

## ภาคผนวก ก.

สำหรับรายละเอียดของตัวเครื่องหยอดเหรียญนั้นมีดังนี้

### 1. SPECIFICATION

Voltage :DC12 V $\pm$ 20%

Coin diameter :18mm-31mm

Coin thikness :1.2mm-3.0mm

Temperater :-15 C- +75 C

### 2. INSTALLATION

1. นำเหรียญพลาสติกที่เป็นตัวอย่างออกจากเครื่อง
2. นำเหรียญต้นแบบที่จะใช้ในการหยอดเหรียญใส่ลงไปเครื่องแทน
3. ใช้ไขควงปรับให้พอดีกับเหรียญ
4. เลือก mode ว่าจะใช้ Normal open หรือ Normal close เลือกระดับความเร็วในการรับค่า ว่าเป็น 100 ms , 50ms ,30 ms จาก timer switch เพื่อให้ตรงกับ soft ware
5. ปรับค่า Clock +,-
6. ความหมายของสายสีต่างๆ
  - สีเทา 1 : counter (+/-)
  - สีแดง :DC+12 V
  - สีขาว : coin singnal
  - สีดำ : GND
  - สีเทา 2 :counter (-/+)

## ภาคผนวก ข.

### Source Code

```
// RsCgDlg.cpp : implementation file
#include "stdafx.h"
#include "RsCgDlg.h"
#include "P_MAINDlg.h"
#include "PASS.h"
#include "winpool.h"
#include "EXIT.h"
#include "PRINTERED.h"
#include "INBANK.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CRsCgDlg dialog
///=====///
///=====///
//=====

CString password_all, printname, name, lname;;
//LPTSTR rrr;

DWORD   dwNeeded, dwReturned, i;
JOB_INFO_1 *pJobInfo;
int now_printed, uuu, your_save;
char Show_Jane[100];
int slot1, slot2, BpP;
int STATUS_OUI_PROGRAM, clock1, clock2;
int status_program; //1=standby 2=login 3=inputcoin 4=false
int Count_Page_in_my_program1, Count_Page_in_my_program2;
```

```

//=====
///=====
///=====
CRsCgDlg::CRsCgDlg(CWnd* pParent /*=NULL*/)
: CDialog(CRsCgDlg::IDD, pParent)
{
    m_strConnection = _T("Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data
    Source=C:\\Program Files\\Prnt_cat\\P_MAIN\\res\\OUI.mdb");
    m_strCmdText = _T("OUI");
    m_pRs = NULL;
    //{AFX_DATA_INIT(CRsCgDlg)
    m_strDlgST_ID = _T("");
    m_strDlgFname = _T("");
    m_strDlgLname = _T("");
    m_strDlgPASSWD = _T("");
    m_IDlgMONEY = 0;
    m_strDlgLASTPRINTDOC = _T("");
    m_AddingFlag=FALSE;
    //}}AFX_DATA_INIT
    }
    CRsCgDlg::~CRsCgDlg()
    {
        m_pRs = NULL;
    }
    //////////////////////////////////////
    BOOL CRsCgDlg::OnInitDialog()
    {
        HRESULT hr = NOERROR;
        IADORecordBinding *piAdoRecordBinding = NULL;
        BOOL jane;
        LPTSTR mrr;

