x78, 0x3E, 0x00,
0x7C, 0xFC, 0x81, 0x3F, 0xF8, 0xFF, 0x3F, 0xF0, 0xFF, 0x1F, 0xC0, 0xFF, 0x07, 0x00,
0x00, 0x00, // Code for char O
0x0D, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
0x70, 0x00, 0x07, 0x70, 0x00, 0x07, 0x70, 0x00, 0x0F, 0x78, 0x00, 0x1F, 0x3C,
```

Share

```
0x00, 0xE0, // Code for char Q
0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
0x78, 0x00, 0x07, 0x78, 0x00, 0x07, 0x78, 0x00, 0x07, 0x7C, 0x00, 0x07, 0xFC,
0x01, 0x0F, 0xFE, 0xFF, 0xFE, 0xEF, 0xFF, 0xFE, 0xC7, 0xFF, 0xFC, 0xC7, 0xFF, 0xF0,
0x01, 0xFF, // Code for char R
0x0D, 0x00, 0x00, 0x3C, 0xF8, 0x00, 0x78, 0xFC, 0x01, 0xF0, 0xFE, 0x07, 0xE0,
0xFF, 0x0F, 0xC0, 0xFF, 0x0F, 0xC0, 0x0F, 0x1F, 0xE0, 0x07, 0x1E, 0xF0, 0x03, 0x7E,
0xFC, 0x07, 0xFC, 0x7F, 0x0F, 0xF8, 0x3F, 0x0E, 0xF0, 0x1F, 0x0C, 0xE0, 0x07, 0x00,
0x00, 0x00, // Code for char S
0x0E, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00,
0x07, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00, 0x07, 0x00, 0x00,
0x07, 0x00, 0x00, // Code for char T
0x0E, 0xFF, 0xFF, 0x07, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0x3F, 0xFF, 0xFF, 0x7F, 0x00,
0x00, 0xFE, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x00,
0xF8, 0x00, 0x00, 0xFE, 0xFF, 0xFF, 0x7F, 0xFF, 0xFF, 0x3F, 0xFF, 0xFF, 0x1F, 0xFF,
0xFF, 0x07, // Code for char U
0x0D, 0xFF, 0x07, 0x00, 0xFF, 0x1F, 0x00, 0xFF, 0xFF, 0x01, 0x00, 0xFC, 0x07, 0x00,
0xF0, 0x1F, 0x00, 0x80, 0xFF, 0x00, 0x00, 0xFE, 0x00, 0x80, 0xFF, 0x00, 0xF0, 0x1F,
0x00, 0xFC, 0x07, 0xFF, 0xFF, 0x01, 0xFF, 0x3F, 0x00, 0xFF, 0x07, 0x00, 0x00, 0x00,
0x00, // Code for char V
0x0E, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0x7F, 0xFF, 0xFF, 0xFF, 0x00, 0xC0, 0xFF, 0x00,
0x80, 0x7F, 0x00, 0xE0, 0x0F, 0x00, 0xFE, 0x03, 0x00, 0xFE, 0x03, 0x00, 0xE0, 0x0F,
0x00, 0x80, 0x7F, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0xFF, 0xFF,
0x0F, // Code for char W
0x0E, 0x03, 0x00, 0xC0, 0x07, 0x00, 0xF8, 0x1F, 0x00, 0xFF, 0xFE, 0xC0, 0x7F, 0xF8,
0xE3, 0x1F, 0xE0, 0xFF, 0x03, 0x80, 0x3F, 0x00, 0x80, 0x3F, 0x00, 0xE0, 0xFF, 0x03,
0xF8, 0xE3, 0x1F, 0xFE, 0xC0, 0x7F, 0x1F, 0x00, 0xFF, 0x07, 0x00, 0xF8, 0x03, 0x00,
0xC0, // Code for char X
0x0E, 0x0F, 0x00, 0x00, 0x7F, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF, 0x03, 0x00, 0xE0,
0x0F, 0x00, 0x80, 0xFF, 0xFF, 0x00, 0xFE, 0xFF, 0x00, 0xFE, 0xFF, 0x80, 0xFF, 0xFF,
0xE0, 0x0F, 0x00, 0xFF, 0x03, 0x00, 0xFF, 0x01, 0x00, 0x7F, 0x00, 0x00, 0x0F, 0x00,
0x00, // Code for char Y
0x0E, 0x07, 0x00, 0xF0, 0x07, 0x00, 0xFC, 0x07, 0x00, 0xFF, 0x07, 0xC0, 0xFF, 0x07,
0xF0, 0xEF, 0x07, 0xFC, 0xE3, 0x07, 0xFF, 0xE0, 0x87, 0x3F, 0xE0, 0xE7, 0x0F, 0xE0,
0xFF, 0x03, 0xE0, 0xFF, 0x00, 0xE0, 0x7F, 0x00, 0xF8, 0x1F, 0x00, 0xF8, 0x07, 0x00,
0xF8, // Code for char Z
0x08, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F,
0x00, 0xF0, 0x0F, 0x00, 0xF0, 0x0F, 0x00, 0xF0, 0x0F, 0x00, 0xF0, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char [
0x0D, 0x03, 0x00, 0x00, 0x1F, 0x00, 0x00, 0xFF, 0x00, 0x00, 0xFF, 0x07, 0x00, 0xFE,
0x3F, 0x00, 0xF0, 0xFF, 0x01, 0x80, 0xFF, 0x0F, 0x00, 0xFC, 0x7F, 0x00, 0xE0, 0xFF,
0x00, 0x00, 0xFF, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00,
0x00, 0x00, // Code for char BackSlash
0x08, 0x0F, 0x00, 0xF0, 0x0F, 0x00, 0xF0, 0x0F, 0x00, 0xF0, 0x0F, 0x00, 0xF0, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char ]
0x0E, 0x00, 0x08, 0x00, 0x00, 0x0E, 0x00, 0x80, 0x0F, 0x00, 0xE0, 0x03, 0x00,
0xF8, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x3C,
0x00, 0x00, 0xF8, 0x00, 0x00, 0xE0, 0x03, 0x00, 0x80, 0x0F, 0x00, 0x00, 0x0E,
0x00, 0x00, 0x08, 0x00, // Code for char ^
0x0D, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0,
0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00,
0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00, 0xC0, 0x00, 0x00,
0xC0, 0x00, 0x00, 0x00, // Code for char _
0x07, 0x00, 0x03, 0x00, 0xE0, 0x03, 0x00, 0xF8, 0x03, 0x00, 0xFC, 0x01, 0x00,
0xFC, 0x01, 0x00, 0x7C, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, // Code for char a
0x0E, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0x80,
0x07, 0x3C, 0x80, 0x03, 0x38, 0x80, 0x01, 0x30, 0x80, 0x03, 0x38, 0x80, 0x07,
0x3C, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x1F, 0x00, 0xFF, 0x1F, 0x00, 0xFE, 0x0F, 0x00,
0xF8, 0x01, // Code for char b
0x0D, 0x00, 0xF0, 0x03, 0x00, 0xFC, 0x0F, 0x00, 0xFE, 0x1F, 0x00, 0xFF, 0x1F, 0x00,
0xFF, 0x3F, 0x80, 0x0F, 0x3E, 0x80, 0x03, 0x38, 0x80, 0x01, 0x30, 0x80, 0x01, 0x30,
0x80, 0x01, 0x30, 0x80, 0x03, 0x38, 0x80, 0x03, 0x38, 0x00, 0x03, 0x18, 0x00,
0x00, 0x00, // Code for char c
0x0D, 0x00, 0xFC, 0x07, 0x00, 0xFE, 0x0F, 0x00, 0xFF, 0x1F, 0x00, 0xFF, 0x3F, 0x80,
0x0F, 0x3C, 0x80, 0x03, 0x38, 0x80, 0x01, 0x30, 0x80, 0x03, 0x38, 0x80, 0x07,
0x3C, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x1F, 0x00,
0x00, 0x00, // Code for char d
0x0E, 0x00, 0xFC, 0x07, 0x00, 0xFE, 0x0F, 0x00, 0xFF, 0x1F, 0x00, 0xFF, 0x3F, 0x80,
0x67, 0x3E, 0x80, 0x63, 0x38, 0x80, 0x61, 0x30, 0x80, 0x63, 0x30, 0x80, 0x67,
0x30, 0x80, 0x7F, 0x30, 0x00, 0x7F, 0x38, 0x00, 0x7F, 0x38, 0x00, 0x7E, 0x18, 0x00,
0x78, 0x00, // Code for char e
0x0B, 0x00, 0x03, 0x00, 0x00, 0x03, 0x00, 0xF0, 0xFF, 0x7F, 0xFC, 0xFF, 0x7F, 0xFC,
0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0x0E, 0x03, 0x00, 0x06, 0x03, 0x00,
0x06, 0x03, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char f
0x0D, 0xF0, 0x3F, 0x30, 0xFC, 0x7F, 0x78, 0xFC, 0x7F, 0xEC, 0xFE, 0xFF, 0xEC, 0x3E,
0xF0, 0xE0, 0x0E, 0xE0, 0xE0, 0x06, 0xC0, 0xE0, 0x0E, 0xE0, 0xF8, 0x1E, 0xF0,
0xFC, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x3F, 0xFC, 0xFF, 0x1F, 0xF8, 0xFF, 0x0F, 0x00,
0x00, 0x00, // Code for char g
0x0C, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0x80,
0x07, 0x00, 0x80, 0x03, 0x00, 0x80, 0x01, 0x00, 0x80, 0x03, 0x00, 0x80, 0xFF,
0x3F, 0x80, 0xFF, 0x3F, 0x00, 0xFF, 0x3F, 0x00, 0xFE, 0x3F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, // Code for char h
0x05, 0x8C, 0xFF, 0x3F, 0x9E, 0xFF, 0x3F, 0x9E, 0xFF, 0x3F, 0x9E, 0xFF, 0x3F, 0x8C,
0xFF, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char i
0x09, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x70, 0x00, 0x00, 0xE0, 0x00, 0x00, 0xC0,
0x00, 0x00, 0xC0, 0xC6, 0xFF, 0xFF, 0xCF, 0xFF, 0xFF, 0xCF, 0xFF, 0x7F, 0xC6, 0xFF,
0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char j
0x0D, 0x00, 0x00, 0x00, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE,
0xFF, 0x3F, 0x00, 0xF8, 0x01, 0x00, 0xFC, 0x03, 0x00, 0xFE, 0x07, 0x00, 0xDF, 0x1F,
0x80, 0x8F, 0x3F, 0x80, 0x07, 0x3E, 0x80, 0x01, 0x38, 0x80, 0x00, 0x30, 0x00,
0x00, 0x00, // Code for char k
0x04, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char l
0x0E, 0xC0, 0xFF, 0x3F, 0xC0, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80, 0x07, 0x00, 0x80,
0x03, 0x00, 0x80, 0x03, 0x00, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80, 0x03, 0x00,
0x80, 0x03, 0x00, 0x80, 0x07, 0x00, 0x80, 0xFF, 0x3F, 0x00, 0xFF, 0x3F, 0x00, 0xFE,
0x3F, // Code for char m
0x0D, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80,
0x07, 0x00, 0x80, 0x03, 0x00, 0x80, 0x01, 0x00, 0x80, 0x03, 0x00, 0x80, 0xFF,
0x3F, 0x00, 0xFF, 0x3F, 0x00, 0xFF, 0x3F, 0x00, 0xFE, 0x3F, 0x00, 0xFC, 0x3F, 0x00,
0x00, 0x00, // Code for char n
0x0E, 0x00, 0xFC, 0x01, 0x00, 0xFF, 0x0F, 0x80, 0xFF, 0x0F, 0xC0, 0xFF, 0x1F, 0xE0,
0x07, 0x3F, 0xE0, 0x01, 0x3C, 0xE0, 0x00, 0x38, 0xE0, 0x00, 0x38, 0xE0, 0x01,
0x3C, 0xE0, 0x07, 0x3F, 0xC0, 0xFF, 0x1F, 0x80, 0xFF, 0x0F, 0x00, 0xFF, 0x07, 0x00,
0xFC, 0x01, // Code for char o
0x0D, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0xFE, 0xFF, 0x7F, 0x1E,
0xF0, 0x00, 0x0E, 0xE0, 0x00, 0x06, 0xC0, 0x00, 0x0E, 0xE0, 0x00, 0x1E, 0xF0,
```

