??? http://www.wifi-robots.com/thread-367-1-1.html ???liuviking WIFI Robot ??
??Sources Code ??

????WIFI Robto ?????????Arduino +MSP 430 ??code ,????????Arduino ?MSP430 ??.

?????WIFI ??????????
Windows ? http://www.wifi-robots.com/thread-1790-1-1.html
Android ? http://www.wifi-robots.com/thread-1180-1-1.html
API http://www.wifi-robots.com/thread-938-1-1.html
??VB6 Code http://www.wifi-robots.com/thread-1816-1-1.html

??

```
        move Forward  'A1');
         turn Left(){A2
         moveBack(){ A3
        turn right a4
```

```
/*
?????
WIFI????·???????????? www.wifi-robots.com
??????????????????????????????????????????????????
By WIFI????·????????
*/

#ifndef _STC_NEW_8051_H_
#define _STC_NEW_8051_H_
//------------------------------------------------------------------------------
//??? 1T 8051?? ???????????? C51 Core SFRs
//                          7    6    5     4    3    2    1    0  Reset Value
sfr ACC  = 0xE0; //Accumulator                                        0000,0000
sfr B    = 0xF0; //B Register                                       0000,0000
sfr PSW  = 0xD0; //Program Status Word     CY   AC   F0   RS1  RS0  OV   F1   P    0000,0000
//-----------------------------------
sbit CY  = PSW^7;
sbit AC  = PSW^6;
sbit F0  = PSW^5;
sbit RS1 = PSW^4;
sbit RS0 = PSW^3;
sbit OV  = PSW^2;
sbit P   = PSW^0;
//-----------------------------------
sfr SP   = 0x81; //Stack Pointer                                    0000,0111
sfr DPL  = 0x82; //Data Pointer Low Byte                                0000,0000
```

```
sfr DPH  = 0x83; //Data Pointer High Byte                                    0000,0000
//---------------------------------------------------------------------------------
//??? 1T 8051?? ??????????????
//                          7    6     5    4    3     2    1    0    Reset Value
sfr PCON   = 0x87; //Power Control       SMOD  SMOD0  LVDF  POF   GF1   GF0   PD   IDL   0001,0000
//                          7    6     5    4    3     2    1    0  Reset Value
sfr AUXR  = 0x8E; //Auxiliary Register  T0x12 T1x12 UART_M0x6 BRTR S2SMOD BRTx12 EXTRAM S1BRS
0000,0000
//-----------------------------------
sfr AUXR1 = 0xA2; //Auxiliary Register 1  -  PCA_P4  SPI_P4  S2_P4  GF2   ADRJ   -   DPS  0000,0000
/*
PCA_P4:
   0, ??PCA ?P1 ?
   1?PCA/PWM ?P1 ????P4 ?: ECI ?P1.2 ???P4.1 ??
                          PCA0/PWM0 ?P1.3 ???P4.2 ?
                          PCA1/PWM1 ?P1.4 ???P4.3 ?
SPI_P4:
   0, ??SPI ?P1 ?
   1?SPI ?P1 ????P4 ?: SPICLK ?P1.7 ???P4.3 ?
                       MISO ?P1.6 ???P4.2 ?
                       MOSI ?P1.5 ???P4.1 ?
                       SS ?P1.4 ???P4.0 ?
S2_P4:
   0, ??UART2 ?P1 ?
   1?UART2 ?P1 ????P4 ?: TxD2 ?P1.3 ???P4.3 ?
                        RxD2 ?P1.2 ???P4.2 ?
GF2: ?????

ADRJ:
   0, 10 ?A/D ??????8 ???ADC_RES ???, ?2 ???ADC_RESL ???
   1?10 ?A/D ???????2 ???ADC_RES ?????2 ?, ?8 ???ADC_RESL ???

