EE 500 Embedded Systems and Applications

Course Project

Go Bang Chess Game

Members: Yuxin Zhang ID: 50167183
Ke Wei ID: 50166207
Instructor: Praveen Meduri

12/11/2015

Content

Description

Go bang is a game of pure strategy type of chess. Two players usually put black or white pieces on the intersection of straight line and horizontal line. The player who is the first to get 5 same color pieces continuously will win the game. At the beginning, two players choose one color chess and then compete with each other on an empty board. Black chess is first, then white chess is second. Each chess is only at a time. Chess needs to be put on the point of intersection of straight line and horizontal line. After the chess is put, it is not allowed to move the chess to another place, not to take off the board or pick up them. The first black chess can be put at any point of the intersection. Then we can see on our board. The first interface is our start mode. We make some decorations. For example, we use chess board as the background. We have a headline and two big chess. Then we press the button to jump to another interface which is related to our information. Press the button continuously. This part is about the principle how to play this game. Let us start the game. The first player is black chess. We can move joystick to control the red cursor up, down, left and right. We can also find on the right side of the board, we have two timers for both players. Black timer is counting, because it is the time for black chess to put. First player can press INT0 to put chess. And then it is white chess's turn. The chess can be put wherever we want except the place where the chess has already exists. At the down corner of right side, there is a notice which will tell us the information about our step, like up, down, left, right and error. At the end of the game, we will have 4 different situation of victory, which are forming same color unbroken 5 chess of the line horizontally, vertically, left to right diagonally and right to left diagonally. Our design can judge all of these situations. Finally when one of two players wins the game, it will appear at the right down corner of the board, like "Black Wins!" or "White wins!".

Development Environment:

Software in the computer:

Keil5 Version.

Hardware of the Embedded System:

LandTiger V2.0 LPC1768 Development Board.

Implementation

1. #define

```
#define uchar unsigned char
    // make it easy to type
#define uint unsigned int
#define ChessX   ((Cursor_x-14)/15)
    //convert chess position on the screen into chess map's position
#define ChessY   ((Cursor_y-14)/15)
```

2. Argument Design

```
uchar ChessMap[15][15]={0};
    //A 2D array carries the situation of chess spot, 0 means no chess, 1 means black, 2 means
  white
uint Cursor_x=119, Cursor_y=119;
    //Initial position of the cursor on the screen
uchar Player=1;
    //Player mark, represents the active player, 1=black, 2=white
uchar Joystick=0;
    //represents the movement of Joystick, 0=blank, 1=select, 2=down, 3=lest, 4=right, 5=up
uchar INT0=0;
    //represents the movement of INT0, 0=blank,1=pressing
uint Timer_Joystick=0, Timer_INT0=0;
    //Used in Timer to count the time that the keys are pressing to confirm the pressing
uint Timer_Time=0;
    //Used in Timer to count the time, increase 1 per second
uchar Timer_TimeSign=0;
    //use in Timer to mark the situation of time counter, 0=disable,1=enable,2=timeup
uint BlackTime=0, WhiteTime=0;
    //stores the time of each player
uchar alert=0;
    //used in alert function to store the situation and information of notice. 1=Joystick up,
 2=Joystick down, 3=Joystick left, 4=Joystick right, 5=ERROR, 6=Black Win, 7=White win.
uchar FinalWinner=0;
    //gives who is the Final Winner, 1=black, 2=white
```

3. INT0 and Joystick initialization

```
void JoyIntInit (void)
{
    LPC_PINCON->PINSEL3 &= ~((1 <<  18)|(1 <<  19)|(1 <<  20)|(1 <<  21)|(1 <<  22)|(1 <<
23)|(1 <<  24)|(1 <<  25)|(1 <<  26)|(1 <<  27));
    //select P1.25~P1.29 as GPIO
     LPC_GPIO1->FIODIR &= ~((1<<25)|(1<<26)|(1<<27)|(1<<28)|(1<<29));
    //set P1.25 SelectP1.26 DownP1.27 LeftP1.28 RightP1.29 Up as input
     LPC_PINCON->PINSEL4 &= ~((1 <<  21)|(1 <<  22));
    //select P2.10 as GPIO, which is INT0
     LPC_GPIO2->FIODIR &= ~(1<<10);
    //set P2.10 as input
}
```

4. Timer Initialization

```
void Timer0Init()
{
     LPC_SC->PCONP |= 1 << 1;
    // Power up Timer 0
     LPC_SC->PCLKSEL0 |= 1 << 2;
    // Clock for timer = CCLK, i.e., CPU Clock
     LPC_TIM0->MR0 = 1 << 17;
    // 17: give a value suitable for the check and refresh
     LPC_TIM0->MCR |= 1 << 0;
    // Interrupt on Match 0 compare
     LPC_TIM0->MCR |= 1 << 1;
    // Reset timer on Match 0
     LPC_TIM0->TCR |= 1 << 1;
    // Manually Reset Timer 0 (forced);
     LPC_TIM0->TCR &= ~(1 << 1);
    // Stop resetting the timer
}
```

5. Draw and Erase cursor

```
void DrawCursor(int x, int y)
{
      GLCD_SetTextColor(Red);
           //cursor's color is red
           CRIS_draw_line( x+1, y+1, x+1, y+5);
           CRIS_draw_line( x+1, y+1, x+5, y+1);
           CRIS_draw_line( x-1, y+1, x-5, y+1);
           CRIS_draw_line( x-1, y+1, x-1, y+5);
           CRIS_draw_line( x-1, y-1, x-5, y-1);
           CRIS_draw_line( x-1, y-1, x-1, y-5);
           CRIS_draw_line( x+1, y-1, x+5, y-1);
           CRIS_draw_line( x+1, y-1, x+1, y-5);
}
```

