Compartilhar Denunciar abuso Próximo blog» Criar um blog Login

myProject: C51, ARM7, Cortex-M3

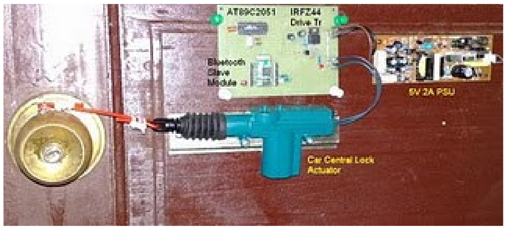
My microcontroller projects commonly using AT89S5X, AT89CX051, LPC21XX (ARM7), STM32F103RxT6 and LPC1768 (Cortex-M3). If you compare microcontrollers in 90s and today, there are so many progress has been made. So keep an eye on the evolution in the exciting microcontroller world.

Main page | Board | Left unused | About me

Oct 29, 2010

Mobile Phone Bluetooth Door Unlock

This module is HC-06 a slave mode serial bluetooth data link. The chips is made by CSR. In this project, my mobile phone will talk to AT89C2051 via HC-06. All the complication has been put in the cage (in the java library and bluetooth module), so what I need to do is just create terminal program for mobile phone written in java (J2ME), and create another program to receive the unlock code for AT89C2051.



This is the result of the project, looks a little ugly but it works. The motor is spring return type therefore when the code sent by mobile phone is correct, the motor energized for 300ms and return to original position by itself. The motor is designed to operate with 12V power but in fact 5V is sufficient to operate it with softer action.

Here the schematic. HC-06 can be setup using AT command. From the factory the default parameters are 9600, N, 8, 1 and 1234 password. Connect the HC-06 module to a PC using USB/TTL then open a terminal software. First you can change the name of the module by command: AT + NAMEnewname (max 20 chars), the reply is OKname. Second you can change the speed by command: AT + BAUD4. The reply is OK9600.
Baud table:
BAUD1---1200
BAUD2---2400
BAUD3---4800
BAUD4---9600
BAUD5---19200
BAUD6---38400
BAUD7---57600
BAUD8---115200
BAUD9---230400
BAUDA---460800

Blog Archive

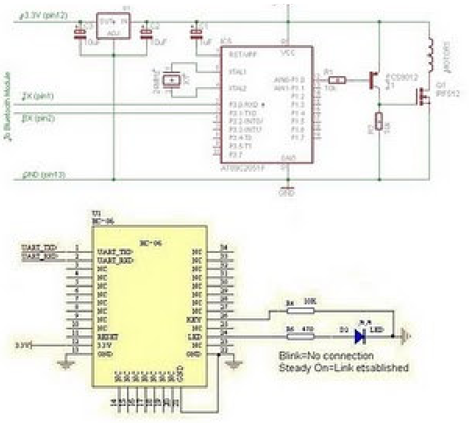
- February (2)
- December (5)
- November (3)
- October (4)
- September (6)
- August (2)

Labels

AT89C2051 (4) STM32F103RBT6 (4) Cortex-M3 (3) LPC 2136 (3) 1 Farad (2) AT89S 51 (2) DS1302 (2) Dot matrix (2) 125kHz (1) 128x64 LCD (1) 320x 240 LCD (1) 4-20mA (1) 433MHz ASK (1) 74HC595 (1) CNC Router (1) CSR HC-06 (1) Christmas Day (1) DC Voltmeter (1) Eagle (1) Flash Loader Demonstrator (1) Flash Magic (1) ILI 9325 (1) Itron Noritake VFD (1) J2ME (1) LPC1768 (1) LPC2103 (1) LPC 2148 (1) Potentiometer (1) RFID (1) SIRF-II GPS (1) STM32F103R8T6 (1) STM32F103VET6 (1) Signal generator (1) Top audience (1) USB HID (1) eV 527 (1) mbed prototyping board (1)

Links

http://www.atmel.com/
http://mbed.org/
http://www.st.com/
http://www.nxp.com/
http://www.pjrc.com/
http://www.friendlyarm.de/
http://www.datasheet4u.com/
http://gandalf.arubi.uni-



Third, you can change the pin by command: AT + PINxxxx, the reply is OKsetpin. The PIN is security code for making connection with the bluetooth module. Please note that those command can't be typed in the terminal software, but you must copy and paste it instead. The module will ignore the command if there is a "long silence" between the characters.

What we try to avoid when using mobile phone as a key is the slowness. Imagine if we have to turn on the bluetooth, scan the device, enter the password, send the code to open the door, waiting for the response.... It takes "million years". We must try to make everything automatic. By openning the java application, the program will take care the rest and giving the report that the door has been unlocked. Total time must be less than 5 seconds.