DPS: 0, ????????DPTR0
    1??????????DPTR1
*/
//-----------------------------------
sfr WAKE_CLKO = 0x8F; //??? SFR WAK1_CLKO
/*
    7         6         5         4         3      2      1      0        Reset Value
  PCAWAKEUP  RXD_PIN_IE  T1_PIN_IE  T0_PIN_IE  LVD_WAKE    _    T1CLKO  T0CLKO     0000,0000B

b7 - PCAWAKEUP : PCA ????? powerdown?
b6 - RXD_PIN_IE : ? P3.0(RXD) ????? RI ???? powerdown(????????)?
b5 - T1_PIN_IE : ? T1 ?????? T1 ???????? powerdown(????????)?
b4 - T0_PIN_IE : ? T0 ?????? T0 ???????? powerdown(????????)?
b3 - LVD_WAKE : ? CMPIN ?????? LVD ???????? powerdown(????????)?
b2 -
b1 - T1CLKO : ?? T1CKO(P3.5) ??? T1 ?????Fck1 = 1/2 T1 ???
b0 - T0CLKO : ?? T0CKO(P3.4) ??? T0 ?????Fck0 = 1/2 T1 ???
*/
```

```
//-----------------------------------
sfr CLK_DIV = 0x97; //Clock Divder        -    -    -     -    - CLKS2 CLKS1 CLKS0 xxxx,x000
//-----------------------------------
sfr BUS_SPEED = 0xA1; //Stretch register     -    -   ALES1   ALES0   -   RWS2  RWS1  RWS0 xx10,x011
/*
ALES1 and ALES0:
00 : The P0 address setup time and hold time to ALE negative edge is one clock cycle
01 : The P0 address setup time and hold time to ALE negative edge is two clock cycles.
10 : The P0 address setup time and hold time to ALE negative edge is three clock cycles. (default)
11 : The P0 address setup time and hold time to ALE negative edge is four clock cycles.

RWS2,RWS1,RWS0:
 000 : The MOVX read/write pulse is 1 clock cycle.
 001 : The MOVX read/write pulse is 2 clock cycles.
 010 : The MOVX read/write pulse is 3 clock cycles.
 011 : The MOVX read/write pulse is 4 clock cycles. (default)
 100 : The MOVX read/write pulse is 5 clock cycles.
 101 : The MOVX read/write pulse is 6 clock cycles.
 110 : The MOVX read/write pulse is 7 clock cycles.
 111 : The MOVX read/write pulse is 8 clock cycles.
*/
//-------------------------------------------------------------------------------
//??? 1T 8051?? ????????????
//??????????????????????????????????????
//??????????????? ??? 1T 8051?? ???????
//                          7    6    5    4    3    2    1    0   Reset Value
sfr IE     = 0xA8;  //???????      EA  ELVD  EADC   ES   ET1  EX1  ET0  EX0  0x00,0000
//-----------------------
sbit EA      = IE^7;
sbit ELVD    = IE^6; //?????????
sbit EADC    = IE^5; //ADC ?????
sbit ES      = IE^4;
sbit ET1     = IE^3;
sbit EX1     = IE^2;
sbit ET0     = IE^1;
sbit EX0     = IE^0;
//-----------------------
sfr IE2      = 0xAF;  //Auxiliary Interrupt   -    -    -    -    -    - ESPI  ES2  0000,0000B
//-----------------------
//                          7    6    5    4    3    2    1    0   Reset Value
sfr IP     = 0xB8; //???????     PPCA  PLVD  PADC  PS   PT1  PX1  PT0  PX0   0000,0000
//--------
sbit PPCA    = IP^7;  //PCA ???????
sbit PLVD    = IP^6;  //?????????
sbit PADC    = IP^5;  //ADC ?????
sbit PS      = IP^4;
sbit PT1     = IP^3;
sbit PX1     = IP^2;
sbit PT0     = IP^1;
sbit PX0     = IP^0;
```

```
//-----------------------
//                          7     6     5     4     3     2     1     0   Reset Value
sfr IPH   = 0xB7; //???????     PPCAH  PLVDH  PADCH  PSH  PT1H  PX1H  PT0H  PX0H   0000,0000
sfr IP2   = 0xB5; //                -     -     -     -     -     -  PSPI   PS2   xxxx,xx00
sfr IPH2  = 0xB6; //                -     -     -     -     -     -  PSPIH  PS2H  xxxx,xx00
//-----------------------
//??? 1T 8051?? ???I/O ????????