```

```

CDialog::OnInitDialog();

SetTimer(JANE_SLOT1,10000,NULL);

SetTimer(JANE_SLOT2,5000,NULL);

SetTimer(JANE_CAT_PRINT,1000,NULL);

Count_Page_in_my_program1=0;

Count_Page_in_my_program2=0;

//===== for show =====

name=AfxGetApp()->GetProfileString("Data Root","name","NoData");

lname=AfxGetApp()->GetProfileString("Data Root","lname","NoData");

printname=AfxGetApp()->GetProfileString("Data Root","printname","NoData");

slot1=AfxGetApp()->GetProfileInt("Data Root","slot1",0);

slot2=AfxGetApp()->GetProfileInt("Data Root","slot2",0);

BpP=AfxGetApp()->GetProfileInt("Data Root","BpP",0);

//===== for system =====

password_all=AfxGetApp()->GetProfileString("Data Root","password","NoData");

clock1=AfxGetApp()->GetProfileInt("Data Root","clock1",0);

clock2=AfxGetApp()->GetProfileInt("Data Root","clock2",0);

//===== Show =====

SetDlgItemText(IDC_EDIT_NAME,name);

SetDlgItemText(IDC_EDIT_LLNAME,lname);

SetDlgItemText(IDC_EDIT_PRINT_N,printname);

sprintf>Show_Jane,"%d",BpP);SetDlgItemText(IDC_EDIT_BPP>Show_Jane);

sprintf>Show_Jane,"%d",slot1);SetDlgItemText(IDC_EDIT_SLOT1>Show_Jane);

sprintf>Show_Jane,"%d",slot2);SetDlgItemText(IDC_EDIT_SLOT2>Show_Jane);

try
{

m_pRs.CreateInstance(__uuidof(Recordset));

m_pRs->CursorLocation = adUseClient;

m_pRs->Open((LPCTSTR)m_strCmdText, (LPCTSTR)m_strConnection, adOpenStatic,

adLockReadOnly, adCmdTableDirect);

```

```

if (FAILED(hr = m_pRs->QueryInterface(__uuidof(IADORecordBinding), (LPVOID
*)&piAdoRecordBinding)))
_com_issue_error(hr);
if (FAILED(hr = piAdoRecordBinding->BindToRecordset(this)))
_com_issue_error(hr);
RefreshBoundData();

//----- HIDE -----

SetWindowPos(NULL,0,0,1,1,SWP_SHOWWINDOW); //hide me
//----- HIDE -----

_tcscpy(rrr,printname);
jane=SetJob_PRINT(3,rrr); //delete
}
catch (_com_error &e)
{
        GenerateError(e.Error(), e.Description());
}
if (piAdoRecordBinding)
piAdoRecordBinding->Release();
//-----

return TRUE;
}

////////////////////////////////////

void CRsCgDlg::RefreshBoundData()
{
if (adFldOK == IST_IDStatus) m_strDlgST_ID = m_szST_ID;
else m_strDlgST_ID = _T("");
if (adFldOK == lFnameStatus) m_strDlgFname = m_szFname;
else m_strDlgFname = _T("");
if (adFldOK == lLnameStatus) m_strDlgLname = m_szLname;
else m_strDlgLname = _T("");
if (adFldOK == lPASSWDStatus)m_strDlgPASSWD = m_szPASSWD;
else m_strDlgPASSWD = _T("");
if (adFldOK == lMONEYStatus)m_IDlgMONEY = m_lMONEY;
else m_IDlgMONEY = 0;
}

```

```

if (adFldOK == ILASTPRINTDOCStatus)m_strDlgLASTPRINTDOC = m_szLASTPRINTDOC;
else
    m_strDlgLASTPRINTDOC = _T("");
UpdateData(FALSE);
}

void CRsCgDlg::GenerateError(HRESULT hr, PWSTR pwszDescription)
{
    CString strError;
    strError.Format("Run-time error '%d (%x)'", hr, hr);
    strError += "\n\n";
    strError += pwszDescription;
    AfxMessageBox(strError);
}

void CRsCgDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CRsCgDlg)
    DDX_Text(pDX, IDC_EDIT_ST_ID, m_strDlgST_ID);
    DDX_Text(pDX, IDC_EDIT_FNAME, m_strDlgFname);
    DDX_Text(pDX, IDC_EDIT_LNAME, m_strDlgLname);
    DDX_Text(pDX, IDC_EDIT_PASSWD, m_strDlgPASSWD);
    DDX_Text(pDX, IDC_EDIT_MONEY, m_IDlgMONEY);
    DDX_Text(pDX, IDC_EDIT_LASTPRINTDOC, m_strDlgLASTPRINTDOC);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CRsCgDlg, CDialog)
//{{AFX_MSG_MAP(CRsCgDlg)
    ON_BN_CLICKED(IDC_SEARCH, OnSearch)
    ON_BN_CLICKED(IDC_RESUME, OnResume)
    ON_BN_CLICKED(IDC_CLOSE_ALL, OnClose_all)
    ON_BN_CLICKED(IDC_CANCEL_PRINT, OnCancel_print)
    ON_BN_CLICKED(IDC_1BATH, On1BATH)
    ON_BN_CLICKED(IDC_5BATH, On5BATH)

