HMExcel

Stand: 11.03.2017

Automatisierung („Fernsteuerung“) von Excel unter Microsoft Windows

Tilman Küpper (tilman.kuepper@hm.edu)

Inhalt

1. Einleitung

HMExcel ermöglicht die Automatisierung („Fernsteuerung“) der Tabellenkalkulation Microsoft Excel unter dem Betriebssystem Microsoft Windows. Der Einsatz von HMExcel unter anderen Betriebssystemen wie Linux oder Mac OS X ist in der aktuellen Version nicht möglich.

In der Entwicklungsumgebung sind die Dateien **xlauto.c** und **xlauto.h** zum Projekt hinzuzufügen. Die folgende Abbildung zeigt Microsoft Visual C++ 2015 mit einem geöffneten Projekt. (Achtung: Bei anderen Entwicklungsumgebungen müssen evtl. auch die Bibliotheken ole32.lib, oleaut32.lib und uuid.lib „manuell“ zum Projekt hinzugefügt werden!).

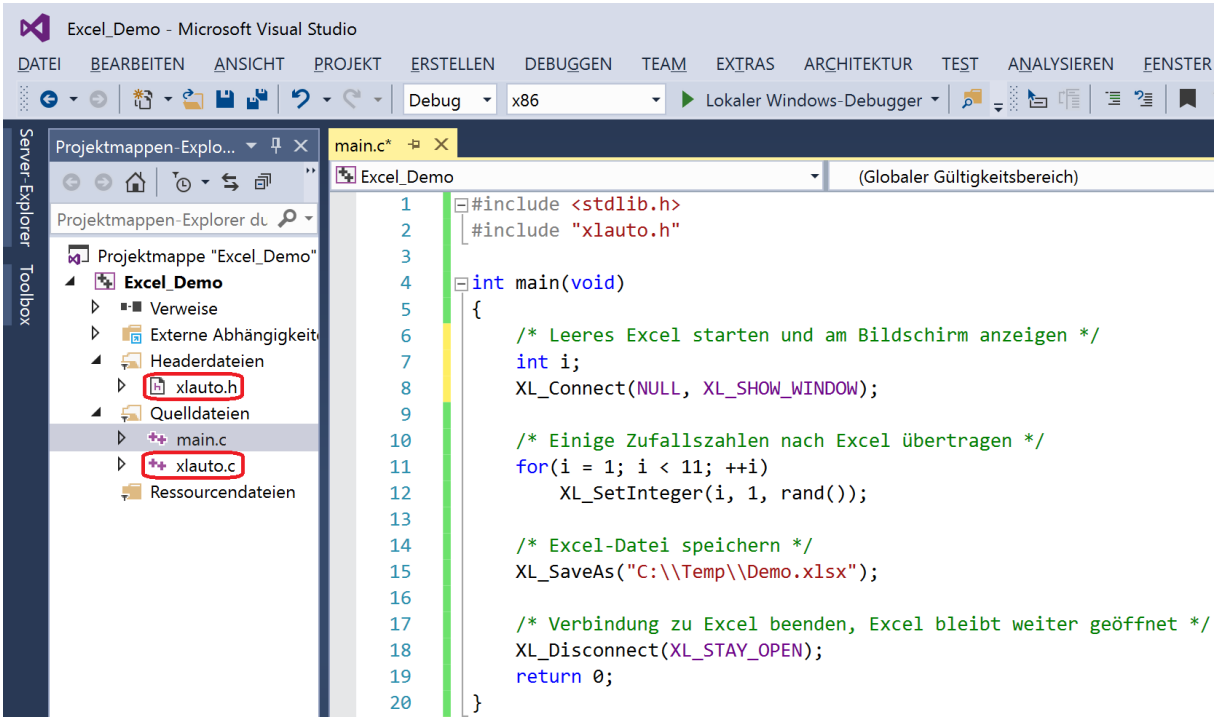


Abbildung 1 – Benutzeroberfläche von Microsoft Visual C++ 2015

2. Beispiele

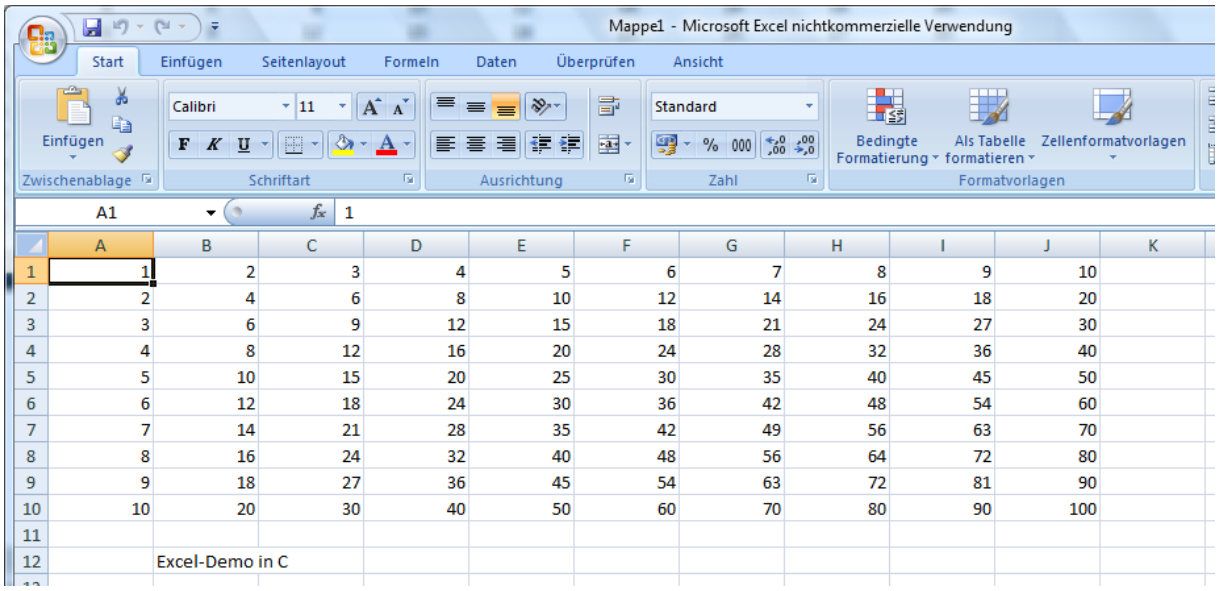
2.1. Daten in ein Tabellenblatt schreiben

```
#include "xlauto.h"
int main(void)
{
    /* Leeres Excel starten und am Bildschirm anzeigen */
    int row, col;
    XL_Connect(NULL, XL_SHOW_WINDOW);

    /* Einmaleins-Tabelle berechnen und nach Excel übertragen */
    for(row = 1; row < 11; ++row)
        for(col = 1; col < 11; ++col)
            XL_SetInteger(row, col, row * col);

    /* Text in Zelle B12 ausgeben */