Java code for making the connection

```
public void Connect ()
    {
        try
        {   URL = "001005280017"; //the mac address of the slave bluetooth module
            connection = (StreamConnection) Connector.open ("btspp://"+URL+":1",
Connector.READ_WRITE);
            reader = new InputStreamReader(connection.openInputStream());
            dos = connection.openOutputStream();
            ow = new OutputStreamWriter(dos, "iso-8859-1");
            //
        }
        catch (IOException ex)
        {
            ex.printStackTrace();
        }
    }
```

Java code for sending the code to unlock the door

```
try
     {
         ow.write("012345"+"\r"); //the unclock code is 012345
         ow.flush();
     }
     catch (IOException ex)
     {
         ostream.write("No connection"+"\n");
         ostream.flush();
     }
```

C code for AT89C2051 mcu

```
main( )
{

    serial_init();
    while(1)
    {
        read_1char = serial_getc();
        if (read_1char ==13) //wait for the CR character
        {
     if ((received_key[5]=='0')
     && (received_key[4]=='1')
     && (received_key[3]=='2')
     && (received_key[2]=='3')
```

kl.de/avr_projects/arm_projects/
http://www.iteadstudio.com/
http://www.futurlec.com.au/
http://www.ni.com/
http://www.elektor.com/
http://www.elm-chan.org/
http://www.satistronics.com/
http://www.yagarto.de/
http://www.kmitl.ac.th/~kswichit/
http://izea.in/rpmg

Pageviews

15,870

Subscribe To

Share it

Share this on Facebook
Tweet this
Get more gadgets for your site

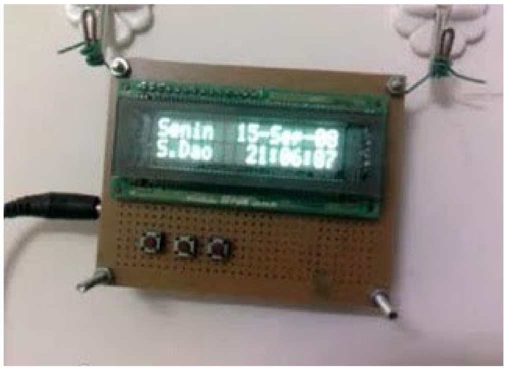
```
    {
         serial_puts("Please comein..."); //if the code match 012345, send reply
         serial_putc(13);
         nMotor=0;                        //turn on the motor (0=ON)
         delayms(300);                    //for 0.3second
         nMotor=1;                        //turn off the motor (1=OFF)
    }
  }
  else
  {
     for (i=8; i>0; i--) received_key[i]=received_key[i-1];
     received_key[0]=read_1char;  //shift the received char in character array
  }

    }
}
```

Links to this post

7 comments

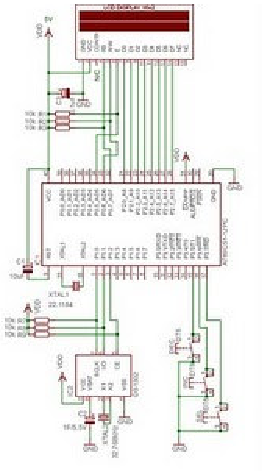
Oct 24, 2010

VFD 16 Chars 2 Lines Digital Clock



VFD (Vacuum Fluorescent Display) just like fluorescent lamp it glows in the dark giving a very high contrast between the characters and its background. In early years, user should prepare the special power supply for the display (12V) and the heater (1.5V) but Itron Noritake CU16025ECPB-W6J do it all, it has been integrated into the module, just plug a 5V supply to it and it works. In fact, the contrast potentiometer is not necessary to use anymore because the contrast setting done in the software.

As beside picture, the circuit is made of a VFD, C51 MCU, DS3021 time keeper, 1F capacitor, high quality 32.768kHz crystal, and push buttons. You can buy a $1 digital clock with alarm, room temperature indication, color backlight from China but it doesn't give you any exciting feeling. That's the reason why the hobbyist like me doing projects, it's a kind of exercise for the brain, at the end of the projects you'll get some "muscles" in your brain, giving you a healthy and happy life ...



