ใบงานที่ 2 Keyboard and Mouse

# **วัตถุประสงค์**

1. ตรวจจับการทำงานของ Keyboard และ Mouse

2. ใช้เครื่องมือต่างๆ ในการสร้าง Application

# **เนื้อหาเบื้องต้น**

https://github.com/Desktop-Programming-Lab-2559/LAB-02/blob/master/LabIntro.md

# **ลำดับการทดลอง**

1. เรียกโปรแกรม Microsoft Visual Studio

2. สร้าง Project ใหม่ โดยเลือกเมนู File >> New >> Project… (Ctrl+Shift+N)

· ช่อง Templates: ให้เลือก Visual C++ และเลือกชนิด project เป็น Empty Project

· ช่อง Name: ให้ใส่ชื่อของ Project เป็น EasyWin32

· ช่อง Location: ให้เลือกตำแหน่งที่จะสร้าง Project (D:\Student\_Code\LAB\_01\_03)

· ส่วนที่เหลือ ให้คงไว้ตามที่ปรากฏ กด OK

3. เพิ่ม source code ให้กับ project โดยการเลือกเมนู PROJECT >> Add New Item…

· ตั้งชื่อไฟล์เป็น main.cpp

· กด Add เพื่อเพิ่มไฟล์

4. พิมพ์โปรแกรมดังต่อไปนี้ลงในไฟล์ main.cpp

|  |
| --- |
| #include <windows.h>  #include <tchar.h>  // Global Variables:  HINSTANCE hInst; // current instance  TCHAR szTitle[]="Easywin32"; // The title bar text  TCHAR szWindowClass[]="WinApp"; // the class name  // Foward declarations of functions included in this code module:  ATOM MyRegisterClass(HINSTANCE hInstance);  BOOL InitInstance(HINSTANCE, int);  LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);  int APIENTRY WinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPSTR lpCmdLine,  int nCmdShow)  {  // TODO: Place code here.  MSG msg;  // Register Class  MyRegisterClass(hInstance);  // Perform application initialization:  if (!InitInstance (hInstance, nCmdShow))  {  return FALSE;  }  // Main message loop:  while (GetMessage(&msg, NULL, 0, 0))  {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return msg.wParam;  }  //  // FUNCTION: MyRegisterClass()  //  // PURPOSE: Registers the window class.  //  // COMMENTS:  //  // This function and its usage is only necessary if you want this code  // to be compatible with Win32 systems prior to the 'RegisterClassEx'  // function that was added to Windows 95.  //  ATOM MyRegisterClass(HINSTANCE hInstance)  {  WNDCLASSEX wcex;  wcex.cbSize = sizeof(WNDCLASSEX);  wcex.style = CS\_HREDRAW | CS\_VREDRAW;  wcex.lpfnWndProc = (WNDPROC)WndProc;  wcex.cbClsExtra = 0;  wcex.cbWndExtra = 0;  wcex.hInstance = hInstance;  wcex.hIcon = LoadIcon (NULL, IDI\_APPLICATION);  wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);  wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);  wcex.lpszMenuName = NULL;  wcex.lpszClassName = szWindowClass;  wcex.hIconSm = LoadIcon (NULL, IDI\_APPLICATION);  return RegisterClassEx(&wcex);  }  //  // FUNCTION: InitInstance(HANDLE, int)  //  // PURPOSE: Saves instance handle and creates main window  //  // COMMENTS:  //  // In this function, we save the instance handle in a global variable and  // create and display the main program window.  //  BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  {  HWND hWnd;  hInst = hInstance; // Store instance handle in our global variable  hWnd = CreateWindow(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,  CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);  if (!hWnd)  {  return FALSE;  }  ShowWindow(hWnd, nCmdShow);  UpdateWindow(hWnd);  return TRUE;  }  //  // FUNCTION: WndProc(HWND, unsigned, WORD, LONG)  //  // PURPOSE: Processes messages for the main window.  //  // WM\_COMMAND - process the application menu  // WM\_PAINT - Paint the main window  // WM\_DESTROY - post a quit message and return  //  //  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  PAINTSTRUCT ps;  HDC hdc;  RECT rt;  char szHello[]="Hello, C-Free!";    switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...    GetClientRect(hWnd, &rt);  DrawText(hdc, szHello, strlen(szHello), &rt, DT\_CENTER);  EndPaint(hWnd, &ps);  break;  case WM\_CLOSE:  DestroyWindow(hWnd);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  }  return 0;  } |

