Mouse Programming with C

Submitted by bOtskOOl on Thu, 06/18/2009 - 17:35

Mouse Programming is a topic which every C programmer from beginner to professional needs to have in his toolbox to have a cutting edge.It will be used almost everywhere. It will embeded in games programming to commerical valued applications.

This tutorial is written Turbo C++ 3.0 IDE and install in folder C:\TC. I recommend to use same IDE and settings to avoid any uncompatibility.

Basic Funda

Before we start programming we must first understand some principles on which mouse programming is based.

First thing you must know how to tell a mouse to do anything. In actual we do not communicate with mouse directly but through the driver provided. We use "Interrupts" to get access to this driver. Each device provide by computer has its own port and more or less we access these ports.

Each device has a unique port which is a hexadecimal value and value is designed to be machine independent enhancing portability of program.

Mouse has port 0X33 attached to it and similarly keyboard has attach port 0X60.

We also make use of address registers. These are basically UNION of type REGS defined in "dos.h". We use two registers to communicate to a device driver one for input and one for output.

We send value to device driver through the input register and recieve information in it embedded in output register.

AX Register

We can access various mouse functions using different values of AX input Register and passing those values to mouse port using a interrupt.

The Functions are listed below - Here AX, BX, CX and DX are members of UNION REGS and more or less integers.

Input	Function Performed	Returns
AX = 0	Get Mouse Status	AX Value = FFFFh support is available. AX Value = 0 ,support is not available.

AX = 1	Show Mouse Pointer	Nothing
AX = 2	Hide Mouse Pointer	Nothing
AX = 3	Mouse Position	CX = Mouse X Coordinate DX = Mouse Y Coordinate
Ax = 3	Mouse Button Press	BX = 0 No Key Is Pressed BX = 1 Left Button is Pressed BX = 2 Right Button is Pressed BX = 3 Centre Button is Pressed
Ax = 7	Set Horizontal Limit	Nothing
CX = MaxX1		
DX =MaxX2		
Ax = 8	Set Vertical Limit	Nothing
CX = MaxX1		
DX =MaxX2		

Detecting Mouse

Before you start your mouse program you should always check whether the mouse programming is supported or not.

If somehow mouse fails to initialise you should always make sure that either program terminates or employ a error handling approach that maybe shift to keyboard interface .

To do mouse programming you must include <dos.h>. We use a function called int86() to access "interupts".

To detect mouse we use a function name detect_mouse() which has following code -

```
#include <dos.h>

union REGS in, out;

void detect_mouse ()
 {
  in.x.ax = 0;
  int86 (0X33,&in,&out);
  if (out.x.ax == 0)
  printf ("\nMouse Failed To Initialize");
  else
  printf ("\nMouse was Succesfully Initialized");
 }

int main ()

{
 detect_mouse ();
```

```
 return 0;
}
```

Showing and Hiding Mouse

Now first we show mouse on screen .

- Mouse works both in text mode and graphic mode.
- In text mode it looks like a square while in graphics mode it looks like a pointer.

Mouse Programming in Text Mode

It was produced from adding a function showmouse_text() to above code so code becomes -

```
#include <dos.h>

union REGS in, out;

void detect_mouse ()
{
 in.x.ax = 0;
 int86 (0X33,&in,&out);
 if (out.x.ax == 0)
 printf ("\nMouse Failed To Initialize");
 else
 printf ("\nMouse was Succesfully Initialized");
}

void showmouse_text ()

{
 in.x.ax = 1;
 int86 (0X33,&in,&out);
}

int main ()

{
 detect_mouse ();
 showmouse_text ();
 getch ();
 return 0;
}
```

Mouse Programming in Graphics Mode

This is achieved using a function showmouse_graphics() added to above code while removing showmouse_text() from main.

```
#include <dos.h>
#include <graphics.h>

union REGS in, out;

void detect_mouse ()
{
 in.x.ax = 0;
 int86 (0X33,&in,&out);
```

```
 printf ("\nMouse Failed To Initialize");
 else
 printf ("\nMouse was Succesfully Initialized");
}

void showmouse_text ()
{
 in.x.ax = 1;
 int86 (0X33,&in,&out);
}

void showmouse_graphics ()
{
 int gdriver = DETECT, gmode, errorcode;
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
 in.x.ax = 1;
 int86 (0X33,&in,&out);
 getch ();
 closegraph ();
}

int main ()

{
 detect_mouse ();
 showmouse_graphics ();
 getch ();
 return 0;
}
```