```
0xFF, 0xFF, // Code for char q
0x09, 0xC0, 0xFF, 0x3F, 0xC0, 0xFF, 0x3F, 0xC0, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x00,
0xFF, 0x3F, 0x80, 0x07, 0x00, 0xC0, 0x03, 0x00, 0xC0, 0x01, 0x00, 0xC0, 0x01,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char r
0x0C, 0x00, 0x0F, 0x0C, 0x80, 0x1F, 0x1C, 0x80, 0x3F, 0x3C, 0xC0, 0x3F, 0x38,
0xE0, 0x7F, 0x38, 0xE0, 0x79, 0x38, 0xE0, 0xF0, 0x3C, 0xE0, 0xF0, 0x3F, 0xE0, 0xF1,
0x1F, 0xE0, 0xE1, 0x0F, 0xC0, 0xC0, 0x0F, 0x00, 0x80, 0x07, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char s
0x0B, 0xC0, 0x01, 0x00, 0xC0, 0x01, 0x00, 0xC0, 0x01, 0x00, 0xF8, 0xFF, 0x0F,
0xF8, 0xFF, 0x1F, 0xF8, 0xFF, 0x3F, 0xF8, 0xFF, 0x3F, 0xF8, 0xFF, 0x3F, 0xC0, 0x01,
0x3C, 0xC0, 0x01, 0x38, 0xC0, 0x01, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char t
0x0D, 0x80, 0xFF, 0x07, 0x80, 0xFF, 0x0F, 0x80, 0xFF, 0x1F, 0x80, 0xFF, 0x3F, 0x00,
0x00, 0x38, 0x00, 0x00, 0x30, 0x00, 0x00, 0x30, 0x00, 0x00, 0x38, 0x00, 0x00,
0x3C, 0x80, 0xFF, 0x1F, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x80, 0xFF, 0x3F, 0x00,
0x00, 0x00, // Code for char u
0x0E, 0x80, 0x07, 0x00, 0x80, 0x3F, 0x00, 0x80, 0xFF, 0x01, 0x80, 0xFF, 0x07, 0x00,
0xF8, 0x1F, 0x00, 0xE0, 0x1F, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x3E, 0x00, 0xE0, 0x1F,
0x00, 0xF8, 0x1F, 0x80, 0xFF, 0x07, 0x80, 0xFF, 0x00, 0x80, 0x3F, 0x00, 0x80, 0x07,
0x00, // Code for char v
0x0E, 0x80, 0xFF, 0x00, 0x80, 0xFF, 0x01, 0x00, 0xFF, 0x0F, 0x00, 0xE0, 0x3F, 0x00,
0xC0, 0x3F, 0x00, 0x80, 0x01, 0x80, 0xFF, 0x03, 0x80, 0xFF, 0x03, 0x00, 0x80, 0x01,
0x00, 0xC0, 0x3F, 0x00, 0xE0, 0x3F, 0x00, 0xFF, 0x0F, 0x80, 0xFF, 0x01, 0x80, 0xFF,
0x00, // Code for char w
0x0D, 0x80, 0x01, 0x30, 0x80, 0x01, 0x30, 0x80, 0x03, 0x3C, 0x80, 0x07, 0x3E,
0x00, 0x9E, 0x0F, 0x00, 0xFC, 0x07, 0x00, 0xF8, 0x03, 0x00, 0xFC, 0x07, 0x00,
0x9E, 0x0F, 0x80, 0x07, 0x3E, 0x80, 0x03, 0x3C, 0x80, 0x01, 0x30, 0x80, 0x01,
0x30, 0x00, 0x00, 0x00, // Code for char x
0x0D, 0x78, 0x00, 0xC0, 0xF8, 0x03, 0xE0, 0xF8, 0x1F, 0xF0, 0xF8, 0x3F, 0xF8, 0x00,
0x7F, 0x7C, 0x00, 0xF8, 0x3F, 0x00, 0xF0, 0x1F, 0x00, 0xF8, 0x07, 0x00, 0xFF, 0x01,
0xF8, 0x7F, 0x00, 0xF8, 0x1F, 0x00, 0xF8, 0x03, 0x00, 0x78, 0x00, 0x00, 0x00, 0x00,
0x00, // Code for char y
0x0D, 0xE0, 0x00, 0x38, 0xE0, 0x00, 0x3E, 0xE0, 0x00, 0x3F, 0xE0, 0xC0, 0x3F,
0xE0, 0xF0, 0x3F, 0xE0, 0xF8, 0x3F, 0xE0, 0xFE, 0x3B, 0xE0, 0xFF, 0x38, 0xE0, 0x7F,
0x38, 0xE0, 0x1F, 0x38, 0xE0, 0x07, 0x38, 0xE0, 0x03, 0x38, 0xE0, 0x00, 0x38,
0x00, 0x00, 0x00, // Code for char z
0x09, 0x00, 0x18, 0x00, 0x00, 0x18, 0x00, 0xFC, 0xFF, 0x3F, 0xFE, 0xFF, 0x7F, 0xFF,
0xE7, 0xFF, 0x7F, 0x00, 0xFC, 0x07, 0x00, 0xE0, 0x03, 0x00, 0xC0, 0x03, 0x00,
0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char {
0x04, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char |
0x09, 0x03, 0x00, 0xC0, 0x03, 0x00, 0xC0, 0x07, 0x00, 0xE0, 0x3F, 0x00, 0xF8,
0xFF, 0x03, 0xFF, 0xFE, 0xFF, 0x7F, 0xFC, 0xFF, 0x3F, 0x00, 0x18, 0x00, 0x00, 0x18,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, // Code for char }
0x0E, 0xC0, 0x03, 0x00, 0xE0, 0x03, 0x00, 0x70, 0x00, 0x00, 0x30, 0x00, 0x00,
0x70, 0x00, 0x00, 0xF0, 0x00, 0x00, 0xE0, 0x00, 0x00, 0xC0, 0x01, 0x00, 0xC0,
0x03, 0x00, 0x80, 0x03, 0x00, 0x00, 0x03, 0x00, 0x80, 0x03, 0x00, 0xF0, 0x01,
0x00, 0xF0, 0x00, 0x00, // Code for char ~
0x08, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0x02, 0x00, 0x20, 0x02, 0x00, 0x20, 0x02,
0x00, 0x20, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0xFE, 0xFF, 0x3F, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00 // Code for char
};
```

Share

Share