5. ทำการทดสอบโปรแกรม/บันทึกผล

|  |
| --- |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| มีหน้าต่างขึ้นมา โดยมีการแสดงผลเป็น Hello, C-Free! อยู่ตรงกลาง |

6. เพิ่ม Function Marker() ในโปรแกรม

|  |
| --- |
| void Marker(LONG x, LONG y, HWND hwnd)  {  HDC hdc;  HPEN hPen;  hdc = GetDC(hwnd);  hPen = CreatePen(PS\_DOT,1,RGB(255,0,0));    SelectObject(hdc, hPen);  MoveToEx(hdc, (int) x - 10, (int) y, (LPPOINT) NULL);  LineTo(hdc, (int) x + 10, (int) y);  MoveToEx(hdc, (int) x, (int) y - 10, (LPPOINT) NULL);  LineTo(hdc, (int) x, (int) y + 10);  DeleteObject(hPen);  ReleaseDC(hwnd, hdc);  } |

7. แก้ไขโปรแกรม

|  |  |
| --- | --- |
| case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...  GetClientRect(hWnd, &rt);  DrawText(hdc, szHello, strlen(szHello), &rt, DT\_CENTER);  EndPaint(hWnd, &ps);  break; | case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...  GetClientRect(hWnd, &rt);  DrawText(hdc, szHello, strlen(szHello), &rt, DT\_CENTER);  Marker(50,50,hWnd)  EndPaint(hWnd, &ps);  break; |

8. ทำการทดสอบโปรแกรม/บันทึกผล

|  |
| --- |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| มีรูปที่มีลักษณะเครื่องหมายบวกสีแดงขึ้นมา |

9. เพิ่มโปรแกรมตามตาราง

|  |
| --- |
| case WM\_LBUTTONDOWN:  fDraw = TRUE;  ptPrevious.x = LOWORD(lParam);  ptPrevious.y = HIWORD(lParam);  return 0L;    case WM\_LBUTTONUP:  if (fDraw)  {  hdc = GetDC(hWnd);  MoveToEx(hdc, ptPrevious.x, ptPrevious.y, NULL);  LineTo(hdc, LOWORD(lParam), HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  fDraw = FALSE;  return 0L;    case WM\_MOUSEMOVE:  if (fDraw)  {  hdc = GetDC(hWnd);  MoveToEx(hdc, ptPrevious.x, ptPrevious.y, NULL);  LineTo(hdc, ptPrevious.x = LOWORD(lParam),  ptPrevious.y = HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  return 0L; |

10. เพิ่มตัวแปร BOOL fDraw = FALSE; POINT ptPrevious; ไว้ที่ Global Area

11. ทำการทดสอบโปรแกรม/บันทึกผล

|  |
| --- |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| สามารถวาด,เขียนรูปขึ้นมาได้ |

12. เพิ่มตัวแปร COLORREF Color[]={RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,0,0)}; ไว้ที่ Global Area

13. แก้ไข Function Marker() ตามตาราง

|  |
| --- |
| void Marker(LONG x, LONG y,int Index, HWND hwnd)  {  HDC hdc;  HPEN hPen;  hdc = GetDC(hwnd);  hPen = CreatePen(PS\_DOT,1,Color[Index]);    SelectObject(hdc, hPen);  MoveToEx(hdc, (int) x - 10, (int) y, (LPPOINT) NULL);  LineTo(hdc, (int) x + 10, (int) y);  MoveToEx(hdc, (int) x, (int) y - 10, (LPPOINT) NULL);  LineTo(hdc, (int) x, (int) y + 10);  DeleteObject(hPen);  ReleaseDC(hwnd, hdc);  } |

14. แก้ไขโปรแกรมตามตาราง

|  |
| --- |
| case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...    GetClientRect(hWnd, &rt);  DrawText(hdc, szHello, strlen(szHello), &rt, DT\_CENTER);  Marker(50,50,Index,hWnd);  EndPaint(hWnd, &ps);  break; |

15. ทำการทดสอบโปรแกรม/บันทึกผล

|  |
| --- |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| ยังไม่มีการเปลี่ยนแปลงเพิ่มเติม แต่มีการเพิ่มฟังก์ชันสีเข้ามา |