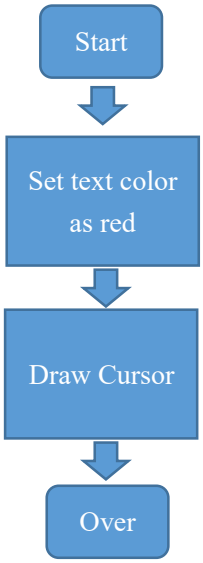
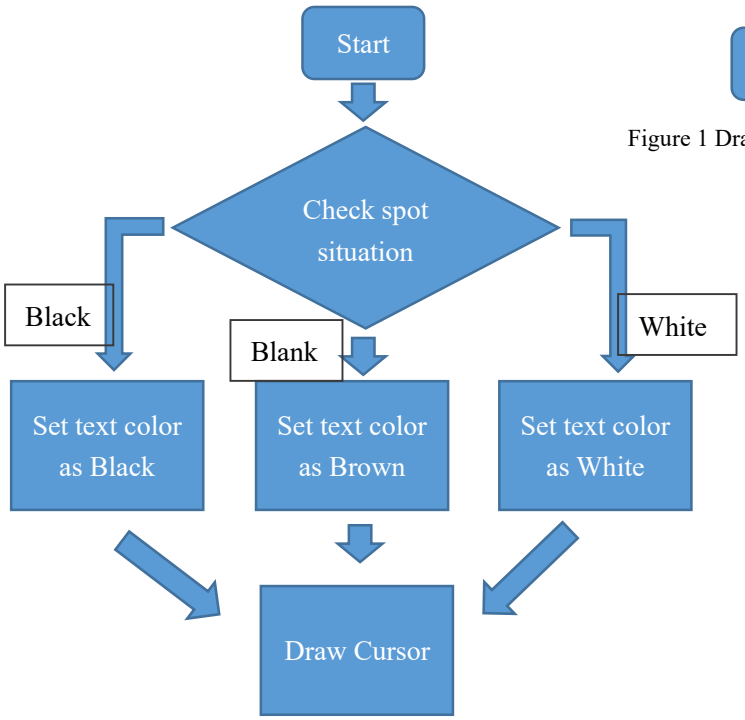


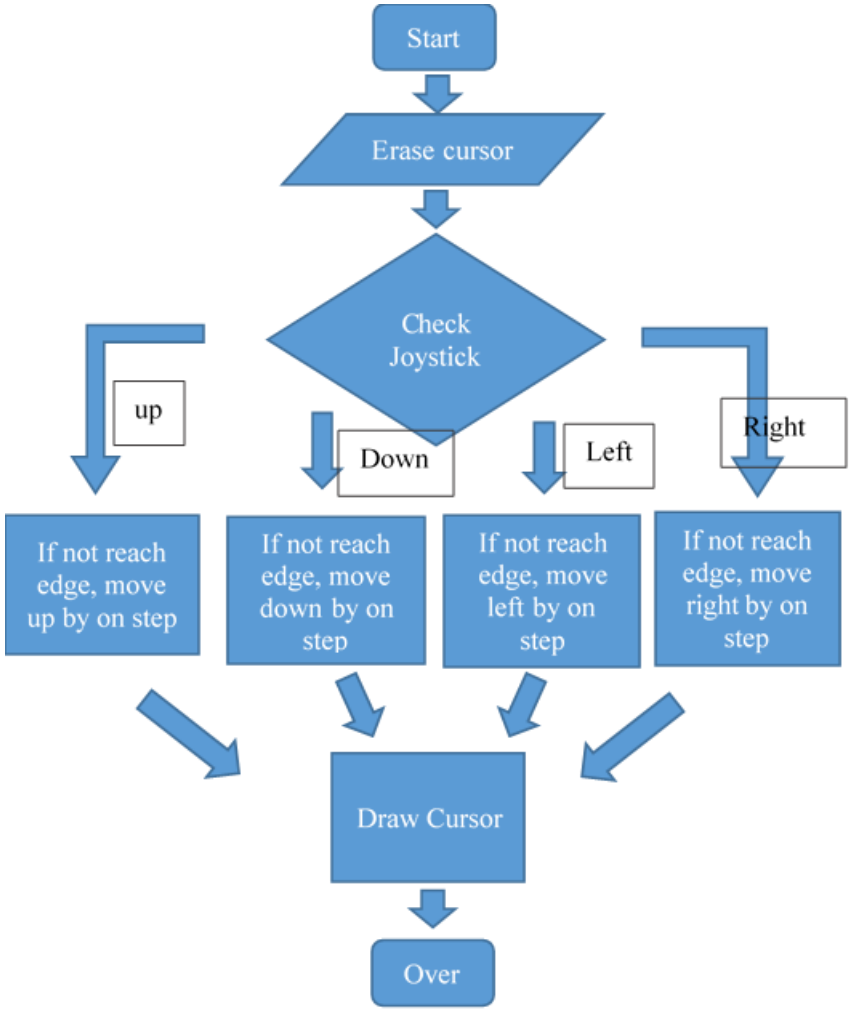
Figure 1 DrawCursor(int x, int y)

```
void EraseCursor(int x, int y)
```