```
int latchPin = 10;
int clockPin = 13;
int dataPin = 11;
int clock = 9;
int Reset = 8;

int latchPinPORTB = latchPin – 8;
int clockPinPORTB = clockPin – 8;
int dataPinPORTB = dataPin – 8;

int i = 0;
long scrolling_word[24];
int array_turn = 0;

byte your_text[14][24] = {"Fingers cross Hoping it works"}; //Put your text here

//********* First message on the screen **************
void setup() {

for (int address = 0; address 0) { //if message is from virtual bluetooth prot
stg_length = BT.available(); // incoming string length
for (int i = 0; i < stg_length; i++) {
incomingByte = BT.read(); //reading incoming byte
Phrase[i] = incomingByte; //store the character into string array
}
}

void setup() {

Serial.begin(9600);

pinMode (dataPin, OUTPUT);
pinMode (clockPin, OUTPUT);
pinMode (latchPin, OUTPUT);
pinMode (clock, OUTPUT);
pinMode (Reset, OUTPUT);

digitalWrite (Reset, HIGH);
digitalWrite (Reset, LOW);

setupSPI();

}

//*** This function displays the symbols***//

void display_word (int loops, byte word_print[][24], int num_patterns, int
delay_length){

i = 0; //resets the counter for the 4017

for (int g = 0; g < 24; g++) //resets the long int where the word goes

scrolling_word[g] = 0;

for (int x = 0; x< num_patterns; x++) { // main loop goes over symbols

for (int r = 0; r < 24; r++) //puts the first build symbol
```

```
// end of scrolling function

for (int t = 0; t < delay_length; t++) { //delay function, it just loop over the same
display

for (int y = 0; y < 24; y++) { //scanning the dispaly

if (i == 24) { //counting up to 24 with the 4017
digitalWrite (Reset, HIGH);
digitalWrite (Reset, LOW);
i = 0;
}

latchOff();
spi_transfer(make_word(0x01000000,y)); //sending the data
spi_transfer(make_word(0x00010000,y));
spi_transfer(male_word(0x00000100,y));

latchOn();
delayMicroseconds (800); //waiting a bit
latchOff();

spi_transfer(0); //clearing the data
spi_transfer(0);
spi_transfer(0);

latchOn();
digitalWrite (clock, HIGH); // counting up with the 4017
digitalWrite (clock, LOW);
i++
}
}
}
}

byte make_word (long position, byte turn) {
byte dummy_word = 0;

for (int q = 0; q < 14; q++) {
if (scrolling_word[turn] & (position << q))
dummy_word |= 0x01 << q;
}
return dummy_word;
}