```

```

ON_WM_SPOOLERSTATUS()

ON_WM_TIMER()

//}}AFX_MSG_MAP

END_MESSAGE_MAP()

////////////////////////////////////

// CRsCgDlg message handlers

//-----

void CRsCgDlg::OnSearch()
{
    BOOL jane;
    LPTSTR rrr;
    _tcscpy(rrr,printname);
    jane=SetJob_PRINT(1,rrr); //pause
}

//-----

void CRsCgDlg::OnResume()
{
    BOOL jane;
    LPTSTR rrr;
    PRINTED ccc;
    INBANK ddd;
    _tcscpy(rrr,printname);
    if(uuu<=0)
    {
        //if(uuu<0)

        //----- HIDE -----
        SetWindowPos(NULL,0,0,1,1,SWP_SHOWWINDOW); //hide me
        //----- HIDE -----

        status_program=6;//1=standby 2=login 3=inputcoin 4=false 5=printed 6=printing

        //save_data_to(); //Save DATABASE

        jane=SetJob_PRINT(2,rrr); //resume
    }
}

```

```

//if(ccc.DoModal()==IDOK)
//{
//now_printed=Check_printed(rrr);
//jane=SetJob_PRINT(3,rrr); //delete all job
//ddd.m_edt_inbank=your_save;
//if(ddd.DoModal()==IDOK)
//{
//save_data_to();
//
//                MessageBox("เก็บข้อมูลจำนวนเงินเรียบร้อยแล้ว");
//}
//}
}
else
{
status_program=3;
//1=standby 2=login 3=inputcoin 4=false 5=printed 6=printing
MessageBox("กรุณายกเหรียญให้เท่ากับหรือมากกว่าที่เครื่องต้องการด้วยครับ");
}
}

//-----

//-----

void CRsCgDlg::OnClose_all()
{
EXIT bbb;
BOOL jane;
LPTSTR rrr;
if(bbb.DoModal()==IDOK)
{
if(bbb.m_edt_passexit==password_all)
{
jane=SetJob_PRINT(3,rrr); //delete all job
CDialog::OnOK();
}
}
}

```

```

else MessageBox("รหัสผ่าน ไม่ถูกต้อง");
}

} //-----

//-----

void CRsCgDlg::OnCancel_print()
{
    BOOL jane;
    LPTSTR mrr;
    //----- HIDE -----
    SetWindowPos(NULL,0,0,1,1,SWP_SHOWWINDOW); //hide me
    //----- HIDE -----
    jane=SetJob_PRINT(3,mrr); //delete all job
} //-----

//-----

void CRsCgDlg::On1BATH()
{
    LPTSTR mrr;
    if(uuu>0)
    {
        uuu=uuu-1;
        if(uuu<0) {your_save=uuu*-1;m_IDlgMONEY=your_save;}
        sprintf>Show_Jane,"%d",uuu);
        SetDlgItemText(IDC_EDIT_PAY,Show_Jane);
    }
} //-----

//-----

void CRsCgDlg::On5BATH()
{
    LPTSTR mrr;
    if(uuu>0)
    {

