

Weichinger Klaus - Weinga-Unity Mechatronik- und EDV-Einzelunternehmer Tätigkeiten, Leistungen, Kontaktinformationen

Weichinger Klaus
snaky.1@gmx.at

August 2006

Übersicht über die Tätigkeitsbereiche

Tätigkeitsbereiche

- Mikrocontroller Programmierung
- Hardware Entwicklung
- Automatisierungstechnik - Wissenschaftliche Anwendungen
- PC Softwareentwicklung

Mikrocontroller-Programmierung

Typen und Sprachen

- 8051 in C und Assembler
- AVR in C (und Assembler)
- MSP430 in C

Mikrocontroller-Programmierung

Typen und Sprachen

- 8051 in C und Assembler
- AVR in C (und Assembler)
- MSP430 in C

Funktionen

- 1wire, I2C, SPI
- LCD-Displays (Graphik)
- Funk, Infrarot
- RS232, RS485, USB
- EEPROM
- AD-DA-Wandler, PWM

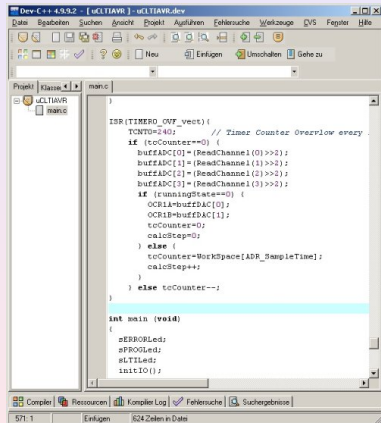
Mikrocontroller-Programmierung

Typen und Sprachen

- 8051 in C und Assembler
- AVR in C (und Assembler)
- MSP430 in C

Funktionen

- 1wire, I2C, SPI
- LCD-Displays (Graphik)
- Funk, Infrarot
- RS232, RS485, USB
- EEPROM
- AD-DA-Wandler, PWM



```
Dev-C++ 4.9.9.2 - [uCLTI.AVR] - uCLTI.AVR.dcx
Date  Spalten  Suchen  Ansicht  Projekt  Applikationen  Fehlersuche  Werkzeuge  DV5  Register  Hilfe

Projekt  Klasse  uCLTI.AVR
        manco

ISR(TIMER0_OVF_vect) {
    TCNT0=240; // Timer Counter Overflow every
    if (toCounter==0) {
        buffADC[0] = (ReadChannel(0)>>2);
        buffADC[1] = (ReadChannel(1)>>2);
        buffADC[2] = (ReadChannel(2)>>2);
        buffADC[3] = (ReadChannel(3)>>2);
        if (runningState==0) {
            OCR1A=buffADC[0];
            OCR1B=buffADC[1];
            toCounter=0;
            calcStep=0;
        } else {
            toCounter=WorkSpace[ADR_SampleTime];
            calcStep++;
        } else toCounter--;
    }
}

int main (void)
{
    nERRORLed;
    nFROGLed;
    sLTiled;
    initIO();
}
```