void loop() {

//calls for the display_pattern function
display_word(1, your_text,8,15);
}

void latchOn() {
bitSet (PORTB, latchPinPORTB);
}

void latchOff() {
bitClear(PORTB, latchPinPORTB);
}
```

```
SPCR &= ~( (1 << SPR1) | (1 << SPRO) ); // clear prescalar bits
clr = SPSR; // clear SPI status reg
clr = SPDR; // clear SPI data reg
SPSR |= (1 << SPI2X); //set prescaler bits
SPSR &= ~(1 << SPI2X); //clear prescaler bits

delay (10);
}

byte spi_transfer(byte data)
{

SPDR = data; // Start the transmission
while (!(SPSR & (1 << SPIF))) //Wait the end of the transmission
{
};

return SPDR; //return the recieved byte
}
```

Share

Reply

murthy November 25, 2015 4:05 am

If i remove USB connection from computer will it display or not.

Reply

murthy November 25, 2015 4:00 am

Should i connect this circuit to computer contineously for displaying.

Reply

AMINE August 31, 2015 7:31 pm

please i need : code + shema isis arduino uno r3/ 2 registre /matrix 8*8
merci de repondre

Reply

Nik Megan August 20, 2015 11:53 am

Using your hex file works fine but if I re-compile I get nothing displayed.

I also cannot get terminal to display anything.

Please advise

Reply

towsif August 18, 2015 2:29 pm

Dear sir/madam good evening.

Reply

Bibin edmond July 18, 2015 2:59 am

Can anyone please send me the code for 8*24 led matrix using pic16f877a.. there's some problem with mine.. its scrolling but some leds lights up unnecessarily when connecting the third shift register..
Bibinedmond@gmail.com

Reply

towsif August 18, 2015 2:33 pm

sorry you can't because 16F877 has not enough RAM space.

Reply

nesnil rose agruda July 9, 2015 7:44 am

would you mind to enumerate the materials used?

Reply

marto May 11, 2015 3:08 am

Can I get a complete range of 8 × 40 matrix LED display please?
I'm using atmega 8535 and the sliding registers using 74HC595 with LED size 5mm.
please send to:martosby@yahoo.com

Reply

marto May 11, 2015 3:00 am

Bisakah saya mendapatkan rangkaian lengkap 8 × 40 tampilan matriks LED please?
saya menggunakan atmega 8535 dan register gesernya menggunakan 74hc595 dengan ukuran led 5mm.
silakan kirim ke: martosby@yahoo.com

Reply

sumit April 28, 2015 12:45 pm

helo sir..
ThAnks for this project guidance. i have gained alot from this.. n i am preparing a gsm based led moving message display.. so if can please help me out in the programing for this..
i m using 89s52 micro controller and size of our display is 16*62.
please reply if any one has any help for me...

Reply

Rohit March 28, 2015 12:59 pm

harisai March 8, 2015 12:52 pm

plesase sir i want code for 8051 micro controller

Reply

harisai March 8, 2015 12:50 pm

please send me code for 8051 micro controller
please sir i was struggling with this

Reply

zino February 14, 2015 7:33 pm

i would like to replace pic 16f147 by the pic16f877 please can you help me
wana know what to do
thank you

Reply

abdullah April 16, 2015 4:57 am

very good & i like it.

but i want to use pic16f877. is it possible?

Reply

abdullah April 16, 2015 5:03 am

how can i apply pic16f877 & what is the code.

Reply

towsif August 18, 2015 2:35 pm

sorry you can't because 16f877 has not enough RAM space.

Reply

towsif August 18, 2015 2:40 pm

sorry you can't because 16F877 has not enough RAM space.

Reply

towsif August 18, 2015 2:45 pm

you can but you have to remove UART configure commands and only default massage can be display

Dear sir good afternoon.

with due respect I would like to know that can I use PIC16F628A to this project and how to I can send my desired message into EPPROM using pickit2?

please kindly help me as soon as possible.

Thank you

Reply

Share

hari January 10, 2015 11:25 am

sir i want 8*160 display sourse coad

Reply

ROBERTO CARLOS DA SILVA ALVARENGA January 1, 2015 7:02 pm

PLEASE CAN SOMEONE SEND TO ME THE CODE NO PIC USE ONLY WITH DRIVER AND PLATE 74HC595 ARDUINO ....
EMAIL: ROBERTOCARLOSALVARENGA@HOTMAIL.COM
HELP ME
(BRAZIL)

Reply

md .aziz December 5, 2014 3:58 am

sir i want need to usage ic pic 16f72 this display same code of hex sand me please///////////////

Reply

towsif August 18, 2015 2:42 pm

sorry you can't because 16F72 and 16f73 has no RAM space.

you can use 18F2550

Reply

anilkuamr October 8, 2014 9:46 am

hai,i have started my mini project on led cube8*8*8 and i have competed my project but i want one help from you guys.i want source codeof led cube 8*8*8 8051VN so,please help me guys

Reply

Pranab Kumar Choudhury August 28, 2014 4:52 am

Can I get the complete circuit of 8×40 LED matrix display please?

dapelow@gmail.com. Thanks again.

Reply

david July 23, 2014 8:26 am

Someone can send me the code for 8×64 and 8×80. Thanks.

Reply

Share

towsif August 18, 2015 2:47 pm

just increase the buffer routine to add more matrix display

Reply

SIRIYADASA July 7, 2014 1:11 pm

4th comment...

Finally, I have successfully build the 8×48 matrix display. (it can be increase to 8×80)

steps I followed
1. compared original LISTS file and my file . (there were some differences . but the hex files size were the same).
2. Remove the ListenSerial function and recompiled than it worked but only with default MSG..
3. Included the ListenSerial function doing some changes. it worked and can send text to eeprom . I forgot to mention that I disabled many library files. and the hex file size is 12.5kb.

My ListenSerial()

```
void ListenSerial(){

char RxByte;

UART_Write_Text("Start");
UART_Write(10);
UART_Write(13);
UART_Write_Text("INPUT? Y/N");
Delay_onesec();
Delay_onesec();
Delay_onesec();
Delay_onesec();
Delay_onesec();