6. Move cursor by Joystick

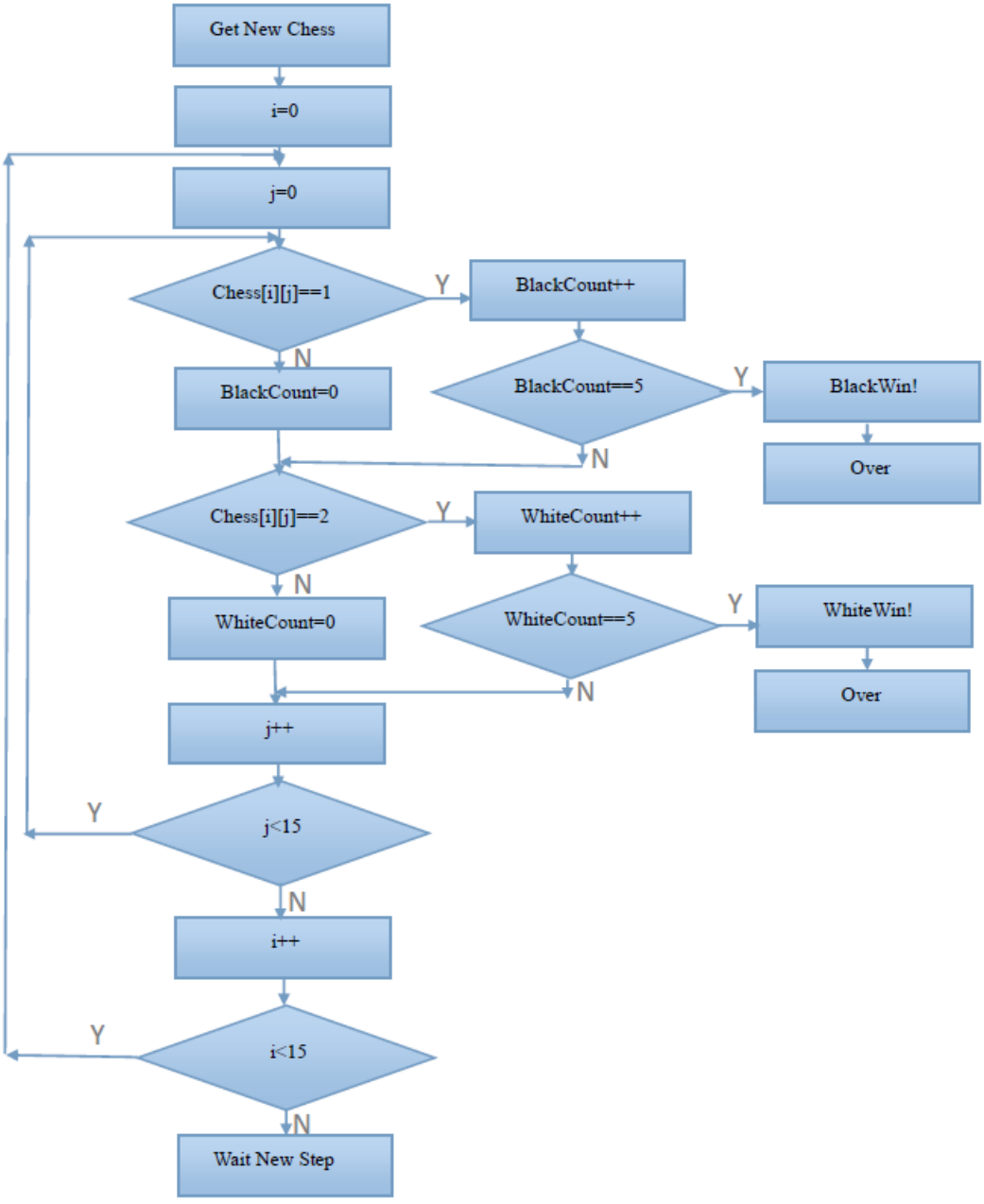
void Move_Cursor(char n)