    XL_SetString(12, 2, "Excel-Demo in C");

    /* Verbindung zu Excel beenden, Excel bleibt weiter geöffnet */
    XL_Disconnect(XL_STAY_OPEN);
    return 0;
}
```



	A	B	C	D	E	F	G	H	I	J	K
1	1	2	3	4	5	6	7	8	9	10	
2	2	4	6	8	10	12	14	16	18	20	
3	3	6	9	12	15	18	21	24	27	30	
4	4	8	12	16	20	24	28	32	36	40	
5	5	10	15	20	25	30	35	40	45	50	
6	6	12	18	24	30	36	42	48	54	60	
7	7	14	21	28	35	42	49	56	63	70	
8	8	16	24	32	40	48	56	64	72	80	
9	9	18	27	36	45	54	63	72	81	90	
10	10	20	30	40	50	60	70	80	90	100	
11											
12		Excel-Demo in C									

Abbildung 2 – Daten in ein Tabellenblatt schreiben

2.2. Daten aus einer bestehenden Excel-Datei lesen

```c
#include <stdio.h>
#include "xlauto.h"
int main(void)
{
    /* Zeilen und Spalten werden ab 1 gezählt */
    int row = 1, col = 1;

    /* Vorhandene Datei öffnen, Excel bleibt unsichtbar im Hintergrund */
    XL_Connect("C:\\Temp\\Mappe1.xlsx", XL_HIDE_WINDOW);

    /* Arbeitsblatt "Tabelle1" auswählen */
    XL_SelectWorksheet("Tabelle1");

    /* Alle Zahlen aus Spalte A von oben nach unten lesen und ausgeben */
    printf("Spalte A:\n\n");
    while(XL_IsEmpty(row, col) == XL_NOT_EMPTY)
    {
        double value = XL_GetDouble(row, col);
        printf("A%d = %f\n", row, value);
        ++row;
    }

    /* Excel schließen und beenden */
    XL_Disconnect(XL_CLOSE);
    return 0;
}
```