if (UART_Data_Ready()) //If data is received,
{
RxByte = UART_Read(); //read the received data,
if(RxByte == 'Y') {
UART_Write(RxByte);
while (!UART_Data_Ready()); // if data is received
UART_Read_Text(message, "#", 250); // reads text until '#' is found
```

}
}

I Think there must be some problem with the windows os (I am using xp home edition.)
or damaged Mirokc file.

Who can a give answer ... I released that nothing to do any changes to the original code . except increasing buffer !!! ???.

siri

Share

Reply

PRABA September 25, 2014 2:58 am

please send your 8×48 program to my email.id(prabakaran_1991@rocketmail.com). i am struggling lot

Reply

DON March 31, 2015 11:40 am

Please can you help me send 8by 48 , the source code and circuit diagram.. I'm using mikro C compiler thanks a lot

Reply

DON March 31, 2015 11:44 am

gaullyubong@yahoo.com.....please send me message display code and circuit diagram using pic16f877a, I'm using mikro C pro

Reply

ganesh January 12, 2018 3:16 am

please bro send your 8×48 program to me
My mail address:-
ganeshpkundapur@gmail.com

Reply

ADEYEYE Kayode July 4, 2014 11:35 am

I want to use 74LS164

Reply

4S7 December 3, 2014 5:41 am

I like this 8×8 LED display systems, and how to do with connected to PC.
Can some one teach me.

MUIGAI can I have your source codes for PIC16f88... plz

My Email ID siriyadasa.umagiliya@gmail.com

Reply

SIRIYADASA June 30, 2014 1:57 am

3rd ... comment
Mr; R.B
Sir,
As I mentioned in my 2nd comment the 8×40 matrix display is working very nicely.
But when I recompiled your c source file the output gave me 14.4kb hex file. it is not working,no display, not even one led glows. __config set according to yours. Are there any changes to be done? if it is ... plz send me the codes (because I want to change it to 8×48)

My Email ID siriyadasa.umagiliya@gmail.com

Reply

SIRIYADASA June 26, 2014 9:00 am

Sir,

This is regarding the comment given by me on June 24,2014 . The problem was with the usb cable. one wire was disconnected (dry joint) replaced with new one . Now I can send text to eeprom .

R.B. Excellent project

Siri

Reply

SIRIYADASA June 24, 2014 6:12 am

Sir ,
I build the 8×40 led matrix using pic16f1847 and programed with your HEX file. It is working
without any problem(Displaying the default message) When I try to send text data to eeprom
(using USB to serial adapter – FT232RL with termite SW) I got only [0][0][0]..... similar to
to comment given by "Prabhat – September – 11- 2013" plz help me.

siri

Reply

ilker June 13, 2014 5:16 pm

excellent tutorial with a great explanation thank you. i have succesfully managed to adopt your code for 16f877a with an 8×72 led matrix display and removed the uart part -for now but will add later.. PROTEUS can't simulate correctly but the circuit is working flawless.

Hi ilker.. can u share with me how to change the code for 16f877a?? i would like to learn from u.. i just need a guide to do that.. doesn't have to send me the full code that u have done by urself.. can u teach me? please email me at azinmil@yahoo.com .. thanks.. ^^

Reply

sharifu rajabu May 31, 2014 3:01 am

can u help me on what is wrong with this code. am using pic16f452. the program compile successfully. but when i simulate it in Proteus all the LED light up.

```
sbit Serial_Data at RA0_bit;
sbit SH_Clk at RA1_bit;
sbit ST_Clk at RA2_bit;

unsigned short Buffer[8][5] = {{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0},
{0,0,0,0,0}};

void Send_Data(unsigned short rw)
{
unsigned short Mask, t, num, Flag;
for (num = 0; num < 5; num++)
{
Mask = 0x01;
for (t=0; t<8; t++)
{
Flag = Buffer[rw][num] & Mask;
if(Flag==0)
Serial_Data = 0;
else
Serial_Data = 1;
SH_Clk = 1;
SH_Clk = 0;
Mask = Mask << 1;
}
}

ST_Clk = 1;
ST_Clk = 0;
}

const unsigned short CharData[][8] ={
{0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000000},
{0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100,
0b00000000, 0b00000100},
{0b00001010, 0b00001010, 0b00001010, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000000},
{0b00000000, 0b00001010, 0b00011111, 0b00001010, 0b00011111, 0b00001010,
```

```
{0b00000110, 0b00001010, 0b00010010, 0b00010100, 0b00001001, 0b00010110,
0b00010110, 0b00001001},
{0b00000100, 0b00000100, 0b00000100, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000000},
{0b00000010, 0b00000100, 0b00001000, 0b00001000, 0b00001000, 0b00001000,
0b00000100, 0b00000010},
{0b00001000, 0b00000100, 0b00000010, 0b00000010, 0b00000010, 0b00000010,
0b00000100, 0b00001000},
{0b00010101, 0b00001110, 0b00011111, 0b00001110, 0b00010101, 0b00000000,
0b00000000, 0b00000000},
{0b00000000, 0b00000000, 0b00000100, 0b00000100, 0b00011111, 0b00000100,
0b00000100, 0b00000000},
{0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000110,
0b00000100, 0b00001000},
{0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00001110, 0b00000000,
0b00000000, 0b00000000},
{0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000100},
{0b00000001, 0b00000010, 0b00000010, 0b00000100, 0b00000100, 0b00001000,
0b00001000, 0b00010000},
{0b00001110, 0b00010001, 0b00010011, 0b00010001, 0b00010101, 0b00010001,
0b00011001, 0b00001110},
{0b00000100, 0b00001100, 0b00010100, 0b00000100, 0b00000100, 0b00000100,
0b00000100, 0b00011111},
{0b00001110, 0b00010001, 0b00010001, 0b00000010, 0b00000100, 0b00001000,
0b00010000, 0b00011111},
{0b00001110, 0b00010001, 0b00000001, 0b00001110, 0b00000001, 0b00000001,
0b00010001, 0b00001110},
{0b00010000, 0b00010000, 0b00010100, 0b00010100, 0b00011111, 0b00000100,
0b00000100, 0b00000100},
{0b00011111, 0b00010000, 0b00010000, 0b00011110, 0b00000001, 0b00000001,
0b00000001, 0b00011110},
{0b00000111, 0b00001000, 0b00010000, 0b00011110, 0b00010001, 0b00010001,
0b00010001, 0b00001110},
{0b00011111, 0b00000001, 0b00000001, 0b00000001, 0b00000010, 0b00000100,
0b00001000, 0b00010000},
{0b00001110, 0b00010001, 0b00010001, 0b00001110, 0b00010001, 0b00010001,
0b00010001, 0b00001110},
{0b00001110, 0b00010001, 0b00010001, 0b00001111, 0b00000001, 0b00000001,
0b00000001, 0b00000001},
{0b00000000, 0b00000100, 0b00000100, 0b00000000, 0b00000000, 0b00000100,
0b00000100, 0b00000000},
{0b00000000, 0b00000100, 0b00000100, 0b00000000, 0b00000000, 0b00000100,
0b00000100, 0b00001000},
{0b00000001, 0b00000010, 0b00000100, 0b00001000, 0b00001000, 0b00000100,
0b00000010, 0b00000001},
{0b00000000, 0b00000000, 0b00000000, 0b00011110, 0b00000000, 0b00011110,
0b00000000, 0b00000000},
{0b00010000, 0b00001000, 0b00000100, 0b00000010, 0b00000010, 0b00000100,
0b00001000, 0b00010000},
{0b00001110, 0b00010001, 0b00010001, 0b00000010, 0b00000100, 0b00000100,
0b00000000, 0b00000100},
{0b00001110, 0b00010001, 0b00010001, 0b00010101, 0b00010101, 0b00010001,
0b00010001, 0b00011110},
{0b00001110, 0b00010001, 0b00010001, 0b00010001, 0b00011111, 0b00010001,
0b00010001, 0b00010001},
{0b00011110, 0b00010001, 0b00010001, 0b00011110, 0b00010001, 0b00010001,
0b00010001, 0b00011110},
{0b00000111, 0b00001000, 0b00010000, 0b00010000, 0b00010000, 0b00010000,
```