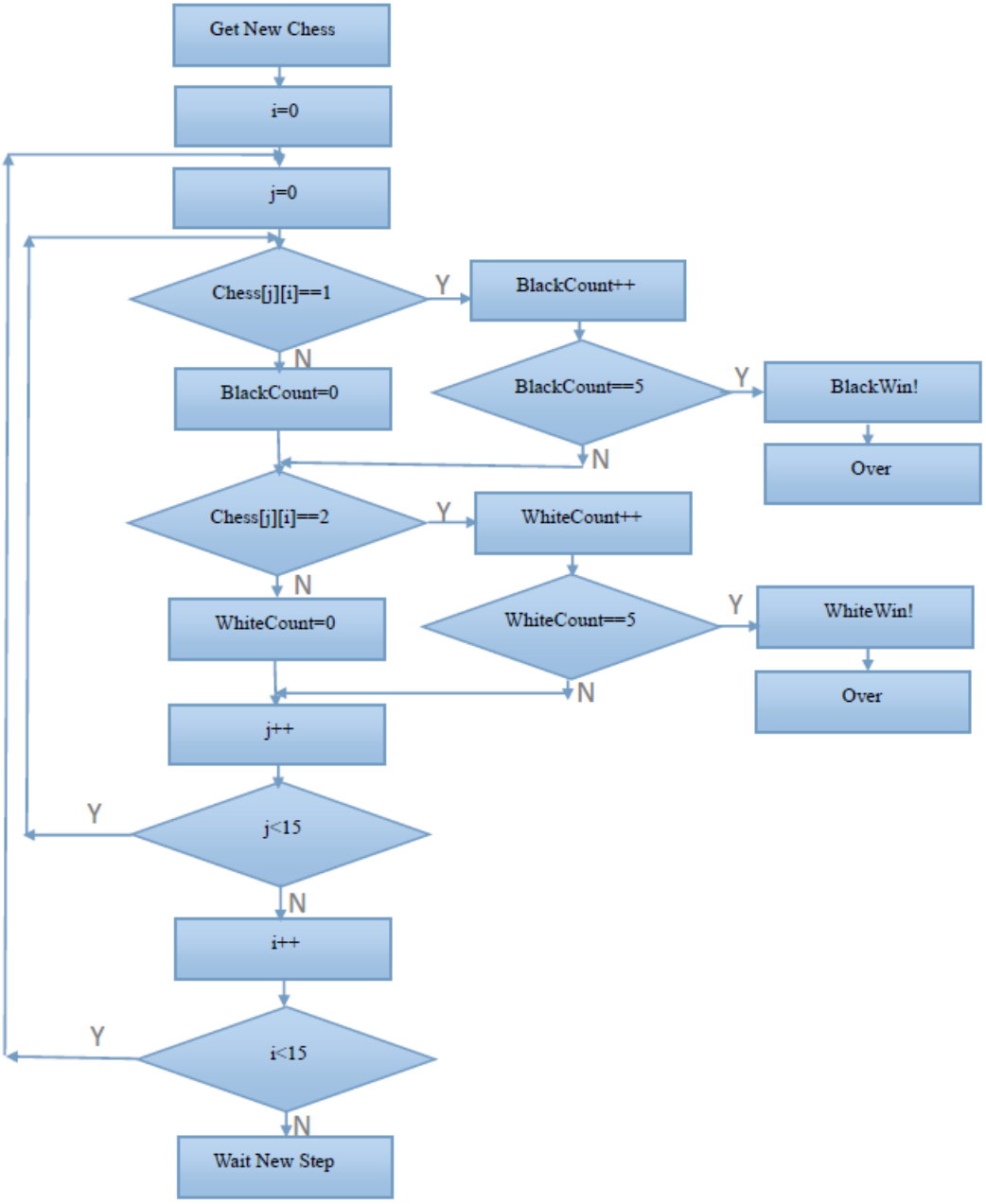
7. The algorithm to Judge winner

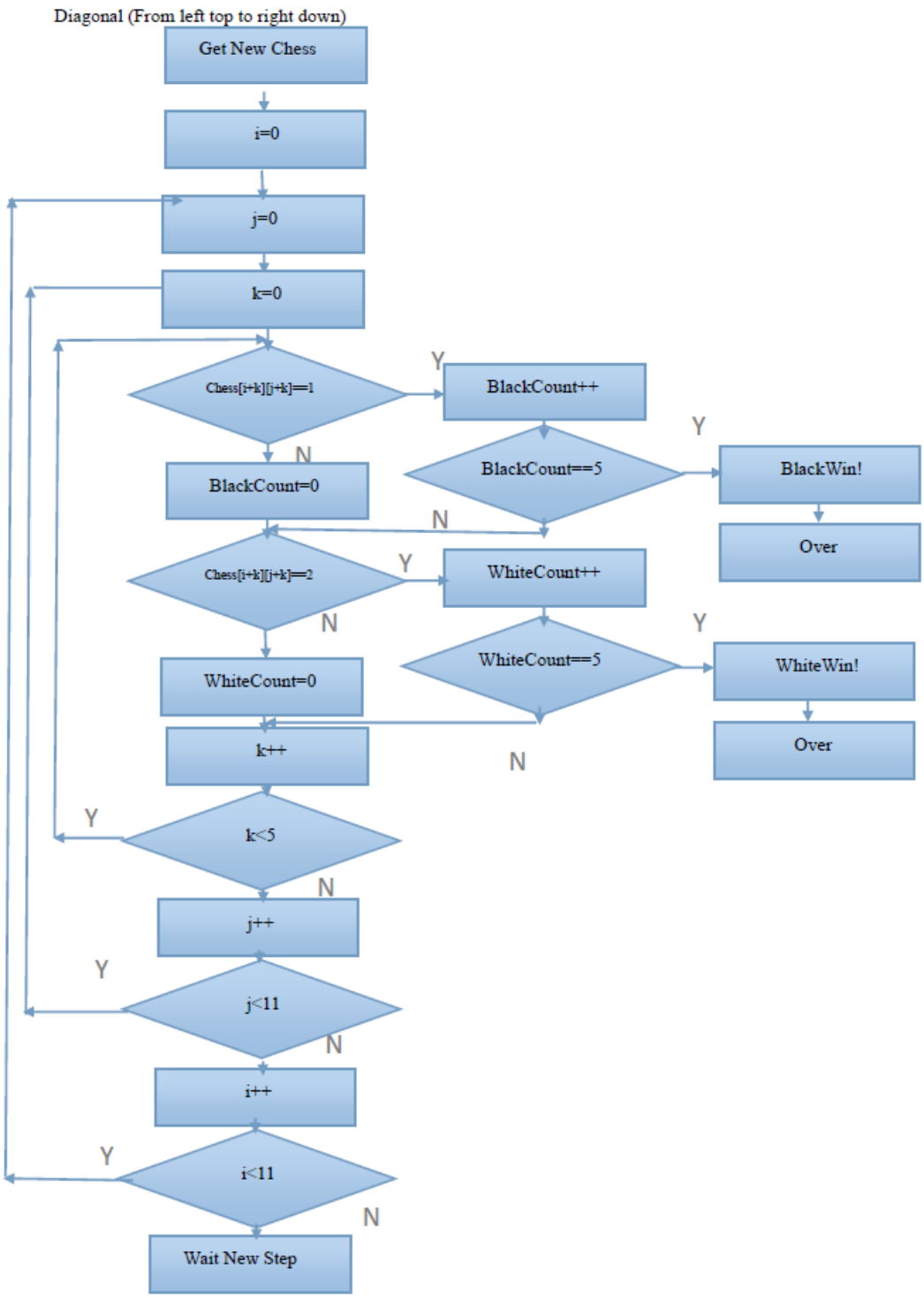
uchar JudgeWinner()

