



**Happ Controls**

106 Garlich Drive

Elk Grove, IL 60007

Tel: 888-289-4277 / 847-593-6130

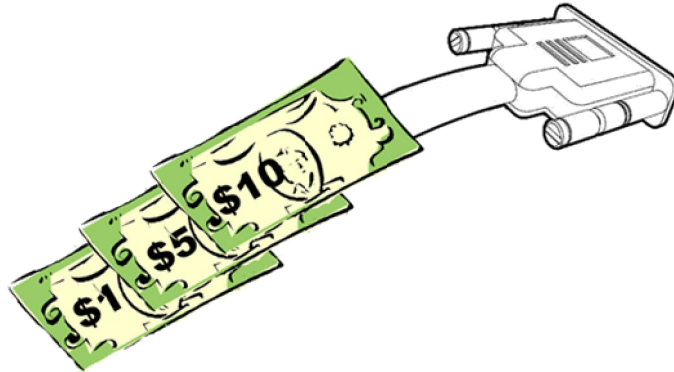
Fax: 847-593-6137

[www.happcontrols.com](http://www.happcontrols.com)

# uVI™

## Universal Validator Interface Software Developer's Kit

Revision 1.11 07/29/04



•  
•  
•  
•  
•  
•  
•

---

## Table of Contents

Table of Contents.....	2
Overview .....	3
Legal Statements .....	3
Connections .....	4
Board Communication .....	5
Basic Structure & Usage .....	5
Sample Code .....	6
Revision History.....	7

## Overview

The Universal Validator Interface is a board that allows you to connect any bill validator to a PC via the RS-232 serial port. The board is powered by a signal line on the serial port, and uses the status lines to communicate with the software. Because RS-232 programming is the most simple, in all operating systems, it was chosen for this interface. You can build your application support without having to have layers of DLL's or drivers.

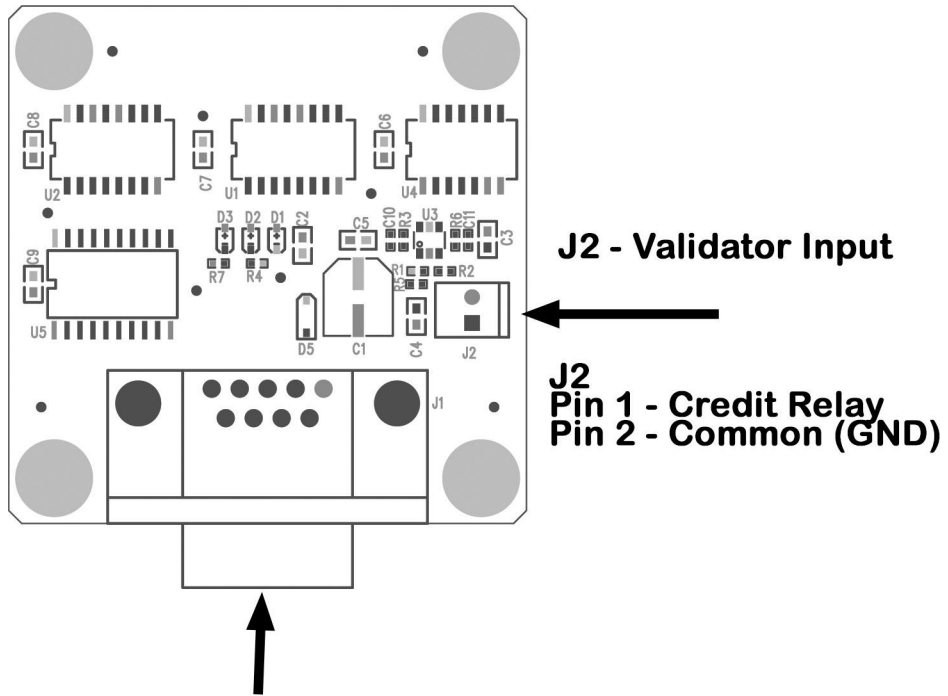
## Legal Statements

The uVI board is copyright © 2004 by UltraCade Technologies, All Rights Reserved. The uVI board and it's algorithms are Patent Pending. uVI is a trademark of UltraCade Technologies. uVI is exclusively distributed by Happ Controls.

.....

---

## Connections



**J2 - Validator Input**

**J2**  
Pin 1 - Credit Relay  
Pin 2 - Common (GND)

**J1 - RS-232 I/O**

**J1**  
Pin 3 - TX - Clear Counters  
Pin 4 - RDY - Power  
Pin 5 - GND - Ground  
Pin 7 - RTS - Read One Credit  
Pin 8 - CTS - Credits in Buffer

## Board Communication

All I/O with the board is done via status flags.