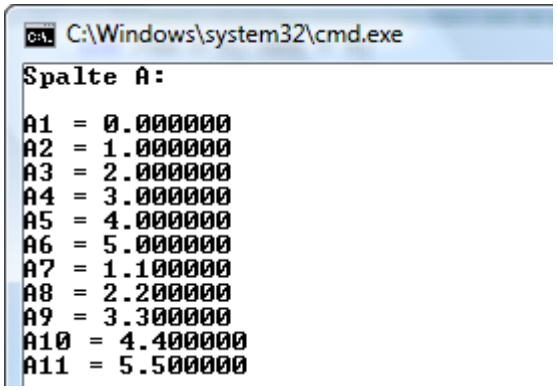


Abbildung 3 – Ausgabe des C-Programms

	A	B	C	D	E
1	0				
2	1				
3	2				
4	3				
5	4				
6	5				
7	1,1				
8	2,2				
9	3,3				
10	4,4				
11	5,5				

Abbildung 4 – Inhalt von C:\Temp\Test.xlsx

2.3. Schreiben und Lesen von Zeichenketten

```c
#include <stdio.h>
#include "xlauto.h"
int main(void)
{
    char txt[100];

    /* Leeres Excel starten und am Bildschirm anzeigen */
    XL_Connect(NULL, XL_SHOW_WINDOW);

    /* Text in Excel-Tabellenblatt schreiben */
    XL_SetString(1, 1, "Text in Zelle A1...");
    XL_SetString(2, 3, "Text in Zelle C2...");

    /* Text aus Excel-Tabellenblatt wieder einlesen */
    XL_GetString(1, 1, txt, sizeof(txt)); printf("%s\n", txt);
    XL_GetString(2, 3, txt, sizeof(txt)); printf("%s\n", txt);

    /* Auch wchar_t-Zeichenketten können geschrieben werden */
    XL_SetWString(3, 1, L"从慕尼黑许多问候");

    /* Verbindung zu Excel beenden, Excel bleibt weiter geöffnet */
    XL_Disconnect(XL_STAY_OPEN);
    return 0;
}
```

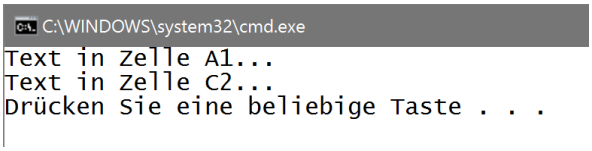


Abbildung 5 – Ausgabe des C-Programms

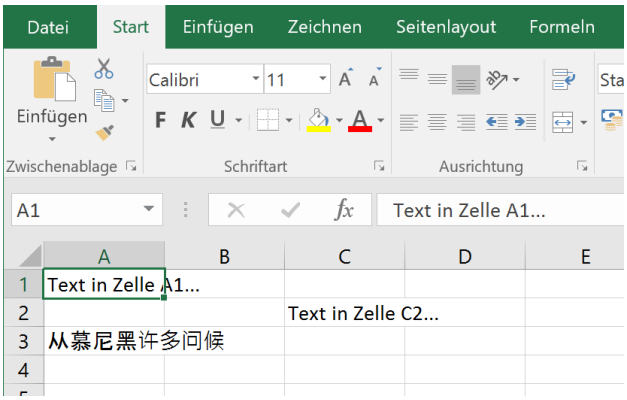


Abbildung 6 – Textfelder in Microsoft Excel

2.4. Excel-Tabelle „im Hintergrund“ erstellen und abspeichern

```c
#include <stdlib.h>
#include "xlauto.h"

int main(void)
{
    /* Leeres Excel starten, nicht am Bildschirm anzeigen */
    int i;
    XL_Connect(NULL, XL_HIDE_WINDOW);

    /* Einige Zufallszahlen nach Excel übertragen */
    for(i = 1; i < 11; ++i)
        XL_SetInteger(i, 1, rand());

    /* Excel-Tabelle speichern */
    XL_SaveAs("C:\\Temp\\Demo.xlsx");

    /* Excel wieder beenden */
    XL_Disconnect(XL_CLOSE);
    return 0;
}
```