Mikrocontroller-Programmierung

Typen und Sprachen

- 8051 in C und Assembler
- AVR in C (und Assembler)
- MSP430 in C

Funktionen

- 1wire, I2C, SPI
- LCD-Displays (Graphik)
- Funk, Infrarot
- RS232, RS485, USB
- EEPROM
- AD-DA-Wandler, PWM

```

Software Development Tool V2.3 | uCSoftware.sdp
Date: Projekt Bearbeiten Suchen Tools Hilfe
Projekt Dateien: | uCSoftware.sdp
  uCSoftware [MAIN]
  uCSoftware.lst
  Includes
  commands.inc
  drivers.inc
  i8048.inc
  i8049.inc
  protocol.inc
  serialCa.inc
  Dokumentation
  MCS1 Befehle.pdf
  Testfiles
  MUI
  Lesesuchen
  Definitionen
  CODE BEGIN
  INITIALISIERUNG
  HAUPTPROGRAMM

doCommand:
    CJNE A, #00, noCommandP
    CALL doCommandP
    noCommandP:
    CJNE A, #03, noCommandS
    CALL doCommandS
    noCommandS:
    CJNE A, #06, noCommandV
    CALL doCommandV
    noCommandV:
    CJNE A, #00, noCommandI
    CALL doCommandI
    noCommandI:
    RET

:#####->HAUPTPROGRAMM<-#####
HAUPTPROGRAMM:  ##. .HAUPTPROGRAMM: Here
                CALL    re232Get          ; Rea
                ; C=1
                JNC     noReceivedByte    ; 3W
                MOV     receivedByte, A
                CALL    doCommand
                noReceivedByte:
                JMP     HAUPTPROGRAMM
                END
    
```

Line 0: Col 0 Line 0 to 54

Developed by
weinw-Unity
technology for e-unity

Mikrocontroller-Programmierung

Typen und Sprachen

- 8051 in C und Assembler
- AVR in C (und Assembler)
- MSP430 in C

Funktionen

- 1wire, I2C, SPI
- LCD-Displays (Graphik)
- Funk, Infrarot
- RS232, RS485, USB
- EEPROM
- AD-DA-Wandler, PWM



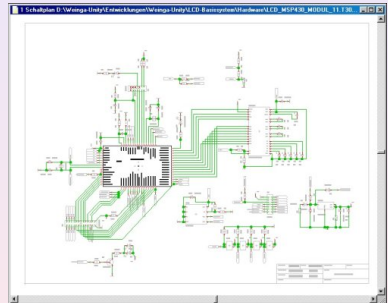
Hardware-Entwicklung

Hardware-Entwicklung

Hardware-Entwicklung

Hardware-Entwicklung

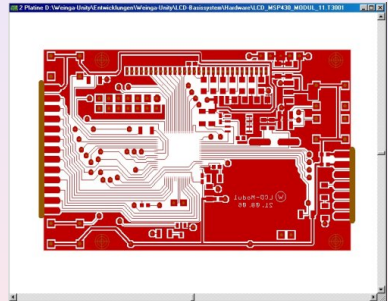
- Schaltplanentwurf



Hardware-Entwicklung

Hardware-Entwicklung

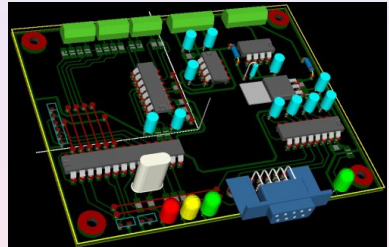
- Schaltplanentwurf
- Layout und 3D Ansicht



Hardware-Entwicklung

Hardware-Entwicklung

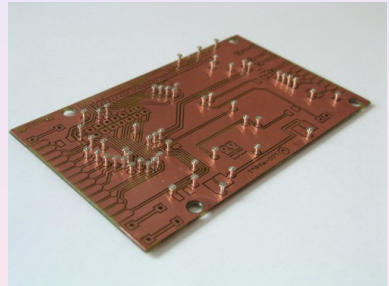
- Schaltplanentwurf
- Layout und 3D Ansicht



Hardware-Entwicklung

Hardware-Entwicklung

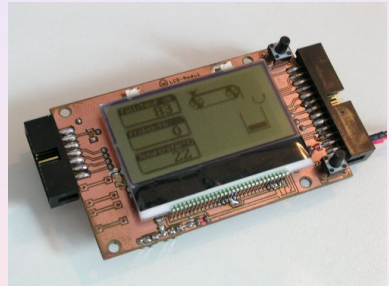
- Schaltplanentwurf
- Layout und 3D Ansicht
- Fertigung eines Prototypen