```
{0b00011111, 0b00010000, 0b00010000, 0b00011110, 0b00010000, 0b00010000,
0b00010000, 0b00010000},
{0b00001110, 0b00010001, 0b00010000, 0b00010000, 0b00010111, 0b00010001,
0b00010001, 0b00001110},
{0b00010001, 0b00010001, 0b00010001, 0b00011111, 0b00010001, 0b00010001,
0b00010001, 0b00010001},
{0b00011111, 0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100,
0b00000100, 0b00011111},
{0b00011111, 0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100,
0b00010100, 0b00001000},
{0b00010001, 0b00010010, 0b00010100, 0b00011000, 0b00010100, 0b00010010,
0b00010001, 0b00010001},
{0b00010000, 0b00010000, 0b00010000, 0b00010000, 0b00010000, 0b00010000,
0b00010000, 0b00011111},
{0b00010001, 0b00011011, 0b00011111, 0b00010101, 0b00010001, 0b00010001,
0b00010001, 0b00010001},
{0b00010001, 0b00011001, 0b00011001, 0b00010101, 0b00010101, 0b00010011,
0b00010011, 0b00010001},
{0b00001110, 0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001,
0b00010001, 0b00001110},
{0b00011110, 0b00010001, 0b00010001, 0b00011110, 0b00010000, 0b00010000,
0b00010000, 0b00010000},
{0b00001110, 0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010101,
0b00010011, 0b00001111},
{0b00011110, 0b00010001, 0b00010001, 0b00011110, 0b00010100, 0b00010010,
0b00010001, 0b00010001},
{0b00001110, 0b00010001, 0b00010000, 0b00001000, 0b00000110, 0b00000001,
0b00010001, 0b00001110},
{0b00011111, 0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100,
0b00000100, 0b00000100},
{0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001,
0b00010001, 0b00001110},
{0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001,
0b00001010, 0b00000100},
{0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010001, 0b00010101,
0b00010101, 0b00001010},
{0b00010001, 0b00010001, 0b00001010, 0b00000100, 0b00000100, 0b00001010,
0b00010001, 0b00010001},
{0b00010001, 0b00010001, 0b00001010, 0b00000100, 0b00000100, 0b00000100,
0b00000100, 0b00000100},
{0b00011111, 0b00000001, 0b00000010, 0b00000100, 0b00001000, 0b00010000,
0b00010000, 0b00011111},
{0b00001110, 0b00001000, 0b00001000, 0b00001000, 0b00001000, 0b00001000,
0b00001000, 0b00001110},
{0b00010000, 0b00001000, 0b00001000, 0b00000100, 0b00000100, 0b00000010,
0b00000010, 0b00000001},
{0b00001110, 0b00000010, 0b00000010, 0b00000010, 0b00000010, 0b00000010,
0b00000010, 0b00001110},
{0b00000100, 0b00001010, 0b00010001, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000000},
{0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00011111},
{0b00001000, 0b00000100, 0b00000000, 0b00000000, 0b00000000, 0b00000000,
0b00000000, 0b00000000},
{0b00000000, 0b00000000, 0b00000000, 0b00001110, 0b00010010, 0b00010010,
0b00010010, 0b00001111},
{0b00000000, 0b00010000, 0b00010000, 0b00010000, 0b00011100, 0b00010010,
0b00010010, 0b00011100},
{0b00000000, 0b00000000, 0b00000000, 0b00001110, 0b00010000, 0b00010000,
```

```
{0b00000000, 0b00000011, 0b00000100, 0b00000100, 0b00000110, 0b00000100,
0b00000100, 0b00000100},
{0b00000000, 0b00001110, 0b00001010, 0b00001010, 0b00001110, 0b00000010,
0b00000010, 0b00001100},
{0b00000000, 0b00010000, 0b00010000, 0b00010000, 0b00011100, 0b00010010,
0b00010010, 0b00010010},
{0b00000000, 0b00000000, 0b00000100, 0b00000000, 0b00000100, 0b00000100,
0b00000100, 0b00000100},
{0b00000000, 0b00000010, 0b00000000, 0b00000010, 0b00000010, 0b00000010,
0b00000010, 0b00001100},
{0b00000000, 0b00010000, 0b00010000, 0b00010100, 0b00011000, 0b00011000,
0b00010100, 0b00010000},
{0b00000000, 0b00010000, 0b00010000, 0b00010000, 0b00010000, 0b00010000,
0b00010000, 0b00001100},
{0b00000000, 0b00000000, 0b00000000, 0b00001010, 0b00010101, 0b00010001,
0b00010001, 0b00010001},
{0b00000000, 0b00000000, 0b00000000, 0b00010100, 0b00011010, 0b00010010,
0b00010010, 0b00010010},
{0b00000000, 0b00000000, 0b00000000, 0b00001100, 0b00010010, 0b00010010,
0b00010010, 0b00001100},
{0b00000000, 0b00011100, 0b00010010, 0b00010010, 0b00011100, 0b00010000,
0b00010000, 0b00010000},
{0b00000000, 0b00001110, 0b00010010, 0b00010010, 0b00001110, 0b00000010,
0b00000010, 0b00000001},
{0b00000000, 0b00000000, 0b00000000, 0b00001010, 0b00001100, 0b00001000,
0b00001000, 0b00001000},
{0b00000000, 0b00000000, 0b00001110, 0b00010000, 0b00001000, 0b00000100,
0b00000010, 0b00011110},
{0b00000000, 0b00010000, 0b00010000, 0b00011100, 0b00010000, 0b00010000,
0b00010000, 0b00001100},
{0b00000000, 0b00000000, 0b00000000, 0b00010010, 0b00010010, 0b00010010,
0b00010010, 0b00001100},
{0b00000000, 0b00000000, 0b00000000, 0b00010001, 0b00010001, 0b00010001,
0b00001010, 0b00000100},
{0b00000000, 0b00000000, 0b00000000, 0b00010001, 0b00010001, 0b00010001,
0b00010101, 0b00001010},
{0b00000000, 0b00000000, 0b00000000, 0b00010001, 0b00001010, 0b00000100,
0b00001010, 0b00010001},
{0b00000000, 0b00000000, 0b00010001, 0b00001010, 0b00000100, 0b00001000,
0b00001000, 0b00010000},
{0b00000000, 0b00000000, 0b00000000, 0b00011111, 0b00000010, 0b00000100,
0b00001000, 0b00011111},
{0b00000010, 0b00000100, 0b00000100, 0b00000100, 0b00001000, 0b00000100,
0b00000100, 0b00000010},
{0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100, 0b00000100,
0b00000100, 0b00000100},
{0b00001000, 0b00000100, 0b00000100, 0b00000100, 0b00000010, 0b00000100,
0b00000100, 0b00001000},
{0b00000000, 0b00000000, 0b00000000, 0b00001010, 0b00011110, 0b00010100,
0b00000000, 0b00000000}
};

unsigned int speed, StringLength;
unsigned short i, l, k, row, scroll, temp, shift_step=1;
unsigned short m;

char message[]="SHARIFU RAJABU SCROLLING TEXT ON DOT MATRIX";
char index;
```

Share