Horizontal

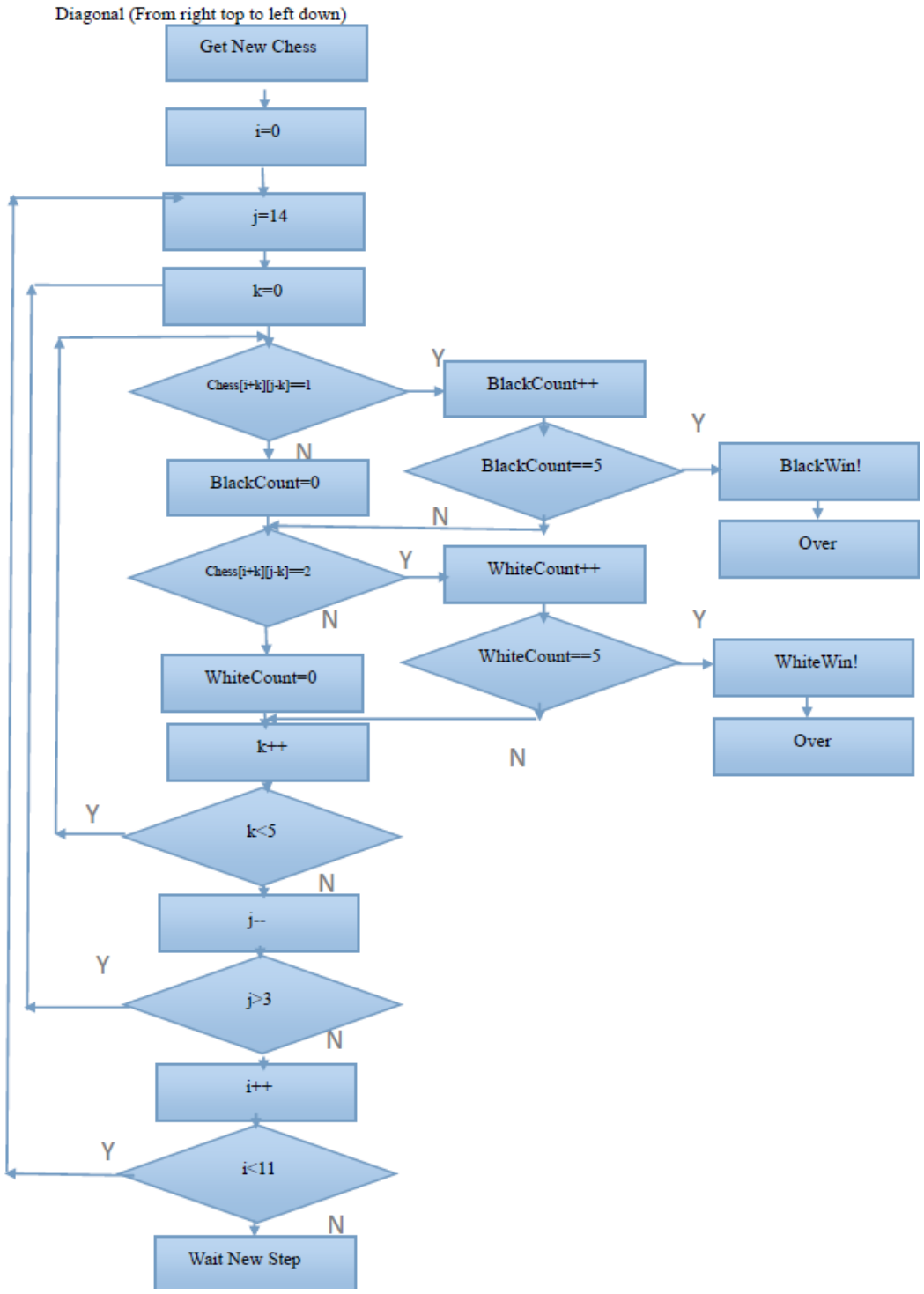


Vertical


Get New Chess
i=0
j=0
Chess[j][i]==1
Y
BlackCount++
N
BlackCount=0
BlackCount==5
Y
BlackWin!
Over
N
Chess[j][i]==2
Y
WhiteCount++
N
WhiteCount=0
WhiteCount==5
Y
WhiteWin!
Over
N
j++
Y
j<15
N
i++
Y
i<15
N
Wait New Step

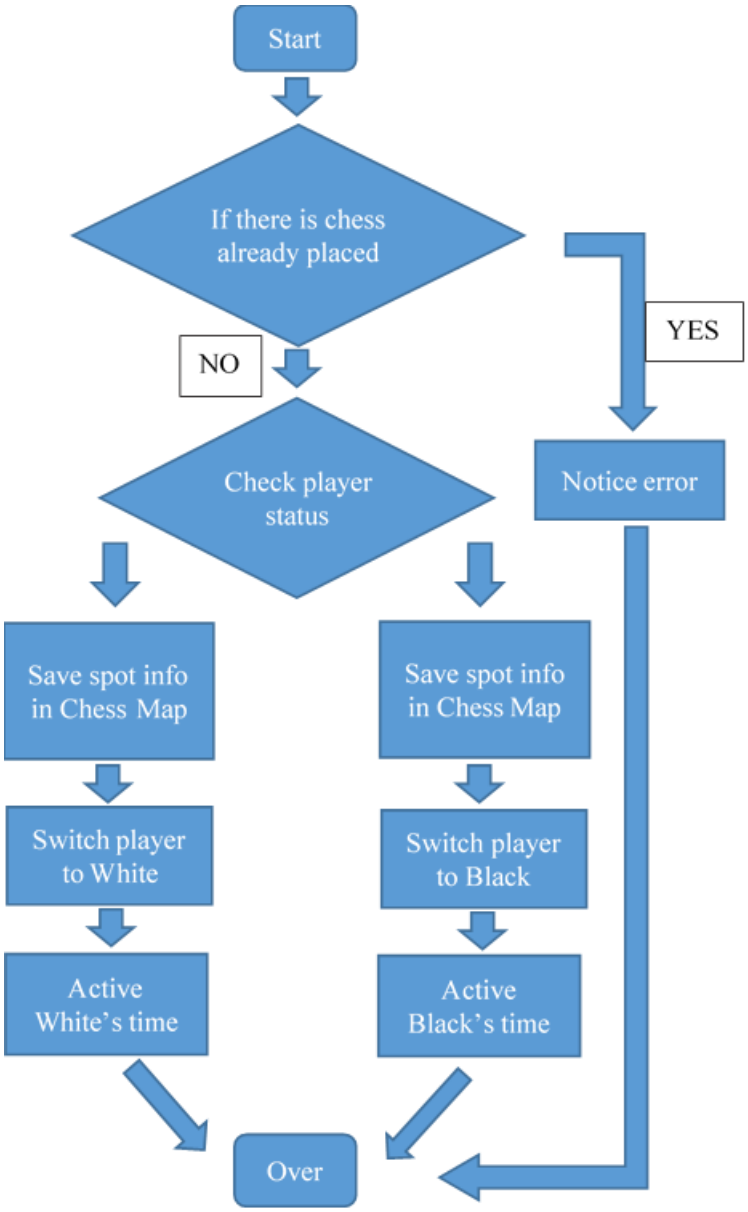

Diagonal (From left top to right down)
Get New Chess
i=0
j=0
k=0
Chess[i+k][j+k]==1
Y
BlackCount++
BlackCount==5
Y
BlackWin!
Over
N
BlackCount=0
N
Chess[i+k][j+k]==2
Y
WhiteCount++
WhiteCount==5
Y
WhiteWin!
Over
N
WhiteCount=0
k++
N
k<5
Y
N
j++
j<11
Y
N
i++
i<11
Y
N
Wait New Step

Diagonal (From right top to left down)



8. Place chess on a spot

void PlaceChess()