```

```

uuu=uuu-5;
if(uuu<0) {your_save=uuu*-1; m_IDlgMONEY=your_save;}
sprintf>Show_Jane,"%d",uuu);
SetDlgItemText(IDC_EDIT_PAY,Show_Jane);
}
}

//----- List Data in Job -----

BOOL CRsCgDlg::ListJobsForPrinter(LPTSTR szPrinterName)
{
HANDLE hPrinter;
if( ! OpenPrinter( szPrinterName, &hPrinter, NULL ) ) return FALSE;
if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &dwNeeded, &dwReturned ) )
{
if( GetLastError() != ERROR_INSUFFICIENT_BUFFER )
{
ClosePrinter( hPrinter ); return FALSE;
}
}
if( (pJobInfo = (JOB_INFO_1 *)malloc( dwNeeded )) == NULL )
{
ClosePrinter( hPrinter ); return FALSE;
}
if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, (LPBYTE)pJobInfo, dwNeeded, &dwNeeded,
&dwReturned ) )
{
ClosePrinter( hPrinter );free( pJobInfo );return FALSE;
}
ClosePrinter( hPrinter );
}

//-----

SetDlgItemText(IDC_EDIT_NAMEDOC,pJobInfo[dwReturned-1].pDocument);
sprintf>Show_Jane,"%d",pJobInfo[dwReturned-1].TotalPages);
SetDlgItemText(IDC_EDIT_PAGE,Show_Jane);
uuu=pJobInfo[dwReturned-1].TotalPages*BpP-m_IDlgMONEY;
sprintf>Show_Jane,"%d",uuu+m_IDlgMONEY);SetDlgItemText(IDC_EDIT_PAYNOW,Show_Jane)
;
if(uuu>0)

```

```

{
    sprintf(Show_Jane,"%d",uuu);SetDlgItemText(IDC_EDIT_PAY,Show_Jane);
}

else SetDlgItemText(IDC_EDIT_PAY,"0");
//-----
/*printf( "%d jobs\n", dwReturned );
for(i=0;i<dwReturned;i++)
{
    m_lst_in_spool.AddString(pJobInfo[i].pDocument);
    sprintf(Show_Jane,"%d",pJobInfo[i].Status);
    sprintf(Show_Jane,"%d",pJobInfo[i].PagesPrinted);
}*/
free( pJobInfo );
return TRUE;
}//-----

//----- Set Status -----
BOOL CRsCgDlg::SetJob_PRINT(int a, LPTSTR szPrinterName)
{
    HANDLE    hPrinter;
    JOB_INFO_1 *pJobInfo;
    if( ! OpenPrinter( szPrinterName, &hPrinter, NULL ) ) return FALSE;
    if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &dwNeeded,&dwReturned ) )
    {
        if( GetLastError() != ERROR_INSUFFICIENT_BUFFER )
        { ClosePrinter( hPrinter ); return FALSE; }
    }
    if( (pJobInfo = (JOB_INFO_1 *)malloc( dwNeeded )) == NULL )
    { ClosePrinter( hPrinter ); return FALSE; }
    if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, (LPBYTE)pJobInfo,dwNeeded, &dwNeeded,
    &dwReturned ) )
    { ClosePrinter( hPrinter ); free( pJobInfo ); return FALSE; }

```

```

if(a==1) //pause
{
    if(! SetPrinter(hPrinter,0,NULL,PRINTER_CONTROL_PAUSE))
    {
        ClosePrinter( hPrinter ); return FALSE;
    }
}
else if(a==2) //resume
{
    if(! SetPrinter(hPrinter,0,NULL,PRINTER_CONTROL_RESUME))
    {
        ClosePrinter( hPrinter ); return FALSE;
    }
}
else if(a==3) //delete all
{
    if(! SetPrinter(hPrinter,0,NULL,PRINTER_CONTROL_PURGE))
    {
        ClosePrinter( hPrinter ); return FALSE;
    }
}
ClosePrinter( hPrinter );
return TRUE;
}

//-----

//-----

void CRsCgDlg::OnTimer(UINT nIDEvent)
{
    LPTSTR mrr;
    // TODO: Add your message handler code here and/or call default
    if(nIDEvent==JANE_SLOT1)// timer for slot1
    {
        //MessageBox("Jane is very handsome");
    }
    if(nIDEvent==JANE_SLOT2)//timer for slot2
    {
    }
}