16. ทำการเพิ่มโปรแกรมตามตาราง

|  |
| --- |
| case WM\_KEYDOWN:  switch (wParam)  {  case VK\_LEFT:    // Process the LEFT ARROW key.    if(Index<4)  {  Index++;  }else  {  Index=0;  }  InvalidateRect(hWnd, NULL, TRUE);  default:  break;  }  break; |

17. ทำการทดสอบโปรแกรม/บันทึกผล

|  |
| --- |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| มีการเปลี่ยนแปลงสีไปตามที่กำหนดไว้ เมื่อกดลูกศร |

# สรุปผลการทดลอง

จากการทดลองคือ สามารถเพิ่มข้อความเข้ามาในหน้าต่างได้ โดยใช้ฟังก์ชัน DrawText, สามารถเพิ่มฟังก์ชัน Marker() ในโปรแกรมได้, สามารถใช้ฟังก์ชัน WM\_LBUTTONDOWN คือ การคลิ๊กเมาส์ลง, ฟังก์ชัน WM\_LBUTTONUP คือ เวลาปล่อยเมาส์, ฟังก์ชัน WM\_MOUSEMOVE คือ การขยับเมาส์ เคลื่อนที่เมาส์, ฟังก์ชัน WM\_KEYDOWN คือ การกดแป้นพิมพ์ โดยมีเงื่อนไขอยู่ในฟังก์ชัน ทั้งหมดนี้จะสามารถวาดภาพได้

# คำถามท้ายการทดลอง

1. เขียนโปรแกรมวาดรูปที่สามารถเปลี่ยนสีปากกาได้

|  |
| --- |
| **โปรแกรม** |
| #include <windows.h>  #include <tchar.h>  //Global Variables:  HINSTANCE hInst; //current instance  TCHAR szTitle[]="Easywin32"; //The title bar text  TCHAR szWindowClass[]="WinApp"; //the class name  BOOL fDraw = FALSE; POINT ptPrevious;  COLORREF Color[]={RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,0,0)};  //Foward declarations of functions included in this code module:  ATOM MyRegisterClass(HINSTANCE hInstance);  BOOL InitInstance(HINSTANCE, int);  LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);  int Index;  void Marker(LONG x, LONG y, int Index, HWND hWnd);  int APIENTRY WinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPSTR lpCmdLine,  int nCmdShow)  {  // TODO: Place code here.  MSG msg;    //Register Class  MyRegisterClass(hInstance);    //Perform application initialization:  if(!InitInstance (hInstance, nCmdShow))  {  return FALSE;  }    //Main message loop:  while (GetMessage(&msg,NULL,0,0))  {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return msg.wParam;  }  //  //FUNCTION: MyRegisterClass()  //  //PURPOSE: Registers the window class.  //  //COMMENTS:  //  // This function and its usage is only necessary if you want this code  // to be compatible with Win32 systems prior to the 'RegisterClassEx'  // function that was added to Windows 95.  //  ATOM MyRegisterClass(HINSTANCE hInstance)  {  WNDCLASSEX wcex;  wcex.cbSize = sizeof(WNDCLASSEX);  wcex.style = CS\_HREDRAW | CS\_VREDRAW;  wcex.lpfnWndProc = (WNDPROC)WndProc;  wcex.cbClsExtra = 0;  wcex.cbWndExtra = 0;  wcex.hInstance = hInstance;  wcex.hIcon = LoadIcon(NULL,IDI\_APPLICATION);  wcex.hCursor = LoadCursor(NULL,IDC\_ARROW);  wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);  wcex.lpszMenuName = NULL;  wcex.lpszClassName = szWindowClass;  wcex.hIconSm = LoadIcon(NULL,IDI\_APPLICATION);    return RegisterClassEx(&wcex);  }  //  //FUNCTION" InitInstance(HANDLE, int)  //  //PURPOSE: Saces instance handle and creates main window  //  //COMMENTS:  //  // In this function, we save the instance handle in a global variable and  // create and display the main program window.  //  BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  {  HWND hWnd;    hInst = hInstance; // Store instance handle in our global variable    hWnd = CreateWindow(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,  CW\_USEDEFAULT,0,CW\_USEDEFAULT,0,NULL,NULL,hInstance,NULL);  if(!hWnd)  {  return FALSE;  }  ShowWindow(hWnd,nCmdShow);  UpdateWindow(hWnd);    return TRUE;  }  //  //FUNCTION: WndProc(HWND,unsigned,WORD,LONG)  //  //PURPOSE: Processes messages for the main window.  //  // WM\_COMMAND - process the application menu  // WM\_PAINT - Paint the main window  // WM\_DESTROY - post a quit message and return  //  //  LRESULT CALLBACK WndProc(HWND hWnd,UINT message,WPARAM wParam,LPARAM  lParam)  {  PAINTSTRUCT ps;  HDC hdc;  RECT rt;  char szHello[]="Hello, C-Free!";    switch (message)  {  case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...    GetClientRect(hWnd, &rt);  DrawText(hdc,szHello,strlen(szHello), &rt, DT\_CENTER);  Marker(50,50,Index,hWnd);  EndPaint(hWnd, &ps);  break;  case WM\_LBUTTONDOWN:  fDraw = TRUE;  ptPrevious.x = LOWORD(lParam);  ptPrevious.y = HIWORD(lParam);  return 0L;  case WM\_LBUTTONUP:  if(fDraw)  {  hdc = GetDC(hWnd);  MoveToEx(hdc,ptPrevious.x, ptPrevious.y,NULL);  LineTo(hdc,LOWORD(lParam),HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  fDraw = FALSE;  return 0L;  case WM\_MOUSEMOVE:  if(fDraw)  {  HDC hdc;  hdc = GetDC(hWnd);  HPEN hPen;  hPen = CreatePen(PS\_DOT,5,Color[Index]);    SelectObject(hdc, hPen);  MoveToEx(hdc,ptPrevious.x, ptPrevious.y, NULL);  LineTo(hdc,ptPrevious.x = LOWORD(lParam),  ptPrevious.y = HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  return 0L;  case WM\_KEYDOWN:  switch (wParam)  {  case VK\_LEFT:  //Process the LEFT ARROW key.  if(Index<4)  {  Index++;  }  else  {  Index=0;  }  //InvalidateRect(hWnd,NULL,TRUE);  default:  break;  }  break;  case WM\_CLOSE:  DestroyWindow(hWnd);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd,message,wParam,lParam);  }  return 0;  }  void Marker(LONG x, LONG y, int Index, HWND hWnd)  {  HDC hdc;  HPEN hPen;  hdc = GetDC(hWnd);  hPen = CreatePen(PS\_DOT,5,Color[Index]);    SelectObject(hdc, hPen);  MoveToEx(hdc, 200 - 10, 50, NULL);  LineTo(hdc, 200 + 10, 50);  MoveToEx(hdc, 200, 50 - 10, NULL);  LineTo(hdc, 200, 50 + 10);  DeleteObject(hPen);  ReleaseDC(hWnd, hdc);  } |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| เมื่อวาดรูปและกดลูกศรไปเรื่อยๆ จะมีการเปลี่ยนแปลงสีไปตามที่กำหนดไว้ |