Links to this post

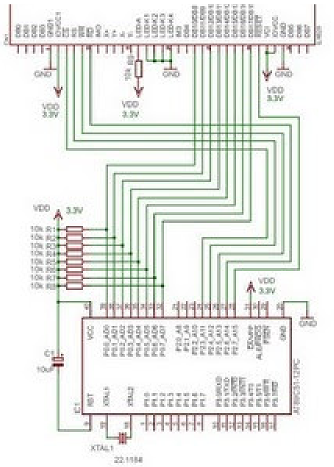
0 comments

Oct 7, 2010

ILI9325 320x240 64k Color LCD

Originally the specification of ILI9325 is 240RGB x 320 resolution and 262K color but in my application I just use 64k color due to memory limitation. As you can see on the picture, the circuit is made of a 2.4 inch LCD, 0.8mm to 2.5mm adapter PCB, AT89S52 mcu, and some passive components. The demo code is to display the lines in random color, 3 lines text with random background color, and 2 integers at left bottom.

Pull-up resistors on P0 is needed because the port is open collector. AT89S51/52 is designed to work on 2.6-5.5V range, so it can be connected directly to ILI9325 when both powered by 3.3V supply. And also very interesting that crystal 12-24MHz works without capacitors, could be because the internal capacitance meet the resonance requirement. Be aware of this, your crystal might need external capacitors.
This LCD has 16-bit width data bus, but I use 8-bit data width to reduce the wiring. There is a resistor placed on R1 or R2 position on the LCD for the data width selection, check the selection before doing connection. Normally you can tell the factory to set it before delivery. The other pins need to be wired to mcu: RS, RD, WR, RST, CS. So at least 13 wires connecting mcu and LCD. LED is connected to +5V by 2.5ohm resistor. This LCD has resistive touch screen output X+, X-, Y+, Y-. You'll need touch screen controller such as ADS7843 to interface those pins to mcu. Be noticed, many companies produce this LCD with different pinouts, so help yourself rearrange the wiring.