//                          7     6     5     4     3     2     1     0       Reset Value
sfr P0   = 0x80; //8 bitPort0       P0.7  P0.6  P0.5  P0.4  P0.3  P0.2  P0.1  P0.0     1111,1111
sfr P0M0 = 0x94; //                                                                    0000,0000
sfr P0M1 = 0x93; //                                                                    0000,0000
sfr P1   = 0x90; //8 bitPort1       P1.7  P1.6  P1.5  P1.4  P1.3  P1.2  P1.1  P1.0     1111,1111
sfr P1M0 = 0x92; //                                                                    0000,0000
sfr P1M1 = 0x91; //                                                                    0000,0000
sfr P1ASF = 0x9D; //P1 analog special function
sfr P2   = 0xA0; //8 bitPort2       P2.7  P2.6  P2.5  P2.4  P2.3  P2.2  P2.1  P2.0     1111,1111
sfr P2M0 = 0x96; //                                                                    0000,0000
sfr P2M1 = 0x95; //                                                                    0000,0000
sfr P3   = 0xB0; //8 bitPort3       P3.7  P3.6  P3.5  P3.4  P3.3  P3.2  P3.1  P3.0     1111,1111
sfr P3M0 = 0xB2; //                                                                    0000,0000
sfr P3M1 = 0xB1; //                                                                    0000,0000
sfr P4   = 0xC0; //8 bitPort4       P4.7  P4.6  P4.5  P4.4  P4.3  P4.2  P4.1  P4.0     1111,1111
sfr P4M0 = 0xB4; //                                                                    0000,0000
sfr P4M1 = 0xB3; //                                                                    0000,0000
//                          7     6      5      4     3     2     1     0    Reset Value
sfr P4SW = 0xBB; //Port-4 switch      -  LVD_P4.6  ALE_P4.5  NA_P4.4  -     -     -     -      x000,xxxx

sfr P5   = 0xC8; //8 bitPort5        -     -     -     -  P5.3  P5.2  P5.1  P5.0   xxxx,1111
sfr P5M0 = 0xCA; //                                                                    0000,0000
sfr P5M1 = 0xC9; //                                                                    0000,0000
//-----------------------------------------------------------------------------
//??? 1T 8051?? ?????????????
//                          7     6     5     4     3     2     1     0   Reset Value
sfr TCON = 0x88; //T0/T1 Control       TF1   TR1   TF0   TR0   IE1   IT1   IE0   IT0   0000,0000
//-----------------------------------
sbit TF1 = TCON^7;
sbit TR1 = TCON^6;
sbit TF0 = TCON^5;
sbit TR0 = TCON^4;
sbit IE1 = TCON^3;
sbit IT1 = TCON^2;
sbit IE0 = TCON^1;
sbit IT0 = TCON^0;
//-----------------------------------
sfr TMOD = 0x89; //T0/T1 Modes         GATE1 C/T1  M1_1  M1_0  GATE0 C/T0  M0_1  M0_0   0000,0000
sfr TL0  = 0x8A; //T0 Low Byte                                                          0000,0000
sfr TH0  = 0x8C; //T0 High Byte                                                         0000,0000
sfr TL1  = 0x8B; //T1 Low Byte                                                          0000,0000
sfr TH1  = 0x8D; //T1 High Byte                                                         0000,0000
//-----------------------------------------------------------------------------
```

```
//??? 1T 8051?? ?????????????