DTR – Set to High tells the board to turn on

BREAK - Pulsing this to LOW clears any pending credits in the board's buffer

RTS - Read One Credit from board buffer

CTS - There is pending credits in board's buffer

## Basic Structure & Usage

When you application starts, it should power on the board, clear anything in the buffer, and then setup a loop. Within the loop, you should monitor the CTS flag. When it goes high, issue an RTS and store 1 credit in your application. Check the CTS flag again, and repeat until all credits have been cleared from the buffer.

Example:

- Validator set to 1 pulse per dollar
- Initialize Application, set DTR, pulse BREAK to low
- Start loop monitoring CTS
- Place \$5 bill in validator
- CTS goes high
- Loop 5 times issuing RTS
- CTS goes low

## Sample Code

```
include <windows.h>
#include "resource.h"

BOOL CALLBACK MainDlg(HWND hWind, UINT nMsg, WPARAM nWParam, LPARAM nLParam);
void uVInit(void);
void uVITerm(void);

HWND hMain;
HINSTANCE hInstance;
HANDLE hComm;
int bCoin=FALSE;

int WINAPI
WinMain(HINSTANCE hInst, HINSTANCE hPrev, LPSTR pArgs, int nShow)
{
    hInstance = hInst;
    hComm = CreateFile("COM1", GENERIC_READ|GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0, NULL);
    if (hComm==NULL)
    {
        MessageBox(NULL, "Couldn't open COM1", "CashBoard Test", MB_ICONERROR|MB_OK);
        return(0);
    }
    uVInit();
    DialogBox(hInst, MAKEINTRESOURCE(IDD_MAIN), NULL, MainDlg);
    uVITerm();
    CloseHandle(hComm);
    return(0);
}

BOOL CALLBACK
MainDlg(HWND hWind, UINT nMsg, WPARAM nWParam, LPARAM nLParam)
{
    int nID,nCmd;
    int v;
    int bNew;

    switch (nMsg)
    {
    case WM_INITDIALOG:
        hMain = hWind;
        SetTimer(hWind, 1, 100, NULL);
        return(TRUE);

    case WM_CLOSE:
        EndDialog(hWind, 0);
        break;

    case WM_COMMAND:
        nID = LOWORD(nWParam);
        nCmd = HIWORD(nWParam);
        if ((nCmd==BN_CLICKED) && (nID==IDC_EXIT))
            EndDialog(hWind, 0);
        if ((nCmd==BN_CLICKED) && (nID==IDC_CLEAR))
        {
            EscapeCommFunction(hComm, CLRBREAK);
            Sleep(100);
            EscapeCommFunction(hComm, SETBREAK);
        }
        if ((nCmd==BN_CLICKED) && (nID==IDC_READ))
        {
            EscapeCommFunction(hComm, CLRRTS);
            Sleep(100);
            EscapeCommFunction(hComm, SETRTS);
        }
        break;

    case WM_TIMER:
        GetCommModemStatus(hComm, &v);
        bNew = (v&MS_CTS_ON) ? TRUE : FALSE;
        if (bNew!=bCoin)
        {
            bCoin = bNew;
            SendDlgItemMessage(hWind, IDC_SIGNAL, WM_SETTEXT, 0, (LPARAM)(bCoin?"More Coins":"Empty"));
        }
    }
    return(FALSE);
}
```

```

void
uVInit(void)
{
    /* Turn off Clear and Read */
    EscapeCommFunction(hComm, SETRTS);
    EscapeCommFunction(hComm, SETBREAK);

    /* Power up board */
    EscapeCommFunction(hComm, SETDTR);
    Sleep(100);

    /* Clear board */
    EscapeCommFunction(hComm, CLRBREAK);
    Sleep(100);
    EscapeCommFunction(hComm, SETBREAK);
}

void
uVTerm(void)
{
    /* Power off board */
    EscapeCommFunction(hComm, CLR DTR);

    /* Set to normal values */
    EscapeCommFunction(hComm, CLR RTS);
    EscapeCommFunction(hComm, CLR BREAK);
}

```

## Revision History

- June 24, 2004, DRF, 1.1 Update
  - Updated SDK so that poweron function is always called during initialization
  - Removed Power On / Power Off toggle from UI
- April 26, 2004, DRF, Original Version