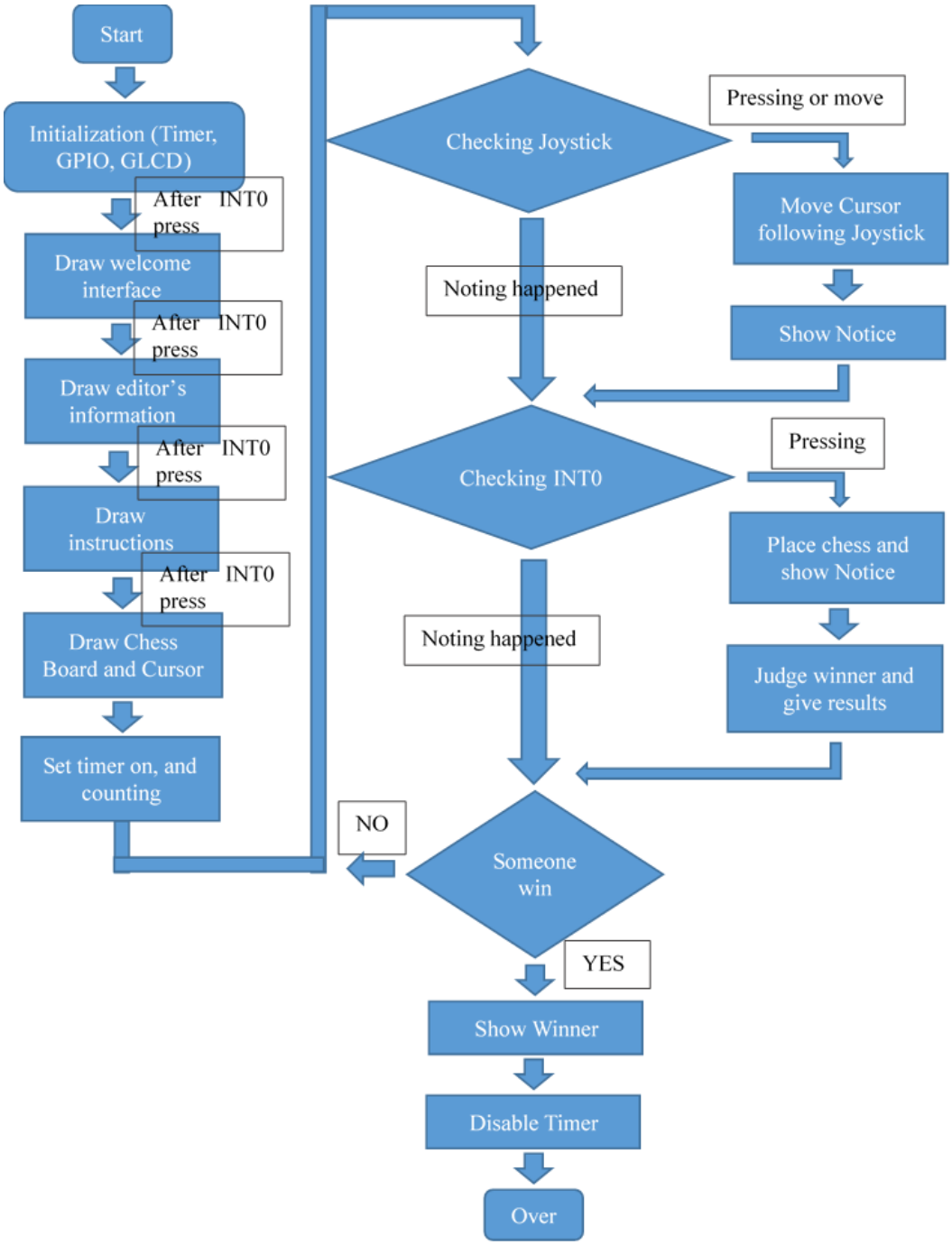
9. TIMER0 setting

```
void TIMER0_IRQHandler(void)
{
    uchar JoystickMove=0,INT0Move=0;
        if ( (LPC_TIM0->IR & 0x01) == 0x01 ) // if MR0 interrupt
     {
        LPC_TIM0->IR |= 1 << 0;
    // Clear MR0 interrupt flag
        JoystickMove = (LPC_GPIO1->FIOPIN >> 25) & (0x1f);
    //get the Value of Joystick
        INT0Move = (LPC_GPIO2->FIOPIN >> 10) & 0x1;
    //get the value of INT0
      /*****************Joystick and INT0 Controlling Part********************/

                    if(JoystickMove==0x1f)              //Joystick blank
                    {
                        Timer_Joystick=0;
                    //once find joystick is blank, reset timer_Joystick
              Joystick=0;
                    }
                  if((JoystickMove&0x01)==0)            //Joystick is selecting
                    {
                        Timer_Joystick+=1;
                            if(Timer_Joystick==60)      //delay for pressing confirmation
                            Joystick=1;                 //confirm Joystick is pressing
                    }
                    if((JoystickMove&0x02)==0)          //Joystick moving to the down
            {
                        Timer_Joystick+=1;
                            if(Timer_Joystick==60)      //delay for pressing confirmation
                            Joystick=3;                 //confirm Joystick is moving down
                    }
                    if((JoystickMove&0x04)==0)          //Joystick moving to the left
            {
                        Timer_Joystick+=1;
                            if(Timer_Joystick==60)      //delay for pressing confirmation
                            Joystick=5;                 //confirm Joystick is moving left
                    }

            if((JoystickMove&0x08)==0)                  //Joystick moving to the right
            {
                            Timer_Joystick+=1;
                            if(Timer_Joystick==60)      //delay for pressing confirmation
                            Joystick=2;                 //confirm Joystick is moving right
                    }

                    if((JoystickMove&0x10)==0)          //Joystick moving to the up
            {
                            Timer_Joystick+=1;
                            if(Timer_Joystick==60)      //delay for pressing confirmation
                            Joystick=4;                 //confirm Joystick is moving up
                    }
            if(INT0Move==0)             //Pressing INT0
                    {
                        Timer_INT0+=1;
                  if(Timer_INT0==60)                    //delay for pressing confirmation
                                INT0=1;                 //confirm, set as pressing

                    }
                    else if(INT0Move==1)                //INT0 do not pressing
                    {
```

```
                Timer_INT0=0;
                INT0=0;                         //set as blank
            }
            /*****************BlackTime,WhiteTime**************/

            if(Timer_TimeSign==1)               //checking time counting enable
            {
                Timer_Time+=1;
                if(Timer_Time==760)             //time up
                    Timer_TimeSign=2;           //set sign, time to count
            }
            if(Timer_TimeSign==2)               //checking time if time up to count
            {
                if(Player==1)                   //if player is black
                {
                    BlackTime+=1;               //add black time
                    GLCD_SetBackColor(Brown);
                    GLCD_SetTextColor(Yellow);
                    TimeRefresh(BlackTime,3,15); //show new time
                    Timer_TimeSign=1;           //reset time sing to enable timer
                    Timer_Time=0;               //reset time counter
                }
                else if(Player==2)
                {
                    WhiteTime+=1;               //add black time
                    GLCD_SetBackColor(Brown);
                    GLCD_SetTextColor(Yellow);
                    TimeRefresh(WhiteTime,6,15);        //show new time
                    Timer_TimeSign=1;           //reset time sing to enable timer
                    Timer_Time=0;               //reset time counter
                }
            }
        }
    }
```