//                          7    6    5    4    3    2    1    0    Reset Value
sfr SCON = 0x98; //Serial Control       SM0/FE SM1  SM2  REN  TB8  RB8  TI   RI   0000,0000
//-----------------------------------
sbit SM0 = SCON^7;  //SM0/FE
sbit SM1 = SCON^6;
sbit SM2 = SCON^5;
sbit REN = SCON^4;
sbit TB8 = SCON^3;
sbit RB8 = SCON^2;
sbit TI  = SCON^1;
sbit RI  = SCON^0;
//-----------------------------------
sfr SBUF = 0x99; //Serial Data Buffer                                    xxxx,xxxx
sfr SADEN = 0xB9; //Slave Address Mask                                     0000,0000
sfr SADDR = 0xA9; //Slave Address                                        0000,0000
//-----------------------------------
//                      7     6     5     4     3     2    1    0      Reset Value
sfr S2CON = 0x9A; //S2 Control  S2SM0  S2SM1  S2SM2  S2REN  S2TB8  S2RB8  S2TI  S2RI     00000000B
sfr S2BUF = 0x9B; //S2 Serial Buffer                                     xxxx,xxxx
sfr BRT = 0x9C; //S2 Baud-Rate Timer                                     0000,0000
//-----------------------------------------------------------------------------
//??? 1T 8051?? ????????????????
sfr WDT_CONTR = 0xC1; //Watch-Dog-Timer Control register
//                          7    6    5    4     3     2  1  0    Reset Value
//                        WDT_FLAG  -  EN_WDT CLR_WDT IDLE_WDT PS2 PS1 PS0    xx00,0000
//-----------------------

//-----------------------------------------------------------------------------
//??? 1T 8051?? ???PCA/PWM ???????
//                           7    6    5    4    3    2    1    0    Reset Value
sfr CCON   = 0xD8;  //PCA ??????   CF   CR   -    -    -    -    CCF1 CCF0   00xx,xx00
//-----------------------
sbit CF    = CCON^7; //PCA???????,????????,??????0?
sbit CR    = CCON^6; //1:?? PCA ?????, ??????0?
//-
//-
sbit CCF1  = CCON^1; //PCA ??1 ????, ?????, ??????0?
sbit CCF0  = CCON^0; //PCA ??0 ????, ?????, ??????0?
//-----------------------
sfr CMOD  = 0xD9; //PCA ????????  CIDL  -    -    -    CPS2  CPS1  CPS0  ECF   0xxx,x000
/*
CIDL: idle ??? PCA ?????????, 0: ????, 1: ?????

CPS2: PCA ????????? 2?
CPS1: PCA ????????? 1?
CPS0: PCA ????????? 0?
  CPS2  CPS1  CPS0
   0     0     0   ?????? fosc/12?
   0     0     1   ?????? fosc/2?
```

```
  0    1    0   Timer0 ???
  0    1    1   ? ECI/P3.4 ??????????? fosc/2?
  1    0    0   ???????  Fosc/1
  1    0    1   ??????/4?Fosc/4
  1    1    0   ??????/6?Fosc/6
  1    1    1   ??????/8?Fosc/8

ECF: PCA??????????, 1--?? CF(CCON.7) ?????
*/
//-----------------------
sfr CL    = 0xE9; //PCA ?????                                    0000,0000
sfr CH    = 0xF9; //PCA ?????                                    0000,0000
//-----------------------
//                          7    6    5    4    3    2    1    0    Reset Value
sfr CCAPM0 = 0xDA; //PCA ??0 PWM ???  -   ECOM0  CAPP0  CAPN0  MAT0  TOG0  PWM0  ECCF0   x000,0000
sfr CCAPM1 = 0xDB; //PCA ??1 PWM ???  -   ECOM1  CAPP1  CAPN1  MAT1  TOG1  PWM1  ECCF1   x000,0000

//ECOMn = 1:???????
//CAPPn = 1:????????????
//CAPNn = 1:????????????
//MATn  = 1:????????, ?? CCON ?? CCFn ???
//TOGn  = 1:????????, CEXn ????
//PWMn  = 1:? CEXn ??? PWM ???
//ECCFn = 1:?? CCON ?? CCFn ?????

//ECOMn  CAPPn  CAPNn  MATn  TOGn  PWMn  ECCFn
// 0     0     0     0     0     0     0   0x00   ????????
// x     1     0     0     0     0     x   0x21   16?CEXn??????????
// x     0     1     0     0     0     x   0x11   16?CEXn??????????
// x     1     1     0     0     0     x   0x31   16?CEXn??(????)???????
// 1     0     0     1     0     0     x   0x49   16???????
// 1     0     0     1     1     0     x   0x4d   16????????
// 1     0     0     0     0     1     0   0x42   8? PWM?

//ECOMn  CAPPn  CAPNn  MATn  TOGn  PWMn  ECCFn
// 0     0     0     0     0     0     0   0x00   ????
// 1     0     0     0     0     1     0   0x42   ??8?PWM, ???
// 1     1     0     0     0     1     1   0x63   PWM???????????
// 1     0     1     0     0     1     1   0x53   PWM???????????
// 1     1     1     0     0     1     1   0x73   PWM?????????????????

//-----------------------
sfr CCAP0L = 0xEA; //PCA ?? 0 ???/?????? 8 ??                    0000,0000
sfr CCAP0H = 0xFA; //PCA ?? 0 ???/?????? 8 ??                    0000,0000
sfr CCAP1L = 0xEB; //PCA ?? 1 ???/?????? 8 ??                    0000,0000
sfr CCAP1H = 0xFB; //PCA ?? 1 ???/?????? 8 ??                    0000,0000
//-----------------------
//                                7  6  5  4  3  2   1    0   Reset Value
sfr PCA_PWM0 = 0xF2; //PCA ??0 PWM ????        -  -  -  -  -  -  EPC0H EPC0L   xxxx,xx00
sfr PCA_PWM1 = 0xF3; //PCA ??1 PWM ????        -  -  -  -  -  -  EPC1H EPC1L   xxxx,xx00
```

```
//PCA_PWMn:   7     6     5     4     3     2     1     0
//          -     -     -     -     -     -   EPCnH  EPCnL
//B7-B2: ??
//B1(EPCnH): ? PWM ????? CCAPnH ?? 9 ???
//B0(EPCnL): ? PWM ????? CCAPnL ?? 9 ???
//-----------------------------------------------------------------------------
//??? 1T 8051?? ??? ADC ???????
//                              7      6     5      4       3     2    1    0   Reset Value
sfr ADC_CONTR = 0xBC; //A/D ??????? ADC_POWER SPEED1 SPEED0 ADC_FLAG ADC_START CHS2 CHS1
CHS0 0000,0000
sfr ADC_RES  = 0xBD;  //A/D ?????8? ADCV.9 ADCV.8 ADCV.7 ADCV.6 ADCV.5 ADCV.4 ADCV.3 ADCV.2
0000,0000
sfr ADC_RESL = 0xBE;  //A/D ??????2?                                  ADCV.1 ADCV.0     0000,0000
//-----------------------------------------------------------------------------
//??? 1T 8051?? ??? SPI ???????