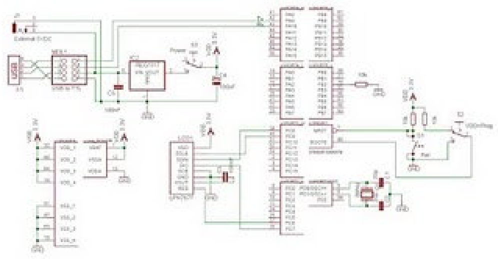
```
#include <reg52.h>
#include <intrins.h>
#include <stdlib.h>
#define uchar unsigned char
#define uint  unsigned int
#define ulong unsigned long

sbit    LCD_CS  = P2^3;
sbit    LCD_RS  = P2^4;
sbit    LCD_WR  = P2^5;
sbit    LCD_RD  = P2^6;
sbit    LCD_RST = P2^7;
#define DataPort  P0

code unsigned char font_8x16[1536] =
{

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x18,0x3C,0x3C,0x3C,0x18,0x18,0x18,0x00
,0x18,0x18,0x00,0x00,0x00,0x00,

0x00,0x66,0x66,0x66,0x24,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x6C,0x6C,0xFE,0x6C,0x6C,0x6C,0xFE,0x6C,0x6C,0x00,0x00,0x00,0x00,
0x18,0x18,0x7C,0xC6,0xC2,0xC0,0x7C,0x06,0x86,0xC6,0x7C,0x18,0x18,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0xC2,0xC6,0x0C,0x18,0x30,0x60,0xC6,0x86,0x00,0x00,0x00,0x00,
0x00,0x00,0x38,0x6C,0x6C,0x38,0x76,0xDC,0xCC,0xCC,0xCC,0x76,0x00,0x00,0x00,0x00,
0x00,0x30,0x30,0x30,0x60,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x0C,0x18,0x30,0x30,0x30,0x30,0x30,0x30,0x18,0x0C,0x00,0x00,0x00,0x00,
0x00,0x00,0x30,0x18,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x18,0x30,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x66,0x3C,0xFF,0x3C,0x66,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x7E,0x18,0x18,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x18,0x30,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x02,0x06,0x0C,0x18,0x30,0x60,0xC0,0x80,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0xCE,0xD6,0xD6,0xE6,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x18,0x38,0x78,0x18,0x18,0x18,0x18,0x18,0x18,0x7E,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0x06,0x0C,0x18,0x30,0x60,0xC0,0xC6,0xFE,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0x06,0x06,0x3C,0x06,0x06,0x06,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x0C,0x1C,0x3C,0x6C,0xCC,0xFE,0x0C,0x0C,0x0C,0x1E,0x00,0x00,0x00,0x00,
0x00,0x00,0xFE,0xC0,0xC0,0xC0,0xFC,0x0E,0x06,0x06,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x38,0x60,0xC0,0xC0,0xFC,0xC6,0xC6,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0xFE,0xC6,0x06,0x06,0x0C,0x18,0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0xC6,0x7C,0xC6,0xC6,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0xC6,0x7E,0x06,0x06,0x06,0x0C,0x78,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00,0x18,0x18,0x30,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x06,0x0C,0x18,0x30,0x60,0x30,0x18,0x0C,0x06,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x00,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x60,0x30,0x18,0x0C,0x06,0x0C,0x18,0x30,0x60,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0x0C,0x18,0x18,0x18,0x00,0x18,0x18,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x7C,0xC6,0xC6,0xDE,0xDE,0xDE,0xDC,0xC0,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x10,0x38,0x6C,0xC6,0xC6,0xFE,0xC6,0xC6,0xC6,0xC6,0x00,0x00,0x00,0x00,
0x00,0x00,0xFC,0x66,0x66,0x66,0x7C,0x66,0x66,0x66,0x66,0xFC,0x00,0x00,0x00,0x00,
0x00,0x00,0x3C,0x66,0xC2,0xC0,0xC0,0xC0,0xC0,0xC2,0x66,0x3C,0x00,0x00,0x00,0x00,
0x00,0x00,0xF8,0x6C,0x66,0x66,0x66,0x66,0x66,0x66,0x6C,0xF8,0x00,0x00,0x00,0x00,
0x00,0x00,0xFE,0x66,0x62,0x68,0x78,0x68,0x60,0x62,0x66,0xFE,0x00,0x00,0x00,0x00,
0x00,0x00,0xFE,0x66,0x62,0x68,0x78,0x68,0x60,0x60,0x60,0xF0,0x00,0x00,0x00,0x00,
0x00,0x00,0x3C,0x66,0xC2,0xC0,0xC0,0xDE,0xC6,0xC6,0x66,0x3A,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xC6,0xC6,0xC6,0xFE,0xC6,0xC6,0xC6,0xC6,0xC6,0x00,0x00,0x00,0x00,
0x00,0x00,0x3C,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x3C,0x00,0x00,0x00,0x00,
0x00,0x00,0x1E,0x0C,0x0C,0x0C,0x0C,0x0C,0xCC,0xCC,0xCC,0x78,0x00,0x00,0x00,0x00,
0x00,0x00,0xE6,0x66,0x6C,0x6C,0x78,0x78,0x6C,0x66,0x66,0xE6,0x00,0x00,0x00,0x00,
0x00,0x00,0xF0,0x60,0x60,0x60,0x60,0x60,0x60,0x62,0x66,0xFE,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xEE,0xFE,0xFE,0xD6,0xC6,0xC6,0xC6,0xC6,0xC6,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xE6,0xF6,0xFE,0xDE,0xCE,0xC6,0xC6,0xC6,0xC6,0x00,0x00,0x00,0x00,
0x00,0x00,0x38,0x6C,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0x6C,0x38,0x00,0x00,0x00,0x00,
0x00,0x00,0xFC,0x66,0x66,0x66,0x7C,0x60,0x60,0x60,0x60,0xF0,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0xD6,0xDE,0x7C,0x0C,0x0E,0x00,0x00,
0x00,0x00,0xFC,0x66,0x66,0x66,0x7C,0x6C,0x66,0x66,0x66,0xE6,0x00,0x00,0x00,0x00,
0x00,0x00,0x7C,0xC6,0xC6,0x60,0x38,0x0C,0x06,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0x7E,0x7E,0x5A,0x18,0x18,0x18,0x18,0x18,0x18,0x3C,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0x6C,0x38,0x10,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xC6,0xC6,0xC6,0xC6,0xD6,0xD6,0xFE,0x6C,0x6C,0x00,0x00,0x00,0x00,
0x00,0x00,0xC6,0xC6,0x6C,0x6C,0x38,0x38,0x6C,0x6C,0xC6,0xC6,0x00,0x00,0x00,0x00,