```

```
if(nIDEvent==JANE_CAT_PRINT) // timer for jane
```

```
{
```

```
    PRINTED rutta;
```

```
    LPTSTR rrr;
```

```
    //rutta.m_edt_name=m_strDlgFname;
```

```
    _tcsncpy(rrr,printname);
```

```
    now_printed=Check_printed(rrr);
```

```
    if(now_printed>=0)
```

```
    {
```

```
        rutta.m_edt_printed=now_printed;
```

```
    }
```

```
    else rutta.m_edt_printed=now_printed;
```

```
}
```

```
CDialog::OnTimer(nIDEvent);
```

```
}//-----
```

```
//-----
```

```
void CRsCgDlg::OnSpoolerStatus(UINT nStatus, UINT nJobs)
```

```
{
```

```
    char Show_Jane[100];
```

```
    //CString name,lname;
```

```
    BOOL jane;
```

```
    LPTSTR rrr;
```

```
    CDialog::OnSpoolerStatus(nStatus, nJobs);
```

```
    //////////////////////////////////
```

```
    your_save=0;
```

```
    //===== for show =====
```

```
    name=AfxGetApp()->GetProfileString("Data Root","name","NoData");
```

```
    lname=AfxGetApp()->GetProfileString("Data Root","lname","NoData");
```

```
    printname=AfxGetApp()->GetProfileString("Data Root","printname","NoData");
```

```
    slot1=AfxGetApp()->GetProfileInt("Data Root","slot1",0);
```

[illegible]

```

long recCount=0;
if(temp.DoModal()==IDOK)
{
try
{
data="ST_ID= '"+temp.m_edit_id+"' and PASSWD= '"+temp.m_edit_pass+"'";
m_pRs->Filter=data.GetBuffer(400);
recCount=m_pRs->GetRecordCount();
if(recCount==0)
{
MessageBox("คุณ ไม่มีรายชื่อในระบบ กรุณาสักครก่อน");
m_pRs->Filter="";
status_program=4;
//1=standby 2=login 3=inputcoin 4=false 5=printed 6=printing
//-----
_tcscpy(rrr,printname);
//jane=SetJob_PRINT(3,rrr); //delete all job
//if(!jane)  MessageBox("Delete Error");
///-----
}
else
{
status_program=3;
//1=standby 2=login 3=inputcoin 4=false 5=printed 6=printing
RefreshBoundData();
SetWindowPos(NULL,0,0,800,600,SWP_SHOWWINDOW); //show me
ListJobsForPrinter(rrr);
}
}
catch(_com_error &e)
{
GenerateError(e.Error(),e.Description());
}
}

```

[illegible]

```

DWORD ttaa;
if( ! OpenPrinter( szPrinterName, &hPrinter, NULL ) )
return -1;
if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &dwNeeded,
&dwReturned ) )
{
if( GetLastError() != ERROR_INSUFFICIENT_BUFFER )
{
ClosePrinter( hPrinter );
return -1;
}
}
if( (pJobInfo = (JOB_INFO_1 *)malloc( dwNeeded )) == NULL )
{
ClosePrinter( hPrinter );
return -1;
}
if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, (LPBYTE)pJobInfo,
dwNeeded, &dwNeeded, &dwReturned ) )
{
ClosePrinter( hPrinter );
free( pJobInfo );
return -1;
}
ClosePrinter( hPrinter );
ttaa=dwReturned;
free( pJobInfo );
return ttaa;
} //-----