Next we do realtively simple task of hiding mouse using a function hide_mouse() as shown below
-

```
#include <dos.h>
#include <graphics.h>

union REGS in, out;

void detectmouse ()
{
 in.x.ax = 0;
 int86 (0X33,&in,&out);
 if (out.x.ax == 0)
 printf ("\nMouse Failed To Initialize");
 else
 printf ("\nMouse was Succesfully Initialize");
}

void showmouse_text ()

{
  in.x.ax = 1;
  int86 (0X33,&in,&out);
}

void showmouse_graphics ()

{
 int gdriver = DETECT, gmode, errorcode;
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
 in.x.ax = 1;
 int86 (0X33,&in,&out);
 getch ();
 closegraph ();
```

```
void hide_mouse ()
{
 in.x.ax = 2;
 int86 (0X33,&in,&out);
}

int main ()

{
 detect_mouse ();
 showmouse_graphics ();
 hide_mouse ();
 getch ();
 return 0;
}
```

Detecting Input

We will now work on a important aspect of mouse programming "Detecting Clicks" i.e. Taking Inputs.

We make use of an aditional function known as kbhit ( ). This functions returns zero till any keypress and when a key is press it returns 1.

kbhit() is used to run an infinite while loop.

For detecting mouseclicks we use a function called detect() which displays on screen the respective button clicked. Press any keyboad key to exit the loop.

```
#include <dos.h>

#include <graphics.h>

union REGS in, out;

void detect_mouse ()

{
 in.x.ax = 0;
 int86 (0X33,&in,&out);
 if (out.x.ax == 0)
 printf ("\nMouse Failed To Initialize");
 else
 printf ("\nMouse was Succesfully Initialized");
}

void showmouse_text ()

{
 in.x.ax = 1;
 int86 (0X33,&in,&out);
}

void showmouse_graphics ()

{
 int gdriver = DETECT, gmode, errorcode;
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
 in.x.ax = 1;
 int86 (0X33,&in,&out);
 getch ();
 closegraph ();
}

void hide_mouse ()
```

```
 in.x.ax = 2;
 int86 (0X33,&in,&out);
}

void detect ()

{
 while (!kbhit () )
  {
   in.x.ax = 3;
   int86 (0X33,&in,&out);
   if (out.x.bx == 1) printf ("Left");
   if (out.x.bx == 2) printf ("Right");
   if (out.x.bx == 3) printf ("Middle");
   delay (200); // Otherwise due to quick computer response 100s of words will get print
  }
}

int main ()

{
  detect_mouse ();
  showmouse_text ();
  detect ();
  hide_mouse ();
  getch ();
  return 0;
}
```

Mouse Coordinates

We can obtain the coordinates of the mouse using same service 3 but using different elments of the union .

This function has a prime use in games programming, application designing and GUI development. Different decisions are taken on same left button click, its the postion of click that matters.

BX element of output registers stores the X Coordinate of the postion of mouse at time of calling function.

CX element of output registers stores the Y Coordinate of the postion of mouse at time of calling function.

Now we demonstrate the use of this function by modifying detect function above to display x and y coordinates on screen when left click is pressed.

Code will be as followed -

```
#include <dos.h>
#include <graphics.h>

union REGS in, out;

void detect_mouse ()
 {
  in.x.ax = 0;
  int86 (0X33,&in,&out);
  if (out.x.ax == 0)
  printf ("\nMouse Failed To Initialize");
  else
  printf ("\nMouse was Succesfully Initialized");
 }

void showmouse_text ()
 {
```

```
  int86 (0X33,&in,&out);
 }

void showmouse_graphics ()
 {
  int gdriver = DETECT, gmode, errorcode;
  initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
  in.x.ax = 1;
  int86 (0X33,&in,&out);
  getch ();
  closegraph ();
 }

void hide_mouse ()
 {
  in.x.ax = 2;
  int86 (0X33,&in,&out);
 }

void detect ()
 {
  while (!kbhit () )
    {
     int x,y;
     in.x.ax = 3;
     int86 (0X33,&in,&out);
     if (out.x.bx == 1)
       {
         x = out.x.cx;
         y = out.x.dx;
         printf ("\nLeft || X - %d  Y - %d", x, y);
       }
         if (out.x.bx == 2) printf ("\nRight");
         if (out.x.bx == 3) printf ("\nMiddle");
         delay (200); // Otherwise due to quick computer response 100s of words will get print
    }
}

int main ()

{
 detect_mouse ();
 showmouse_text ();
 detect ();
 hide_mouse ();
 getch ();
 return 0;
}
```

Restricting Mouse

We now restrict the mouse in particular rectangle .

We create a function called restrict which takes four paramters, two cartesian points each containing one x coordinate and one y coordinate.

First point mentions the top of the rectangle while second point mention the bottom bottom point of rectangle.

This service can be quite handy in special circumstances, for eg - if you want to restrict your mouse in one particular size window in GUI or In Games Programming.