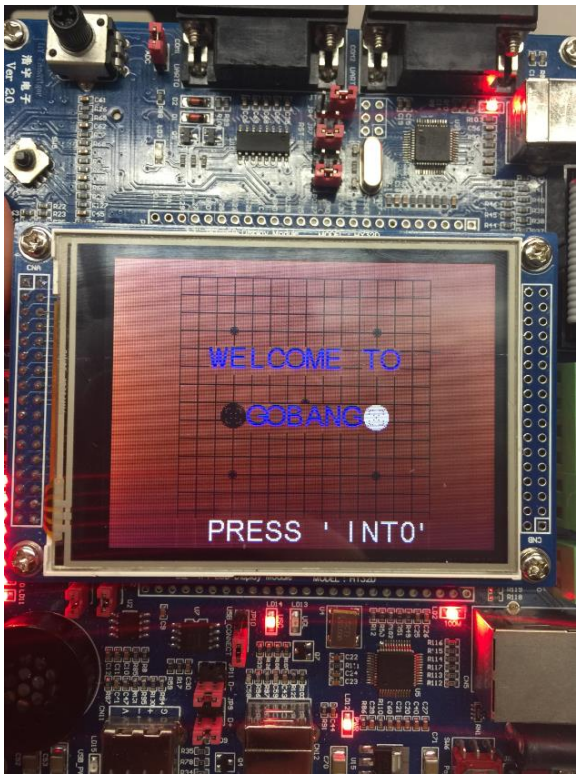
10.Main function

int main (void)



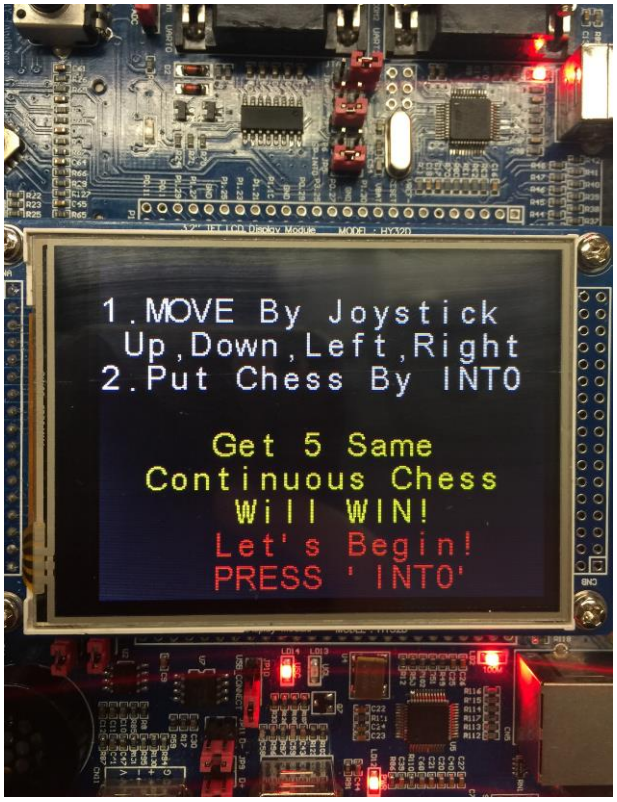
Experimental Results

This is our start mode.