```
StringLength = strlen(message) ;
do
{
for (k=0; k<StringLength; k++)
{
for (scroll=0; scroll<(8/shift_step); scroll++)
{
for (row=0; row<8; row++)
{
temp = CharData[index-32][row];
Buffer[row][4] = (Buffer[row][4] <> (8-Shift_Step));
Buffer[row][3] = (Buffer[row][3] <> (8-Shift_Step));
Buffer[row][2] = (Buffer[row][2] <> (8-Shift_Step));
Buffer[row][1] = (Buffer[row][1] <> (8-Shift_Step));
Buffer[row][0] = (Buffer[row][0] <> ((8-shift_step)-scroll*shift_step));
}
speed = 15;
for(l=0; l<speed;l++)
{
m = 1;
for (i=0; i<8; i++)
{
Send_Data(i);
PORTB = m;
m = m << 1;
Delay_us(1000);
} // i
} // l
} // scroll
} // k
} while(1);
}
```

Reply

hakim May 5, 2014 12:57 am

is it possible to change the PIC16f1847 to pic16f877a?

Reply

hakim May 5, 2014 12:46 am

is it possible to change the PIC to 16f877a?

Reply

towsif August 19, 2015 4:14 am

no you can't because 16F877 has not enough RAM space.

Reply

```
LATB = m;
m = m << 1;
Delay_ms(20);
}
```

in the above code why you sent i value to the Send_Data() function
by send i value to the Send_Data() function , i am got no output.
why u not sent Buffer[ ][ ] value to the Send_Data() function.
sir please i need your help

Reply

Share

Ramalingm May 2, 2014 5:39 am

please sent the code to mail id ramalingaqmr0@gmail.com
sorry for poor engilish

Reply

hiran April 21, 2014 1:10 pm

sir,
I want this project changing the cirxuit16f877 pic,how?

Reply

ACHU KARSTEN ACHIRI April 15, 2014 2:01 pm

SIR, I AM REALLY INTERESTED IN THE PROJECT BUT I CAN HAVE THE EXACT PIC 18F1847. I WILL BE INDEED GREAT IF YOU CAN HELP ME WITH THE EQUIVALENT.

Reply

abubaker makki April 5, 2014 9:33 am

dear sir\
i want to use pic16f877a and my project is not work properly please help me and you can send to me protues simulation and micro c code

Reply

Naveen Sehra April 4, 2014 2:51 am

very helpful.....but need some improvement.....????????

Reply

manik March 26, 2014 10:49 am

sir,
I need a 16×32 led dot matrix basic code using shift resister code.
plz help me.

manik talukder

Reply

SILK February 8, 2014 2:13 pm

Great project , i make and i love it , how increasing speed of scrolling ? What I have to change the firmware to increase speed, what value should be changed ?

Reply

Share

MUIGAI January 30, 2014 11:47 am

Great project with good explanation. i have been able to implement it using pic16f88 and can view the default message. However, i would like to interface it with a keyboard directly without using a pc. I would appreciate your help. thanks.

Reply

ALEN March 17, 2014 12:27 am

Hello MUIGAI can i have ur sample codes of interfacing using PIC16f88...

Reply

WILBERT February 22, 2015 6:27 pm

Hie ALEN.Can you please help me on how to design and implement an led matrix display.it will be much helpfull if you provide me with the circuit diagrams and source code in mikroC.Thank you

Reply

Joseph Weir January 30, 2014 2:08 am

Can I also extend this to say 8×96 display for my led message sign?

Reply

M.Junaid January 10, 2014 8:50 am

Hello Every one!
It is very useful knowledge about dot matrix displays but I want to developed a project basically a schedule board in which i want to use 128 dot matrix(8 for rows & 16 for columns) so i want to ask that can i use the same logic to control such a large number of LED dot matrix? Is it a reliable technique? or Is there any alternate method?

Reply

sawssan January 2, 2014 4:52 pm

i will very recognise to any one helps me

Reply

p.Voesten January 1, 2014 2:11 pm

Is it posseble to program the Pic 16F1847 with pickit2 i cante find them in the list with supportet pic
ps sorry for the bat englisch

Reply

Share

indicates December 9, 2013 8:34 am

which microcontroller we can use instead of 16f1847.

Reply

Rahat November 30, 2013 6:38 am

Could u please send me a sample code for 40*240. Only for shift a word, any word. It will be very helpful for me.

Reply

Aayush November 12, 2013 10:57 am

how to store the message in the microcontroller inspite of inserting it through pc???
i want to build this plz hlp me....

Reply

alireza November 12, 2013 12:32 am

hi
the hex file is 14.4kb and Program Memory for PIC16F1847 is 14kb
is it possible program this micro?

Reply

PIDOU November 11, 2013 3:40 am

Hello,

I can get another pattern of modifying the function :

Buffer[row][4] = (Buffer[row][4] <> (8-Shift_Step));

Best regards

Sébastien

Reply

mahesh j November 3, 2013 9:33 am

R-B November 3, 2013 10:12 pm

I am using PICKit2.

Reply

mahesh j November 7, 2013 8:03 am

Does the 3rd and 4th ice should connect to ra1 and ra2?

Reply

mahesh j November 3, 2013 3:16 am

Does the 11th and 12th pins of 3rd,4th and 5th ic's should be connected to ra2 and ra1?

Reply

mahesh j shetty November 3, 2013 3:08 am

In the circuit diagram the first two 74hc595 IC's 11th and 12th pins are connected to st_cp(ra2) and sh_cp(ra1) pins.does the 3rd,4th and 5th ic's 11th and 12th pins also have to be connected to ra2 and ra1?

Reply

sam November 1, 2013 12:49 pm

hi
tanks for your best project. how i can used more than 8 matrix LED. because when i want expansion this project to 13 matrix LED but the project not working.! please guide me

regards
Sam

Reply

prabhat September 28, 2013 5:57 am

when ever i burn your project hex code onto my 16f1847 mcu using pic2 programmer i'm only getting [00][00][00][00][00][00][00][00] [00][00][00][00][00][00][00][00][00][00][00][00] continually at hyper terminal. can you plz help me to short out wher is the problem.my mail id is prabhat4mail@gmail.com

Reply

raffy September 23, 2013 6:08 am

hi...sir pls send me the code using pic16f877A:-) thanks ang God bless..here's my email raffy_102488@yahoo.com

Led Dot matrix.Then what will be the program and hw to modify it? The message I want to print in the Led matrix is "Mahadi \ Mobile num.,address etc".

I want that "Mahadi" should scroll from left, scroll from up and down after it stay in the led matrix display for few second and bliking.
After that the message "Mobile phn num and address" should scroll from left.
And that way it will continue.
Plz send me the Program code to my email address.
(mahadihasan102@gmail.com)
plz help me.I waiting for it.

Reply

Mahadi September 21, 2013 12:38 am

I want to make different animation with message with your 8X40 Led Dot matrix.Then what will be the problem and hw to modify it? The message I want to print in the Led matrix is "Mahadi \ Mobile num.,address etc".

I want that "Mahadi" should scroll from left, scroll from up and down after it stay in the led matrix display for few second and bliking.
After that the message "Mobile phn num and address" should scroll from left.
And that way it will continue.
Plz send me the source code to my email address.
(mahadihasan102@gmail.com)

Reply

Electron September 15, 2013 2:11 am

Is the same logic apply, for UP and Down Scrolling?

Reply

Dhaval December 16, 2015 12:38 pm

Hi bro plz send me full project with circuit,Code and description
My email address
dhavalpanchal14590@gmail.com

Reply

Prabhat September 13, 2013 2:28 pm

I'm trying to recompile your c file but it is giving. is there anything to change in the program.....plz help me Rajendra Bhatt.my email id is prabhat4mail@gmail.com