Final code of the tutorial -

```
#include <dos.h>

#include <graphics.h>

#include <conio.h>

#include <stdio.h>


union REGS in, out;


void restrict (int x1,int y1,int x2, int y2)


{

 in.x.ax = 7;

 in.x.cx = x1;

 in.x.dx = x2;

 int86 (0X33,&in,&out);

 in.x.ax = 8;

 in.x.cx = y1;

 in.x.dx = y2;

 int86 (0X33,&in,&out);

}


void detect_mouse ()


{

 in.x.ax = 0;

 int86 (0X33,&in,&out);

 if (out.x.ax == 0)

 printf ("\nMouse Fail To Initialize");

 else

 printf ("\nMouse Succesfully Initialize");

}


void showmouse_text ()


{

 in.x.ax = 1;

 int86 (0X33,&in,&out);

}


void showmouse_graphics ()
```

```
{
 int gdriver = DETECT, gmode, errorcode;
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
 in.x.ax = 1;
 int86 (0X33,&in,&out);
 getch ();
 closegraph ();
}

void hide_mouse ()
{
 in.x.ax = 2;
 int86 (0X33,&in,&out);
}

void detect ()

{
 while (!kbhit () )
   {
    int x,y;
    in.x.ax = 3;
    int86 (0X33,&in,&out);
    if (out.x.bx == 1)
     {
       x = out.x.cx;
       y = out.x.dx;
       printf ("\nLeft || X - %d  Y - %d", x, y);
     }
    if (out.x.bx == 2) printf ("\nRight");
    if (out.x.bx == 3) printf ("\nMiddle");
    delay (200); // Otherwise due to quick computer response 100s

of words will get print
   }
}

int main ()
```

```
 detect_mouse ();
 showmouse_text ();
 restrict (100,100,500,500); // Change values here to create
```

different mouse movement space.

```
 detect ();
 hide_mouse ();
 getch ();
 return 0;
}
```

This was the most basic tutorial in Mouse Programming. We will be back with more advanced once very soon.

Please leave your comments, suggestions and report bugs(if any).

PS:- library graphics.h is incompatible with vista (working in Normal mode), because Vista doesnt support TC in fullscreen mode.

Average:

Your rating: None Average: 4.8 (13 votes)

Tags: C Tutorials

10100 reads

Comments

1 2 next › last »

Submitted by shashwat on Fri, 04/02/2010 - 23:55. #1

Member since:
18 February 2009
Last activity:
39 sec

Hi msdewana,

This code has been tested in Turbo C++ 3.0. Are you facing any problem?

Submitted by msdewana on Fri, 04/02/2010 - 20:02. #2

Member since:
2 April 2010
Last activity:
1 week 6 days

sir could this code is working on dev c++ compiler.please tell

Submitted by bOtskOOl on Sun, 11/01/2009 - 08:15. #3

Member since:
21 February 2009
Last activity:
12 weeks 4 days

Hi AugustineJees,

Check out the sample program below.. Mouse has been restricted in a circle with centre (300,100) and radius 100. Move your mouse slowly and check whether it works!

```
#include <dos.h>

#include <graphics.h>

#include <conio.h>

#include <stdio.h>

#include <math.h>


union REGS in, out;


void restrict (int x1,int y1,int x2, int y2)


{

 in.x.ax = 7;

 in.x.cx = x1;

 in.x.dx = x2;

 int86 (0X33,&in,&out);

 in.x.ax = 8;

 in.x.cx = y1;

 in.x.dx = y2;

 int86 (0X33,&in,&out);

}


void detect_mouse ()
```

```
{
 in.x.ax = 0;
 int86 (0X33,&in,&out);
 if (out.x.ax == 0)
 printf ("\nMouse Fail To Initialize");
 else
 printf ("\nMouse Succesfully Initialize");
}

void showmouse_text ()

{
 in.x.ax = 1;
 int86 (0X33,&in,&out);
}

void showmouse_graphics ()

{
 int gdriver = DETECT, gmode, errorcode;
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
 in.x.ax = 1;
 int86 (0X33,&in,&out);
 getch ();
 closegraph ();
}

void hide_mouse ()
{
 in.x.ax = 2;
 int86 (0X33,&in,&out);
}

void detect ()

{
 while (!kbhit () )
   {
    double x=100,y=100;
```

```
    int86 (0X33,&in,&out);
    if (out.x.bx == 0)
     {
        double x,y;
        //clrscr();
        x = out.x.cx;
        y = out.x.dx;
        double c = (((300-out.x.cx)*(300-out.x.cx))+((100-out.x.dx)*
(100-out.x.dx)));
        //printf ("\nCoordinates || X - %5.5f  Y - %5.5f distance - %5.5f",x,
y, c);
        if(c>=10000)
            if(x>300)
                  if(y>100)
            restrict(300,100,x,y);
                  else
            restrict(300,y,x,100);
            else
                  if(y>100)
            restrict(x,100,300,y);
                  else
            restrict(x,y,300,100);
        else
            restrict(0,0,500,500);
     }
    if (out.x.bx == 2) printf ("\nRight");
    if (out.x.bx == 3) printf ("\nMiddle");
   // delay (2); // Otherwise due to quick computer response 100s of
words will get print
   }
}

int main ()

{
 //clrscr();
 //printf("Press any key to start...");
 //getch();

 detect_mouse ();
 showmouse_text ();
```

```
 detect ();
 hide_mouse ();
 getch ();
 return 0;
}
```