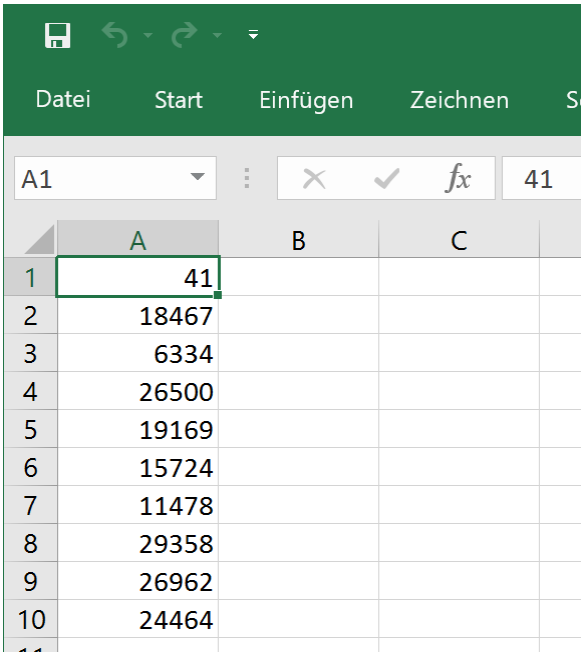


Abbildung 7 – Inhalt der abgespeicherten Excel-Tabelle

3. Funktionsübersicht

- `int XL_Connect(const char *filename, int show_window);`

  Verbindung zu Microsoft Excel herstellen. Der Parameter show_window gibt an, ob Excel sichtbar (XL_SHOW_WINDOW) oder unsichtbar (XL_HIDE_WINDOW) gestartet werden soll. Mit dem Parameter filename wird der Name der zu öffnenden Excel-Datei übergeben, bzw. filename == NULL, falls keine Datei geöffnet werden soll. Mögliche Rückgabewerte: XL_OK, XL_ERR_NOT_FOUND, XL_ERR_NOT_INSTALLED oder XL_ERR_ALREADY_RUNNING.

- `int XL_Disconnect(int mode);`

  Verbindung zu Microsoft Excel beenden, bei mode == XL_STAY_OPEN bleibt Excel weiterhin offen, bei mode == XL_CLOSE wird Excel geschlossen. Mögliche Rückgabewerte: XL_OK oder XL_ERR_NOT_RUNNING.

- `int XL_SelectWorksheet(const char *name);`

  Innerhalb der aktuellen Excel-Arbeitsmappe wird das Arbeitsblatt mit dem angegebenen Namen (zum Beispiel "Tabelle1") gesucht und aktiviert. Mögliche Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_NOT_FOUND.

- `int XL_SetInteger(int row, int col, int value);`

  Eine Integer-Zahl wird an der angegebenen Position in das Arbeitsblatt geschrieben. Achtung: Die Zeilen- bzw. Spaltennummerierung ist eins-basiert, die oberste linke Zelle befindet sich an der Position row = 1 und col = 1. Mögliche Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_WRITE_FAILED.

- `int XL_SetDouble(int row, int col, double value);`

  Eine Double-Zahl wird an der angegebenen Position in das Arbeitsblatt geschrieben. Achtung: Die Zeilen- bzw. Spaltennummerierung ist eins-basiert, die oberste linke Zelle befindet sich an der Position row = 1 und col = 1. Mögliche Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_WRITE_FAILED.

- `int XL_SetString(int row, int col, const char *value);`
- `int XL_SetWString(int row, int col, const wchar_t *wvalue);`

  Ein String (wahlweise const char* oder const wchar_t*) wird an der angegebenen Position in das Excel-Arbeitsblatt geschrieben. Achtung: Die Zeilen-bzw. Spaltennummerierung ist eins-basiert, die oberste linke Zelle befindet sich an der Position row = 1 und col = 1. Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_WRITE_FAILED.

- `int XL_GetInteger(int row, int col);`

  Die Integer-Zahl an der angegebenen Position des Arbeitsblatts wird zurückgegeben. Hinweis: Falls die Zelle an der angegebenen Position leer ist, keine Zahl enthält oder ein Fehler auftritt (falls z. B. Excel nicht gestartet wurde), ist der Rückgabewert gleich null.

- `double XL_GetDouble(int row, int col);`

  Die Double-Zahl an der angegebenen Position des Arbeitsblatts wird zurückgegeben. Hinweis: Falls die Zelle an der angegebenen Position leer ist, keine Zahl enthält oder ein Fehler auftritt (falls z. B. Excel nicht gestartet wurde), ist der Rückgabewert gleich null.