Reply

R-B September 13, 2013 11:10 pm

What error are you getting?

mail id is prabhat4mail@gmail.com

Reply

Prabhat September 11, 2013 3:13 am

PLZ help me, when ever i'm trying your project i'm getting "[00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00][00]" this only at the termite as output. where is the problem?

Reply

Share

mahesh j August 29, 2013 12:55 pm

Please send me the hex file for 8*80 led matrix and my email is maheshjshetty@gmail.com

Reply

Marc August 28, 2013 11:23 am

Let's say i want to print the character degree, how can i do that?

thank you!
marC:)

Reply

nav August 17, 2013 6:40 am

hello!
Alawode Basit. Pls could you send me the code for 8*80.i too have it as my project.please mail me to navjn22@gmail.com.
thanks in advance!

Reply

Varga Lajos August 14, 2013 8:27 am

Hello!
Someone could send 8×48 ledmatrix program!
Thank you!
lajos1969@citromail.hu

Reply

Balaji August 11, 2013 9:44 am

nice work sir,can you send me code for pic18f4520 in micro c or let me know what changes should i make in given code

Reply

Prabhat August 5, 2013 12:12 am

Can you send the program to my mail id....my mail id is prabhat4mail@gmail.com

please help with the code since it is not working after recompiling.

Reply

suju August 2, 2013 4:23 am

Sir,
in proteus, it's not working.I tried it with PIC16f877a with some code modification(avoid UART).can u help me?

Reply

raghab ben hammouda March 22, 2016 1:10 pm

dear sir
i want to use pic16f877a and This project is about constructing a mono-color LED matrix display board that consists 320 LEDs arranged in 8 rows and 40 columns. The heart of this project is PIC16F877 microcontroller please
can you help me

Reply

Nikhil August 1, 2013 5:18 pm

Please help! Hex given is working properly but if I compile it (without any changes) then it does not work. Is there any correction to be made in the code? Also i have observe that on compiling it gives warning that variable Flag & Rxbyte are eliminated by optimizer (i tried dissabling the optimizer but no luck) and boundrate 115942 error 0.64%. what to do. Please give me working code. My email id is nikhilpnp@yahoo.com

Reply

Guillermo Lamana July 31, 2013 8:40 am

Excelente proyecto. Felicitaciones

Reply

buannih July 18, 2013 11:56 am

thanks a lot for this great tutorial. U'r a blessing to so many lives, families, nations.. U can't tell how farl this tutorial can go in helping people.. Please keep up the good work [+ I wish I could gift a reward to u]

Reply

Alawode Basit July 17, 2013 8:32 am

PRAGATHEESWARN.V, sorry for the late response, I"ll send it soon.

prakatheeshece@gmail.com

Reply

Electron July 3, 2013 12:38 pm

it can be scrolled from left to right or vice versa at the same time., please guide

Reply

Share

nasuha July 1, 2013 5:04 am

can i use file .hex PIC16F1847 for PIC16F84A-04/P sir ? please help me... thx

Reply

de silva June 12, 2013 11:26 am

hello sir,

onley ican fined PIC16F88-PDIP micro chip in my place.please guide me to match the pinout with the circuit diagram.if possible send me more information.(desilva_7@yahoo.com)
thanks,

Reply

Ola June 11, 2013 3:40 pm

@Alawode Basit. Thank you very much. I have seen it. Am gonna try it out.

Reply

Alawode Basit June 11, 2013 9:37 am

@ Ola,ok. I've done just that. Check your mail.

Reply

Ola June 9, 2013 4:09 pm

@ Alawode Basit, thanks so much for the reply. I have checked my mail but its not there. Pls maybe you should help me through this one (olaide4real1@yahoo.com) . Thanks.

Reply

Alawode Basit June 9, 2013 1:20 am

Ola. check your mail, I've done that

Alawode Basit. Pls could you send me the code for 8*80. My gmail is olaide4real86@gmail.com. Thanks.

Reply

Alawode Basit June 5, 2013 5:18 pm

@ de silva, of course, any other pic chip can be used. Just make sure that the pic supports uart and make sure you alter the configuration part of the code so that it will match that of the chip you are using. Check you chip's datasheet for configuration information.

Reply

ALEN April 3, 2014 3:14 am

Hello Basit Could you send me the codes for the 8×80 inclusive of the UART to kkaalan0000@yahoo.com, can i use it for GSM receive message??

Reply

WILBERT February 22, 2015 6:31 pm

Alen sir.can you please help me on this project.am interested but am stuck on early stage

Reply

Alawode Basit June 5, 2013 5:14 pm

@ Olayide, i think the code should work for 8×96 display, you just need to alter the code a bit. I altered mine such that it worked for 8×80 display which i think it will also work for yours. Thanks.

Reply

imran June 5, 2013 2:50 am

very nice project thanks...sir how can we attach gsm module with this project can u plz modified code for gsm based led notice board.

Reply

Alawode Basit May 31, 2013 3:29 am

I modified the code for 8×80 display and it worked perfectly. Thanks.

Reply

olayide May 28, 2013 5:42 am

Nice work. May God bless you with your Knowledge. pls can this code work for 8*96 display. thanks

can I use any other pic chip for this project.

Reply

de silva May 25, 2013 10:48 pm

hello sir
how can I make usb relay control board (software)

Reply

Share

Alawode Basit May 25, 2013 5:23 pm

Hi Raj, I've tried it out and I must comment that its really great. But, i found out that there is similarity between this and the one of lab 15 where you declare Displaybuffer[8][16] and here, you declare Buffer[8][5] where the [5] is of type byte. Also, the scrolling code looks technically the same.
I want to ask why you didn't declare the buffer as Buffer[8][40] and also can the same procedure be used for more than 5 8×8 displays? Thanks.

Reply

Marc p May 24, 2013 12:22 pm

Thank you Raj

Reply

Alawode Basit May 23, 2013 4:33 am

Nice work. I'll try this out. Can I also extend this to say 8×96 display?

Reply

kiani May 22, 2013 11:30 pm

hi i need proteus simulation of this. will u send me ??

Reply

sahu May 21, 2013 1:27 pm

nice very nice project
how can we use HINDI langveg ?

Reply

Pingback: LED Matrix Scrolling Marquee using PIC MCU and Shift Registers | Arduino collector blog

Reply

kabir May 19, 2013 11:04 pm

I need your email add. I want to talk with u for another project, please feed back at prime.2000@yahoo.com

Reply

kabir May 19, 2013 10:57 pm

Great job. I like it very much

Reply

de May 19, 2013 10:48 pm

whthout programmer can i use this?

Reply

Pingback: LED Matrix Scrolling Marquee using PIC MCU and Shift Registers | Make, Electronics projects, electronic Circuits, DIY projects, Microcontroller Projects - makeelectronic.com

Marc P. May 19, 2013 11:26 am

i don't want to use this with my pc? only with 5vdc adapter is it possible without UART?

thank you!

marC:)

Reply

R-B May 19, 2013 12:04 pm

Yes, you can. When you don't use UART it plays the prestored message.

Reply

muhammad ahad April 19, 2016 8:31 am

plz sir snd me the coding of message display on dot matrix plz plz

Reply

Marc P. May 19, 2013 11:23 am

Awesome project!!! 🙂

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Post Comment

Share

Error: Invalid Affiliate