Submitted by bOtskOOl on Sun, 11/01/2009 - 06:47. #4

Member since:
21 February 2009
Last activity:
12 weeks 4 days

Hi tanny,

Make the following changes in the program and see if it works. I have highlighted the changes that have been made in the program.

```
#include <dos.h>
#include <graphics.h>
#include <conio.h>
#include <stdio.h>

union REGS in, out;

void detect_mouse ()
 {
  in.x.ax = 0;
  int86 (0X33,&in,&out);
  if (out.x.ax == 0)
  printf ("\nMouse Failed To Initialize");
  else
  printf ("\nMouse was Succesfully Initialized");
 }

void showmouse_text ()
 {
   in.x.ax = 1;
   int86 (0X33,&in,&out);
 }

void showmouse_graphics ()
 {
   int gdriver = DETECT, gmode, errorcode;
   initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
   in.x.ax = 1;
```

```
  //getch ();
  //closegraph();



 }


void hide_mouse ()
 {
  in.x.ax = 2;
  int86 (0X33,&in,&out);
 }


void detect ()
 {
  while (!kbhit () )
   {
    int x,y;
    in.x.ax = 3;
    int86 (0X33,&in,&out);
    if (out.x.bx == 1)
     {
       x = out.x.cx;
       y = out.x.dx;
       printf ("\nLeft || X - %d  Y - %d", x, y);
     }
       if (out.x.bx == 2) printf ("\nRight");
       if (out.x.bx == 3) printf ("\nMiddle");
       delay (200); // Otherwise due to quick computer response 100s of
words will get print
   }
}


int main ()


{
 detect_mouse ();
 showmouse_graphics ();
 detect ();
 hide_mouse ();
 getch ();
```

```
return 0;

}
```

Submitted by pratik on Wed, 10/28/2009 - 08:35. #5

Member since: 28 October 2009
Last activity: 32 weeks 4 days

I think From here we can get help.

Submitted by AugustineJees on Thu, 10/22/2009 - 14:09. #6

Member since: 22 October 2009
Last activity: 33 weeks 2 days

Can any one know the program for restricting the mouse pointer inside the circle with a given radius and center. also i need ti restrict the pointer inside a hexagon.

The program should use the the interrupt int33h.

Thanks in advance,

Augustine Jees

Submitted by Mafia_nic on Thu, 10/15/2009 - 07:28. #7

Member since: 15 October 2009
Last activity: 34 weeks 2 days

This is way too good.

may be you should extend this mouse programming tutorial as Dragon suggested. Anyway keep up the good work.

Submitted by tanny on Wed, 09/16/2009 - 16:32. #8

Member since: 16 September 2009
Last activity: 38 weeks 3 days

hi,

The program worked extremely well!!

But I tried one thing.

In the mouse coordinates code, in the main function ,when i called for showmouse_graphics() function instead of showmouse_text() function ,it did not show the x and y coordinates of the click on every left click of the mouse.

Could you help me in this problem!!

I am running the program with Turbo C++

Thanks!!

Note:- This post has been edited by ***tanny*** *at* ***Wed, 09/16/2009 - 16:45****.*

Submitted by neeraj146b on Tue, 09/08/2009 - 10:43. #9

Member since:
8 September 2009
Last activity:
13 weeks 2 days

plzz.... suggest some best books on advanced c/c++

and if any1 know how 1 should start with c/c++ library development, then plz.. share it with us...

with regards

neeraj

Submitted by neeraj146b on Tue, 09/08/2009 - 08:19. #10

Member since:
8 September 2009
Last activity:
13 weeks 2 days

u r a great programmer... and yes ... as dragon has said... plz extend this programming to advanced level...

with regards

neeraj

1 2 next › last »

popular robotics tutorials

Building an RF remote control
Total views: 28,251

Line Follower
Total views: 23,581

Input-Output instructions for 8051 in Assembly language
Total views: 15,501

Motor controller: H-Bridge
Total views: 12,744

Introduction to 8051 Assembly Language Programming
Total views: 9,004

popular programming tutorials

Mouse Programming with C
Total views: 10,100

Tic Tac Toe - Game programming
Total views: 5,037

AJAX Basic Tutorial
Total views: 4,104

AJAX Search Tutorial
Total views: 4,039

Functions in C++ - Advanced features
Total views: 3,580

who's online

There are currently *1 user* and *3 guests* online.

online users

shashwat