- `int XL_GetString(int row, int col, char *result, size_t len);`

  Der Zellen-Inhalt an der angegebenen Position des Arbeitsblatts wird als String zurückgegeben. Beim Aufruf der Funktion XL_GetString muss ein Zeiger (result) auf einen Buffer von ausreichender Größe (len) für die Rückgabe des Zellen-Inhalts bereitgestellt werden. Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_READ_FAILED.

- `int XL_IsEmpty(int row, int col);`

  Eine Zelle (Reihe = row, Spalte = col) im aktuellen Excel-Arbeitsblatt wird darauf überprüft, ob sie leer ist. Mögliche Rückgabewerte: XL_EMPTY, XL_NOT_EMPTY, XL_ERR_READ_FAILED, XL_ERR_NOT_RUNNING.

- `int XL_SaveAs(const char *filename);`

  Die aktuelle Arbeitsmappe wird unter dem angegebenen Dateinamen auf der Festplatte gespeichert. Rückgabewerte: XL_OK, XL_ERR_NOT_RUNNING oder XL_ERR_SAVE_FAILED (falls beim Speichern ein Fehler aufgetreten ist).

4. Hinweise

- HMExcel kann nur auf solchen Computern eingesetzt werden, die unter dem Betriebssystem Microsoft Windows laufen und auf denen Microsoft Excel installiert ist.
- Je nach eingesetzter Entwicklungsumgebung müssen evtl. die folgenden Bibliotheken „manuell“ zum Projekt hinzugefügt werden: ole32.lib, oleaut32.lib und uuid.lib. Bei Qt Creator muss beispielsweise die Zeile „`LIBS += -lole32 -loleaut32 -luuid`“ in der Projektdatei eingefügt werden. (Bei Microsoft Visual C++ und LccWin32 geschieht dies automatisch.)
- Bevor Daten in ein Excel-Tabellenblatt geschrieben bzw. von dort gelesen werden können, muss mittels XL_Connect eine Verbindung zu Microsoft Excel geöffnet werden. Dabei kann eine bestehende Excel-Datei geladen oder eine neue, leere Excel-Instanz gestartet werden:

  ```
  /* Leeres Excel starten und am Bildschirm anzeigen */
  XL_Connect(NULL, XL_SHOW_WINDOW);
  ```

  Der folgende Funktionsaufruf dient zum Laden einer bestehenden Excel-Datei – man beachte die doppelten Rückwärts-Schrägstriche ("Backslash") in der Pfadangabe:

  ```
  /* Vorhandene Datei öffnen, Excel bleibt unsichtbar im Hintergrund */
  XL_Connect("C:\\Temp\\Mappe1.xlsx", XL_HIDE_WINDOW);
  ```

- Wenn der Datenaustausch mit Microsoft Excel abgeschlossen wurde – spätestens aber zum Ende des Programms – muss die Verbindung zu Microsoft Excel wieder geschlossen werden. Wird dies vergessen, bleiben inaktive Excel-Prozesse "hängen", die nur über den Task-Manager beendet werden können:

  ```
  /* Excel schließen und beenden */
  XL_Disconnect(XL_CLOSE);
  ```

  Auf Wunsch kann Excel auch nach dem Ende des C-Programms weiterhin geöffnet bleiben:

  ```
  /* Verbindung zu Excel beenden, Excel bleibt weiter geöffnet */
  XL_Disconnect(XL_STAY_OPEN);
  ```

- Die Programmierschnittstelle ist bewusst einfach gehalten. So sind zur Nutzung von HMExcel keine Zeiger oder "Handles" erforderlich. Dies wurde durch die Speicherung des aktuellen Verbindungszustands in statischen, globalen Variablen erreicht. HMExcel ist aus diesem Grund nicht threadsicher: Falls in einer Applikation mehrere parallele Threads existieren, kann HMExcel nur in einem dieser Threads genutzt werden.
- Herzlichen Dank an Hans-Peter Berger, Firma LuK, für seine Rückmeldungen und Anregungen!

5. Kontakt

Tilman Küpper
tilman.kuepper@hm.edu

Hochschule München
Fakultät für Maschinenbau, Fahrzeugtechnik, Flugzeugtechnik
Dachauer Straße 98 B
D-80335 München

http://kuepper.userweb.mwn.de

6. Lizenz

Die Funktions- und Klassenbibliothek HMExcel darf unter Beachtung der folgenden Lizenzbedingungen frei verwendet und weitergegeben werden: