

```

/*-----+
| SPI data transfer for MC145170 VHF frequency synthesiser |
| 8051 is Master; MC145170 is Slave. |
| The following SPI-defined pins are used. (write only, so only 3 wires are needed.) |
| CLK: Serial Data Clock Input (pin 7 of MC145170) |
| DIN: Serial Data input (pin 5 of MC145170) |
| ENB: Active-Low Enable Input (pin 6 of MC145170) |
+-----*/

#include <reg51.h>
#include <stdio.h>

sbit CLK = P1^0; // serial data clock
sbit ENB = P1^1; // active low enable
sbit DIN = P1^2; // serial data input
void cdata(char c_Reg);
void rdata(char r_RegH,r_RegL);
void ndata(char n_RegH,n_RegL);

/*
A sample for program SPI chip MC145170
*/
void main(void){
/* C7=0 Unchanged polarity C6=0 Enable detector B C5=1 Enable Lock detector */
/* C4=1 C3=0 C2=0 OSC/8 C1=1 Enable Fv output C0=1 Enable Fr output */
cdata(0X33);
rdata(0x00,0x78); // N=1000 */
ndata(0X03,0xe8); // R=120 */
while(1); // Loop ever
}

/*
Function for Register C data transfer, 8 data elements
*/
void cdata(char c_Reg){
char c[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
int i;

ENB=1;
CLK=0; // Serial Data Clock starts off low.
ENB=0; // Lowering ENB begins the data transfer.
for(i=0;i<8;i++) // Eight bits to transfer.
{ // send a Data bit from c_Reg bit 7 to
DIN=c_Reg&c[i]; // bit 0.
CLK=1; // Generate Serial Data Clock rising edge.
CLK=0; // Generate Serial Data Clock falling edge.
}
ENB=1; // Setting ENB terminate the data transfer.
}

/*
Function for Register R data transfer, 15 data elements
*/
void rdata(char r_RegH,r_RegL){
char c[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
int i;

ENB=1;
CLK=0; // Serial Data Clock starts off low.
ENB=0; // Lowering ENB begins the data transfer.
for(i=1;i<8;i++) // Seven bits to transfer.
{ // send a Data bit from r_Reg bit 6 to
DIN=r_RegH&c[i]; // bit 0.
CLK=1; // Generate Serial Data Clock rising edge.
CLK=0; // Generate Serial Data Clock falling edge.
}
for(i=0;i<8;i++) // Eight bits to transfer.
{ // send a Data bit from r_Reg bit 7 to
DIN=r_RegL&c[i]; // bit 0.
CLK=1; // Generate Serial Data Clock rising edge.
CLK=0; // Generate Serial Data Clock falling edge.
}
ENB=1; // Setting ENB terminate the data transfer.
}

/*
Function for Register N data transfer, 16 data elements
*/
void ndata(char n_RegH,n_RegL){

```

```
char c[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
int i;

ENB=1;
CLK=0;                                     // Serial Data Clock starts off low.
ENB=0;                                     // Lowering ENB begins the data transfer.
for(i=0;i<8;i++)                           // Eight bits to transfer.
{                                           // send a Data bit from n_Reg bit 7 to
    DIN=n_RegH&c[i];                     // bit 0.
    CLK=1;                               // Generate Serial Data Clock rising edge.
    CLK=0;                               // Generate Serial Data Clock falling edge.
}
for(i=0;i<8;i++)                           // Eight bits to transfer.
{                                           // send a Data bit from n_Reg bit 7 to
    DIN=n_RegL&c[i];                     // bit 0.
    CLK=1;                               // Generate Serial Data Clock rising edge.
    CLK=0;                               // Generate Serial Data Clock falling edge.
}
ENB=1;                                     // Setting ENB terminate the data transfer.
}
```