0x00,0x00,0x66,0x66,0x66,0x66,0x3C,0x18,0x18,0x18,0x18,0x3C,0x00,0x00,0x00,0x00,
0x00,0x00,0xFE,0xC6,0x86,0x0C,0x18,0x30,0x60,0xC2,0xC6,0xFE,0x00,0x00,0x00,0x00,
0x00,0x00,0x3C,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x3C,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x80,0xC0,0xE0,0x70,0x38,0x1C,0x0E,0x06,0x02,0x00,0x00,0x00,0x00,
0x00,0x00,0x3C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x3C,0x00,0x00,0x00,0x00,
0x10,0x38,0x6C,0xC6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,
0x30,0x30,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x78,0x0C,0x7C,0xCC,0xCC,0xCC,0x76,0x00,0x00,0x00,0x00,
    0x00,0x00,0x1C,0x0C,0x0C,0x3C,0x6C,0xCC,0xCC,0xCC,0xCC,0x76,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x7C,0xC6,0xFE,0xC0,0xC0,0xC6,0x7C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x38,0x6C,0x64,0x60,0xF0,0x60,0x60,0x60,0x60,0xF0,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x76,0xCC,0xCC,0xCC,0xCC,0xCC,0x7C,0x0C,0xCC,0x78,0x00,
    0x00,0x00,0xE0,0x60,0x60,0x6C,0x76,0x66,0x66,0x66,0x66,0xE6,0x00,0x00,0x00,0x00,
    0x00,0x00,0x18,0x18,0x00,0x38,0x18,0x18,0x18,0x18,0x18,0x3C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x06,0x06,0x00,0x0E,0x06,0x06,0x06,0x06,0x06,0x06,0x66,0x66,0x3C,0x00,
    0x00,0x00,0xE0,0x60,0x60,0x66,0x6C,0x78,0x78,0x6C,0x66,0xE6,0x00,0x00,0x00,0x00,
    0x00,0x00,0x38,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x3C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xEC,0xFE,0xD6,0xD6,0xD6,0xD6,0xD6,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xDC,0x66,0x66,0x66,0x66,0x66,0x66,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x7C,0xC6,0xC6,0xC6,0xC6,0xC6,0x7C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xDC,0x66,0x66,0x66,0x66,0x66,0x7C,0x60,0x60,0xF0,0x00,
    0x00,0x00,0x00,0x00,0x00,0x76,0xCC,0xCC,0xCC,0xCC,0xCC,0x7C,0x0C,0x0C,0x1E,0x00,
    0x00,0x00,0x00,0x00,0x00,0xDC,0x76,0x62,0x60,0x60,0x60,0xF0,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x7C,0xC6,0x60,0x38,0x0C,0xC6,0x7C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x10,0x30,0x30,0xFC,0x30,0x30,0x30,0x30,0x36,0x1C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xCC,0xCC,0xCC,0xCC,0xCC,0xCC,0x76,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x66,0x66,0x66,0x66,0x66,0x3C,0x18,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xC6,0xC6,0xC6,0xD6,0xD6,0xFE,0x6C,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xC6,0x6C,0x38,0x38,0x38,0x6C,0xC6,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0xC6,0xC6,0xC6,0xC6,0xC6,0xC6,0x7E,0x06,0x0C,0xF8,0x00,
    0x00,0x00,0x00,0x00,0x00,0xFE,0xCC,0x18,0x30,0x60,0xC6,0xFE,0x00,0x00,0x00,0x00,
    0x00,0x00,0x0E,0x18,0x18,0x18,0x70,0x18,0x18,0x18,0x18,0x0E,0x00,0x00,0x00,0x00,
    0x00,0x00,0x18,0x18,0x18,0x18,0x00,0x18,0x18,0x18,0x18,0x18,0x00,0x00,0x00,0x00,
    0x00,0x00,0x70,0x18,0x18,0x18,0x0E,0x18,0x18,0x18,0x18,0x70,0x00,0x00,0x00,0x00,
    0x00,0x00,0x76,0xDC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x10,0x38,0x6C,0xC6,0xC6,0xC6,0xFE,0x00,0x00,0x00,0x00,0x00
};

void  delay(uint t)
{
    t+=170;
    while(--t);
}

void  delay_ms(uint t)
{
    uchar i,j;
    while(--t)
    {
        j=8;
        while(--j) while(--i);
    }
}

void LCD_Write_Cmd(uint inpcmd)
{
    LCD_CS = 0;
    LCD_RS = 0;
    //LCD_RD = 1;
    DataPort = ((unsigned char *) &inpcmd) [0];
    LCD_WR = 0;
    LCD_WR = 1;
    DataPort = ((unsigned char *) &inpcmd) [1];
    LCD_WR = 0;
    LCD_WR = 1;
    LCD_CS = 1;
    //LCD_RS = 1;
}
void  LCD_Write_Data(uint inpdata)
{
    LCD_CS = 0;
    LCD_RS = 1;
    //LCD_RD = 1;
    DataPort = ((unsigned char *) &inpdata) [0];
    LCD_WR = 0;
    LCD_WR = 1;
    DataPort = ((unsigned char *) &inpdata) [1];
    LCD_WR = 0;
    LCD_WR = 1;
    LCD_CS = 1;
    //LCD_RS = 0;
}
void    LCD_Write_CmdData(uint inpcmd, uint inpdata)
{
    LCD_CS = 0;
    LCD_RS = 0;
    //LCD_RD = 1;
    DataPort = ((unsigned char *) &inpcmd) [0];
    LCD_WR = 0;
    LCD_WR = 1;
    DataPort = ((unsigned char *) &inpcmd) [1];
    LCD_WR = 0;
    LCD_WR = 1;
    LCD_RS = 1;
    DataPort = ((unsigned char *) &inpdata) [0];
    LCD_WR = 0;
    LCD_WR = 1;
    LCD_WR = 1;
    LCD_CS = 1;
    //LCD_RS = 0;
}
void LCD_SetCursor(uint x,uint y)
{
        LCD_Write_CmdData(0x0020,y);
        LCD_Write_CmdData(0x0021,319-x);
}

void LCD_SetWindow(uint StartX, uint StartY, uint EndX, uint EndY)
{
    LCD_Write_CmdData(0x0050, StartY);
    LCD_Write_CmdData(0x0052, 319-EndX);
    LCD_Write_CmdData(0x0051, EndY);
    LCD_Write_CmdData(0x0053, 319-StartX);
    LCD_SetCursor(StartX,StartY);
}

void LCD_Clear(uint color)
{
    uint  i,j;
    LCD_SetWindow (0,0,319,239);
    LCD_Write_Cmd(0x0022);
    for(i=0;i<(240);i++)  for(j=0;j<(320);j++) LCD_Write_Data(color);

}

void LCD_Init(void)
{
    LCD_RST=1;
    delay(20);
    LCD_RST=0;
    delay(60);
    LCD_RST=1;
    delay(60);
    //Display Setting
    LCD_Write_CmdData(0x00e7,0x0010);
    LCD_Write_CmdData(0x0000,0x0001);                    //start internal osc
    LCD_Write_CmdData(0x0001,0x0100);
    LCD_Write_CmdData(0x0002,0x0700);                    //power on sequence
    //LCD_Write_CmdData(0x0003,(1<<12)|(1<<5)|(1<<4) );  //65K
    LCD_Write_CmdData(0x0003,(1<<12)|(1<<4)|(1<<3) );
    LCD_Write_CmdData(0x0004,0x0000);
    LCD_Write_CmdData(0x0008,0x0302);
    LCD_Write_CmdData(0x0009,0x0000);
    LCD_Write_CmdData(0x000a,0x0004);                    //display setting
    LCD_Write_CmdData(0x000c,0x0001);                    //display setting
    LCD_Write_CmdData(0x000d,0x0000);                    //0f3c
    LCD_Write_CmdData(0x000f,0x0000);
    //Power On sequence //
    LCD_Write_CmdData(0x0010,0x0000);
    LCD_Write_CmdData(0x0011,0x0007);
    LCD_Write_CmdData(0x0012,0x0000);
    LCD_Write_CmdData(0x0013,0x0000);
    delay(1000);
    LCD_Write_CmdData(0x0010,0x1290);
    LCD_Write_CmdData(0x0011,0x0015);
    delay(1000);
    LCD_Write_CmdData(0x0012,0x0092);
    delay(1000);
    LCD_Write_CmdData(0x0013,0x1800);
    LCD_Write_CmdData(0x0029,0x000f);
    LCD_Write_CmdData(0x002b,0x000d);
    delay(1000);
    LCD_Write_CmdData(0x0020,0x0000);
    LCD_Write_CmdData(0x0021,0x0000);
    //////////////////////
    delay(1000);
    LCD_Write_CmdData(0x0030,0x0000);
    LCD_Write_CmdData(0x0031,0x0104);
    LCD_Write_CmdData(0x0032,0x0000);
    LCD_Write_CmdData(0x0035,0x0203);
    LCD_Write_CmdData(0x0036,0x0405);
    LCD_Write_CmdData(0x0037,0x0000);
    LCD_Write_CmdData(0x0038,0x0203);
    LCD_Write_CmdData(0x0039,0x0000);
    LCD_Write_CmdData(0x003c,0x0203);
    LCD_Write_CmdData(0x003d,0x0405);
    delay(1000);
    LCD_Write_CmdData(0x0050, 0x0000);
    LCD_Write_CmdData(0x0051, 0x00ef);
    LCD_Write_CmdData(0x0052, 0x0000);
    LCD_Write_CmdData(0x0053, 0x013f);
    LCD_Write_CmdData(0x0060,0xa700);
    LCD_Write_CmdData(0x0061,0x0001);
    LCD_Write_CmdData(0x006a,0x0000);
    LCD_Write_CmdData(0x0080,0x0000);
    LCD_Write_CmdData(0x0081,0x0000);
    LCD_Write_CmdData(0x0084,0x0000);
    LCD_Write_CmdData(0x0085,0x0000);
    LCD_Write_CmdData(0x0090,0x0033);
    LCD_Write_CmdData(0x0092,0x0000);
    LCD_Write_CmdData(0x0093,0x0003);
    LCD_Write_CmdData(0x0095,0x0110);
    LCD_Write_CmdData(0x0097,0x0000);
    LCD_Write_CmdData(0x0098,0x0000);
    LCD_Write_CmdData(0x0007,0x0133);
    LCD_SetWindow (0,0,319,239);
    delay(1000);
    LCD_Clear(0xf8);
}

void LCD_DrawPoint(uint x, uint y, uint pointcolor)
{
    if ( (x>=320)||(y>=240) ) return;
    LCD_SetCursor(x,y);
    LCD_Write_Cmd(0x0022);
    LCD_Write_Data(pointcolor);
}
void LCD_PutChar(uint x,uint y,uchar c,uint charColor,uint bkColor)
{
    uint i=0;
    uint j=0;
    uchar tmp_char=0;
    for (i=0;i<16;i++)
    {
        tmp_char=font_8x16[((c-0x20)*16)+i];
        for (j=0;j<8;j++)
        {
            if ((tmp_char  & 0x80) == 0x80)
            {
                LCD_DrawPoint(x+j,y+i,charColor);
            }
            else
            {
                LCD_DrawPoint(x+j,y+i,bkColor);
            }
            tmp_char<<=1;
        }
    }
}
void LCD_PutText(uint x, uint y, uchar *str, uint len , uint Color, uint bkColor)
{
    uchar i;
    for (i=0;i<len;i++) LCD_PutChar((x+8*i), y, *str++, Color, bkColor);
}

void LCD_Print_Int(uint inpInt, uint x, uint y)
{
    uint tmpValue;
    uchar tmpStr[5];
    tmpValue = inpInt;
    tmpStr[0]=(tmpValue/10000)+48; tmpValue%=10000;
    tmpStr[1]=(tmpValue/1000)+48; tmpValue%=1000;
    tmpStr[2]=(tmpValue/100)+48; tmpValue%=100;
    tmpStr[3]=(tmpValue/10)+48; tmpValue%=10;
    tmpStr[4]=(tmpValue/1)+48;
    LCD_PutText(x,y,tmpStr, 5, 0xffff, 0);
}

void  main(void)
{
    uint i,j,k;
    uint randcolor;
    uint color_foreground, color_background;
    LCD_Init();
    while(1)
    {
        LCD_SetWindow (0,0,319,239);
        LCD_Write_Cmd(0x0022);
        for(i=0;i<12;i++)
        {
            for(j=0;j<20;j++)
            {
                randcolor=rand() + rand();
                for(k=0;k<320;k++) LCD_Write_Data(randcolor);
            }
        }
        color_foreground = rand() + rand();
        color_background = rand() + rand();
        LCD_PutText(75,24, " Atmojo S.Dao    ", 17, color_foreground, 0);
        LCD_PutText(75,44, " Silvester S.Dao ", 17, color_foreground, 0);
        LCD_PutText(75,64, " Lusia Lusiana   ", 17, color_foreground, 0);
        for (i=1; i<50; i++)
        {
            LCD_Print_Int(i, 5,  220);
            LCD_Print_Int(rand(), 75,  220);
        }
}
```

Links to this post

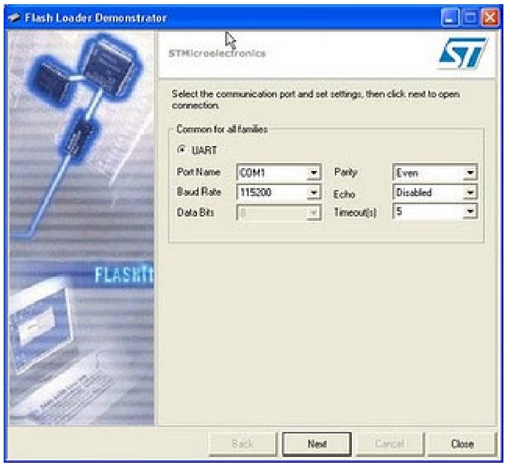
0 comments

Oct 6, 2010

Very Low Cost Cortex-M3 Board: STM32



This is fun. Playing with mcu from different manufacturer. They are all ARM architectured but after playing with some demo codes you'll realize that the way they are programmed is different, not only because the prepared library is different but also because the hardware features. STM32F103RBT6 has 128kbytes flash while STM32F103R8T6 has 64kbytes only. Both equipped with USB interface and 20kbytes SRAM. I found one minus point of this mcu; the compiling time is a bit longer than LPC2100. STM32 has Boot0 and Boot1 pins, on this board Boot1 is always 0 (PB2). To program, switch Boot0 to VDD and reset. To run your code, switch Boot0 to GND and reset. VDDA and VSSA are power supply for analog circuit (ADC and DAC), as usual make sure they are connected as close as possible to power supply to avoid digital noise come in to the analog circuit. Or, you can prepare a separated filter for analog power supply.