```

```

//-----
int CRsCgDlg::Check_printed(LPTSTR szPrinterName)
{
HANDLE    hPrinter;

//          char Show_Jane[100]; //Buffer before Show

DWORD ttaa;

DWORD     dwNeeded, dwReturned, i;
JOB_INFO_1 *pJobInfo;

if( ! OpenPrinter( szPrinterName, &hPrinter, NULL ) )
return -1;

if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &dwNeeded,
&dwReturned ) )
{
if( GetLastError() != ERROR_INSUFFICIENT_BUFFER )
{
ClosePrinter( hPrinter );
return -1;
}
}

if( (pJobInfo = (JOB_INFO_1 *)malloc( dwNeeded )) == NULL )
{
ClosePrinter( hPrinter );
return -1;
}

if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, (LPBYTE)pJobInfo,
dwNeeded, &dwNeeded, &dwReturned ) )
{
ClosePrinter( hPrinter );
free( pJobInfo );
return -1;
}
}

```

```

ClosePrinter( hPrinter );
ttaa=pJobInfo[0].PagesPrinted;
free( pJobInfo );
return ttaa;
} //-----

int CRsCgDlg::Check_TotalPage(LPTSTR szPrinterName)
{
    HANDLE    hPrinter;
    DWORD ttaa;
    DWORD    dwNeeded, dwReturned, i;
    JOB_INFO_1  *pJobInfo;
    if( ! OpenPrinter( szPrinterName, &hPrinter, NULL ) )
        return -1;
    if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &dwNeeded,
        &dwReturned ) )
    {
        if( GetLastError() != ERROR_INSUFFICIENT_BUFFER )
        {
            ClosePrinter( hPrinter );
            return -1;
        }
    }
    if( (pJobInfo = (JOB_INFO_1 *)malloc( dwNeeded )) == NULL )
    {
        ClosePrinter( hPrinter );
        return -1;
    }
    if( ! EnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, (LPBYTE)pJobInfo,
        dwNeeded, &dwNeeded, &dwReturned ) )
    {
        ClosePrinter( hPrinter );
        free( pJobInfo );
    }
}

```

```

return -1;
}

ClosePrinter( hPrinter );

ttaa=pJobInfo[dwReturned-1].PagesPrinted;
free( pJobInfo );

return ttaa;
}

void CRsCgDlg::save_data_to()
{
try
{
UpdateData(TRUE);
COleVariant column[6],datamem[6];
column[0]="Fname";//name
datamem[0]=m_strDlgFname;
column[1]="Lname";//lname
datamem[1]=m_strDlgLname;
column[2]="PASSWD";//money
datamem[2]=m_strDlgPASSWD;
column[3]="ST_ID";//id
datamem[3]=m_strDlgST_ID;
column[4]="MONEY";//money
datamem[4]=m_IDlgMONEY;
column[5]="LASTPRINTDOC";//lastprint
datamem[5]=m_strDlgLASTPRINTDOC;
VARIANT sacolumn,sadatamem;
sacolumn.vt=VT_ARRAY|VT_VARIANT;
sadatamem.vt=VT_ARRAY|VT_VARIANT;
SAFEARRAYBOUND bound[]={6,0}};
sadatamem.parray = SafeArrayCreate(VT_VARIANT,1,bound);
if(sadatamem.parray==NULL)

```

```

AfxThrowMemoryException();
sacolumn.parray=SafeArrayCreate(VT_VARIANT,1,bound);
if(sacolumn.parray==NULL)
AfxThrowMemoryException();
long index[1];
for(int i;i<6;i++)
{
    index[0]=i;
    SafeArrayPutElement(sacolumn.parray,index,&column[i]);
    SafeArrayPutElement(sadatamem.parray,index,&column[i]);
}
if(m_AddingFlag)
{
    m_pRs->Update(&sacolumn,&sadatamem);
    m_AddingFlag=FALSE;
}
else
    m_pRs->Update(&sacolumn,&sadatamem);
}
catch(_com_error &e)
{
    GenerateError(e.Error(),e.Description());
}
catch(...)
{
    MessageBox("Unknown failure");
}
}

```