1. เขียนโปรแกรมพิมพ์ข้อมูลจากคีย์บอร์ด แล้วแสดงผลที่หน้าจอ

|  |
| --- |
| **โปรแกรม** |
| #include <windows.h>  #include <tchar.h>  // Global Variables:  HINSTANCE hInst; // current instance  TCHAR szTitle[] = "Easywin32"; // The title bar text  TCHAR szWindowClass[] = "WinApp"; // the class name  void Marker(LONG x, LONG y, int Index, HWND hwnd);  int Index;  int HelloIndex =1 ;  char szchar[50] = { 0 };  HPEN hPen;  BOOL fDraw = FALSE; POINT ptPrevious;  COLORREF Color[] = { RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,0,0) };  // Foward declarations of functions included in this code module:  ATOM MyRegisterClass(HINSTANCE hInstance);  BOOL InitInstance(HINSTANCE, int);  LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);  int APIENTRY WinMain(HINSTANCE hInstance,  HINSTANCE hPrevInstance,  LPSTR lpCmdLine,  int nCmdShow)  {  // TODO: Place code here.  MSG msg;  // Register Class  MyRegisterClass(hInstance);  // Perform application initialization:  if (!InitInstance(hInstance, nCmdShow))  {  return FALSE;  }  // Main message loop:  while (GetMessage(&msg, NULL, 0, 0))  {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return msg.wParam;  }  //  // FUNCTION: MyRegisterClass()  //  // PURPOSE: Registers the window class.  //  // COMMENTS:  //  // This function and its usage is only necessary if you want this code  // to be compatible with Win32 systems prior to the 'RegisterClassEx'  // function that was added to Windows 95.  //  ATOM MyRegisterClass(HINSTANCE hInstance)  {  WNDCLASSEX wcex;  wcex.cbSize = sizeof(WNDCLASSEX);  wcex.style = CS\_HREDRAW | CS\_VREDRAW;  wcex.lpfnWndProc = (WNDPROC)WndProc;  wcex.cbClsExtra = 0;  wcex.cbWndExtra = 0;  wcex.hInstance = hInstance;  wcex.hIcon = LoadIcon(NULL, IDI\_APPLICATION);  wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);  wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);  wcex.lpszMenuName = NULL;  wcex.lpszClassName = szWindowClass;  wcex.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);  return RegisterClassEx(&wcex);  }  //  // FUNCTION: InitInstance(HANDLE, int)  //  // PURPOSE: Saves instance handle and creates main window  //  // COMMENTS:  //  // In this function, we save the instance handle in a global variable and  // create and display the main program window.  //  BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  {  HWND hWnd;  hInst = hInstance; // Store instance handle in our global variable  hWnd = CreateWindow(szWindowClass,szTitle,WS\_OVERLAPPEDWINDOW,  CW\_USEDEFAULT,0,CW\_USEDEFAULT,0, NULL, NULL,hInstance, NULL);  if (!hWnd)  {  return FALSE;  }  ShowWindow(hWnd, nCmdShow);  UpdateWindow(hWnd);  return TRUE;  }  //  // FUNCTION: WndProc(HWND, unsigned, WORD, LONG)  //  // PURPOSE: Processes messages for the main window.  //  // WM\_COMMAND - process the application menu  // WM\_PAINT - Paint the main window  // WM\_DESTROY - post a quit message and return  //  //  LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)  {  PAINTSTRUCT ps;  HDC hdc;  RECT rt;  char szHello[50] = "Hello, C-Free!";  switch (message)  {  case WM\_LBUTTONDOWN:  fDraw = TRUE;  ptPrevious.x = LOWORD(lParam);  ptPrevious.y = HIWORD(lParam);  return 0L;  case WM\_LBUTTONUP:  if (fDraw)  {  hdc = GetDC(hWnd);  MoveToEx(hdc, ptPrevious.x, ptPrevious.y, NULL);  LineTo(hdc, LOWORD(lParam), HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  fDraw = FALSE;  return 0L;  case WM\_MOUSEMOVE:  if (fDraw)  {  hdc = GetDC(hWnd);  MoveToEx(hdc, ptPrevious.x, ptPrevious.y, NULL);  LineTo(hdc, ptPrevious.x = LOWORD(lParam),  ptPrevious.y = HIWORD(lParam));  ReleaseDC(hWnd, hdc);  }  return 0L;  case WM\_PAINT:  hdc = BeginPaint(hWnd, &ps);  // TODO: Add any drawing code here...  GetClientRect(hWnd, &rt);  DrawText(hdc, szHello, strlen(szHello), &rt, DT\_CENTER);  Marker(50, 50, Index, hWnd);  EndPaint(hWnd, &ps);  break;  case WM\_CLOSE:  DestroyWindow(hWnd);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  default:  return DefWindowProc(hWnd, message, wParam, lParam);  case WM\_CHAR:  hdc = GetDC(hWnd);  szchar[HelloIndex] = (TCHAR)wParam;;  HelloIndex++;  TextOut(hdc, 30, 100, szchar, 10);  ReleaseDC(hWnd, hdc);  break;  case WM\_KEYDOWN:  switch (wParam)  {  case VK\_LEFT:  // Process the LEFT ARROW key.  if (Index<4)  {  Index++;  }  else  {  Index = 0;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  default :  break;  }  break;  }  return 0;  }  void Marker(LONG x, LONG y, int Index, HWND hwnd)  {  HDC hdc;  hdc = GetDC(hwnd);  hPen = CreatePen(PS\_DOT, 5, Color[Index]);    SelectObject(hdc, hPen);  MoveToEx(hdc, (int)x - 10, (int)y, (LPPOINT)NULL);  LineTo(hdc, (int)x + 10, (int)y);  MoveToEx(hdc, (int)x, (int)y - 10, (LPPOINT)NULL);  LineTo(hdc, (int)x, (int)y + 10);  DeleteObject(hPen);  ReleaseDC(hwnd, hdc);  } |
| **ทดสอบโปรแกรม** |
|  |
| **บันทึกผล** |
| สร้างแปรตัวเก็บค่า แล้วสามารถพิมพ์ข้อความแล้วแสดงบนหน้าจอได้ |