//                           7     6     5     4     3     2     1     0    Reset Value
sfr SPCTL  = 0xCE; //SPI Control Register  SSIG  SPEN  DORD  MSTR  CPOL  CPHA  SPR1  SPR0  0000,0100
sfr SPSTAT = 0xCD; //SPI Status Register   SPIF  WCOL  -     -     -     -     -     -     00xx,xxxx
sfr SPDAT  = 0xCF; //SPI Data Register                                          0000,0000
//-----------------------------------------------------------------------------
//??? 1T 8051?? ??? IAP/ISP ???????
sfr IAP_DATA    = 0xC2;
sfr IAP_ADDRH   = 0xC3;
sfr IAP_ADDRL   = 0xC4;
//                                  7   6   5     4   3   2   1     0    Reset Value
sfr IAP_CMD     = 0xC5; //IAP Mode Table        0   -   -     -   -   -   MS1   MS0   0xxx,xx00
sfr IAP_TRIG    = 0xC6;
sfr IAP_CONTR   = 0xC7; //IAP Control Register  IAPEN SWBS SWRST CFAIL  -   WT2  WT1   WT0   0000,x000
//-----------------------------------------------------------------------------
#endif
```

```
??????/*
/*************************************************************************
?????
??   ??? ???   ???
0XFF     1        1       0XFF
??????
???  ???   ??
0X00   0X01   ??
0X00     0X02   ??
0X00     0X03   ??
0X00     0X04   ??
0X00     0X00   ??
0X01     0X01   ???
```

```
0X01      0X02    ???
0X02      0X01     ???
0X02      0X02     ???
0X03     ????   ??????
**********************************************************************/
?????
WIFI????·???????????? www.wifi-robots.com
??????????????????????????????????????????????????
By WIFI????·????????
*/
#include "uart.h"
#include <stdio.h>
#include "motor.h"
#include "steer.h"



extern uint16 se_timer[5];
uint8 buffer[3];
uint8 rec_flag=0;    //??0???? ??1????

void UART_init(void)
{
   PCON |= 0x80;       //????????SMOD
   SCON = 0x50;        //8???,?????
   BRT = RELOAD_COUNT;       //?????????????
   AUXR |= 0x04;       //???????????Fosc,?1T
   AUXR |= 0x01;       //??1??????????????????
   AUXR |= 0x10;       //??????????
   ES     =   1;    //??????
   EA     =   1;    //????
   TI     =   1;
}

void UART_send_byte(uint8 byte)
{
   ES     =   0;  //?????
   TI     =   0;  //??????????????
   SBUF   =   byte;
   while(TI ==0); //??????
   TI     =   1;  //??????????????
   ES     =   1;  //??????
}

void UART_send(uint8 *Buffer, uint8 Length)
{
   while(Length != 0)
   {
      UART_send_byte(*Buffer);
      Buffer++;
```

```
      Length--;
    }

}

void Communication_Decode(void)
{
   if(buffer[0]==0x00)
   {
      switch(buffer[1])
      {
         case 0x01:MOTOR_GO_FORWARD; return;
         case 0x02:MOTOR_GO_BACK;    return;
         case 0x03:MOTOR_GO_LEFT;    return;
         case 0x04:MOTOR_GO_RIGHT;   return;
         case 0x00:MOTOR_GO_STOP;    return;
         default: return;
      }
   }
   else if(buffer[0]==0x01)
   {
      if(buffer[2]>180)
         return;
      switch(buffer[1])
      {
         case 0x01:se_timer[0]=buffer[2]; return;
         case 0x02:se_timer[1]=buffer[2]; return;
         case 0x03:se_timer[2]=buffer[2]; return;
         case 0x04:se_timer[3]=buffer[2]; return;
         case 0x05:se_timer[4]=buffer[2]; return;
         default : return;
      }
   }
   else
   {
      return;
   }
}

void UART_Interrupt_Receive(void) interrupt 4
{
   static uint8 i;

   if(RI==1)
   {
      RI = 0;
      if(rec_flag==0)
      {
         if(SBUF==0xff)
         {
```

```
            rec_flag=1;
            i=0;
        }
    }
    else
    {
        if(SBUF==0xff)
        {
            rec_flag=0;
            if(i==3)
            {
                Communication_Decode();
            }
            i=0;
        }
        else
        {
            buffer[i]=SBUF;
            i++;
        }
    }
  }
  else
  {
    TI = 0;
  }
}
```