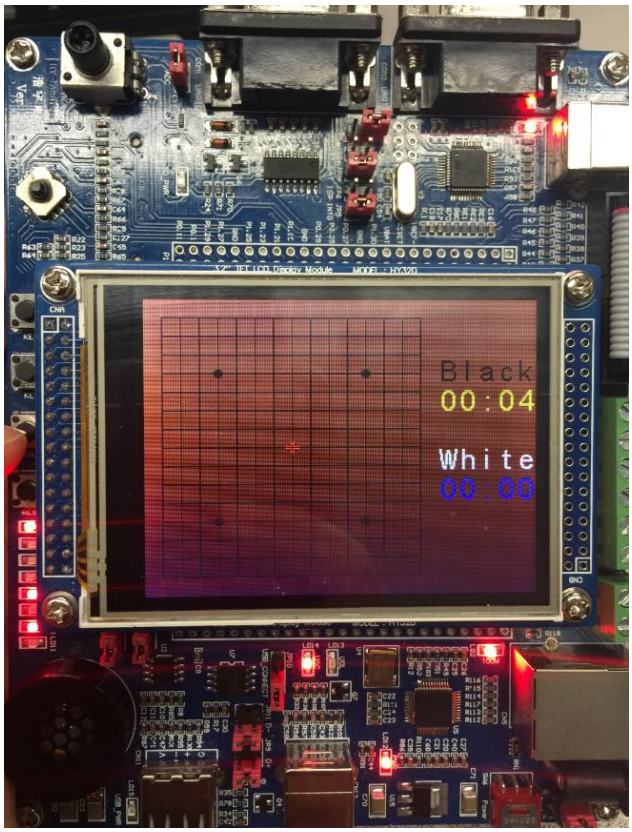


Press INT0, we will get the surface of our group information

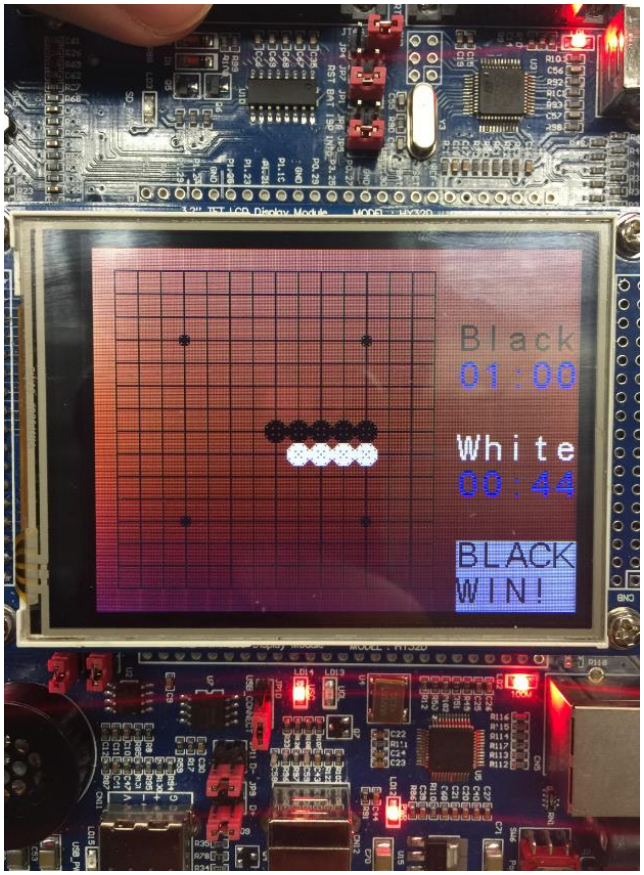
Continue to press INT0, we will get the surface of introduction.



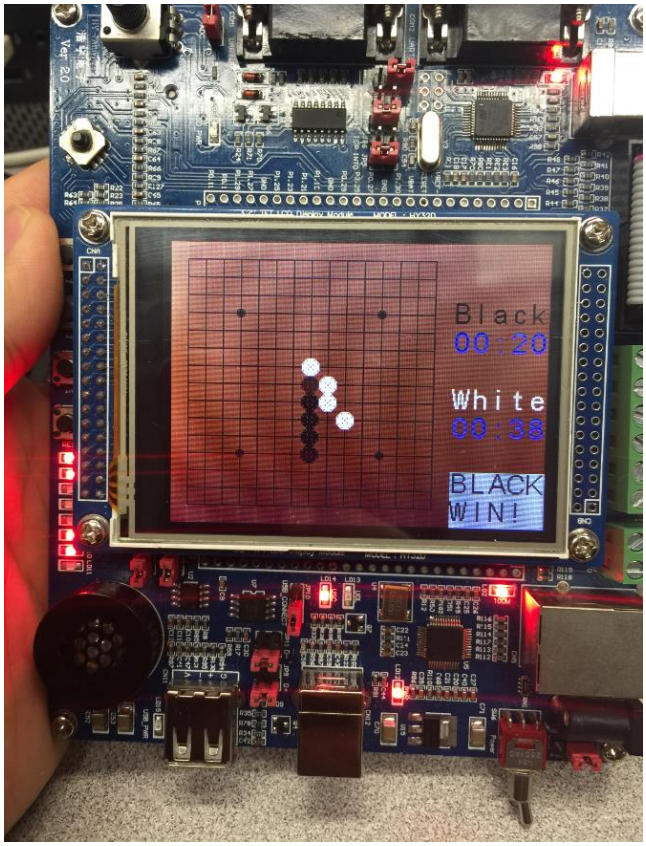
Now we can start our game. Press INT0 button to put chess.



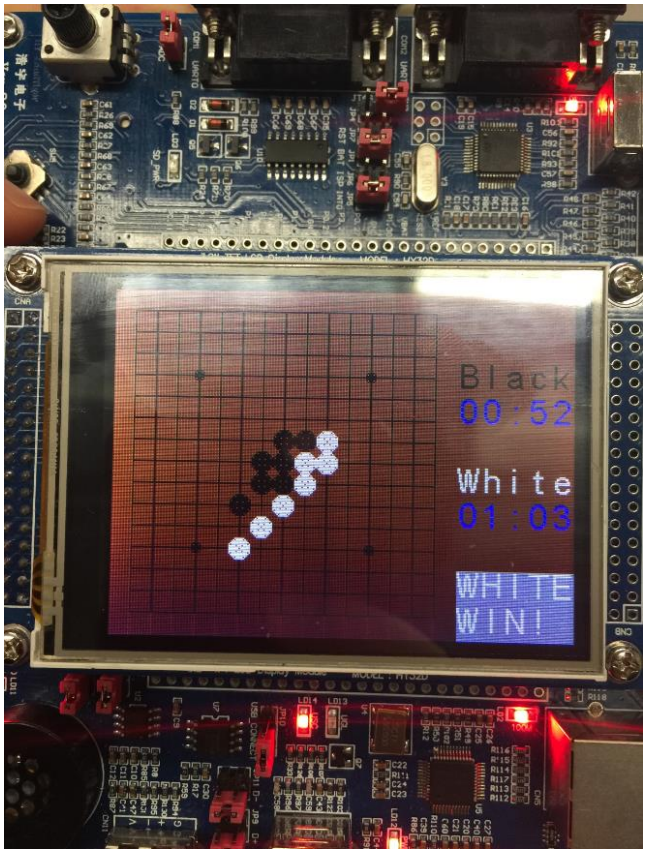
Horizontal Victory Judgment



Vertical Victory Judgment



First Diagonal Victory Judgment (From right up to left down)



First Diagonal Victory Judgment (From left up to right down)