Select the communication port and set settings, then click next to open connection.

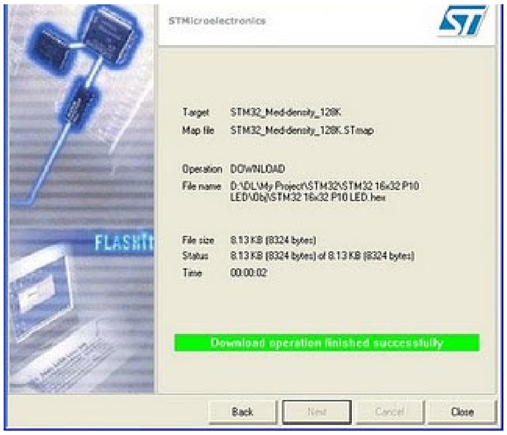
Common for all families

● UART

Port Name: COM1
Baud Rate: 115200
Data Bits: 8
Parity: Even
Echo: Disabled
Timeout(s): 5

I use Flash Loader Demonstrator software to transfer the hex code from PC to mcu, it's free software provided by STMicroelectronics for non-commercial use (I think so, then it's called demonstrator).You can use RS232/TTL converter if you PC has COM port, or use USB/TTL converter if no COM port. So exciting when top speed 115200bps works perfectly, it doesn't take that long to download a big size hex.

Finally, there is a report telling the status of download. Instead of hex, the code size is shown in decimal numbers makes user easily estimate the flash space left. STM32F103RBT6 costs less than 5USD/pc. Need cheaper one? STM32F103R6T6 with 32kbytes flash and 10kbytes SRAM. All those three chips are 64 pins package. Googling for STM32 family, there are wide range of ports, memory, speed and peripherals available. If you rich enough, of course you can buy a ready made development system costs 100-



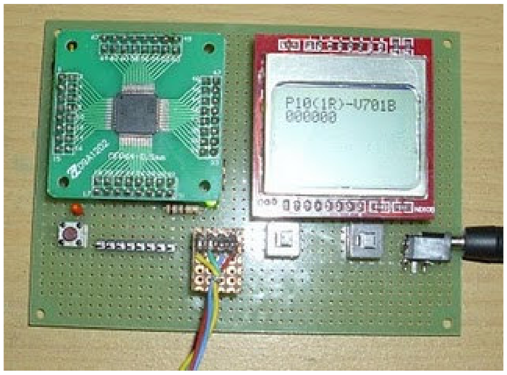
Target STM32_Med-density_128K
Map file STM32_Med-density_128K.STmap

Operation DOWNLOAD
File name D:\DL\My Project\STM32\STM32 16x32 P10 LED\Obj\STM32 16x32 P10 LED.hex

File size 8.13 KB (8324 bytes)
Status 8.13 KB (8324 bytes) of 8.13 KB (8324 bytes)
Time 00:00:02

Download operation finished successfully

with software will make you less experience with the hardware. Try it and enjoy!



Here we are, finally I managed to upload a photo of the matrix board. It looks similar with LPC2100 mcu because same LQFP64 footprint. Zoom in a bit on the chip, you'll see the ST logo on it. One more interesting thing, STM32 has internal oscillator so you can take the crystal off the circuit.

Below is example code of LED blink on PB5.

```
int main(void)
{
  RCC_Configuration();
  RCC_APB2PeriphClockCmd( RCC_APB2Periph_USART1 |RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB |
                          RCC_APB2Periph_GPIOC | RCC_APB2Periph_GPIOD |
                          RCC_APB2Periph_GPIOE, ENABLE);


  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOB, &GPIO_InitStructure);


  while (1)
  {
    GPIO_SetBits  (GPIOB, GPIO_Pin_5);
    delay_ms(1000);
    GPIO_ResetBits(GPIOB, GPIO_Pin_5);
    delay_ms(1000);
  }
}
```

Links to this post

0 comments

Newer Posts Home Older Posts