Hardware-Entwicklung

Hardware-Entwicklung

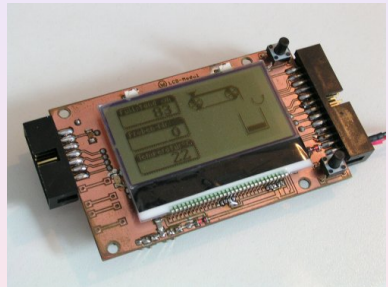
- Schaltplanentwurf
- Layout und 3D Ansicht
- Fertigung eines Prototypen
- Bestückung des Prototypen



Hardware-Entwicklung

Hardware-Entwicklung

- Schaltplanentwurf
- Layout und 3D Ansicht
- Fertigung eines Prototypen
- Bestückung des Prototypen
- Download der Software in die Hardware



Automatisierungstechnik - Wissenschaftliche Anwendungen

Entwurf Digitale Regler und Systeme

- Mathematische Modellbildung
- Identifizierung und Diskretisierung des Systems
- Lineare Regler mittels Zustandsbeobachter und Zustandsregler
- Reglerentwurf mit Frequenzkennlinienverfahren (PID, PI)

Automatisierungstechnik - Wissenschaftliche Anwendungen

Entwurf Digitale Regler und Systeme

- Mathematische Modellbildung
- Identifizierung und Diskretisierung des Systems
- Lineare Regler mittels Zustandsbeobachter und Zustandsregler
- Reglerentwurf mit Frequenzkennlinienverfahren (PID, PI)

Implementierung der Regler

- Auf PC und Mikrocontroller (AVR, MSP430)
- Programmierung in C (C/C++ bei PC)

Automatisierungstechnik - Wissenschaftliche Anwendungen

Entwurf Digitale Regler und Systeme

- Mathematische Modellbildung
- Identifizierung und Diskretisierung des Systems
- Lineare Regler mittels Zustandsbeobachter und Zustandsregler
- Reglerentwurf mit Frequenzkennlinienverfahren (PID, PI)

Implementierung der Regler

- Auf PC und Mikrocontroller (AVR, MSP430)
- Programmierung in C (C/C++ bei PC)

Umsetzung in Form eines diskreten LTI-Systems

$$\vec{x}_{k+1} = \mathbf{A} \vec{x}_k + \mathbf{B} \vec{u}_k$$

$$\vec{y}_k = \mathbf{C} \vec{x}_k + \mathbf{D} \vec{u}_k$$

PC-Softwareentwicklung

Programmiersprachen

- Objekt Pascal (Delphi)
- C/C++
- Basic-Dialekte
- Bei Bedarf auch JAVA
und andere
Programmiersprachen

PC-Softwareentwicklung

Programmiersprachen

- Objekt Pascal (Delphi)
- C/C++
- Basic-Dialekte
- Bei Bedarf auch JAVA und andere Programmiersprachen

Gebiete

- Objektorientierte Programmierung
- RS232, USB
- Flexibilität durch Script-Funktionen
- Datenbankbindung (SQL)
- FTP, E-Mail, ...
- GUI - Benutzeroberflächen
- Visualisierungen
- Anfertigung von Setups

PC-Softwareentwicklung

Programmiersprachen

- Objekt Pascal (Delphi)
- C/C++
- Basic-Dialekte
- Bei Bedarf auch JAVA und andere Programmiersprachen

Plattformen

- WIN32
- Parallel wird jetzt auch mit LINUX gearbeitet
- PDA's

Gebiete

- Objektorientierte Programmierung
- RS232, USB
- Flexibilität durch Script-Funktionen
- Datenbankbindung (SQL)
- FTP, E-Mail, ...
- GUI - Benutzeroberflächen
- Visualisierungen
- Anfertigung von Setups

PC-Softwareentwicklung Beispiele-Setup



PC-Softwareentwicklung Beispiele-RS232 Tool

The screenshot shows the RS232-Access V1.2 software interface. The window title is "RS232-Access V1.2" and the address bar shows "http://www.weinga-unity.de.vu".

Protocol: Port open

FS232-Settings:

- COM 1 (selected)
- Baud: 2400
- Data bits: 8
- Stop bits: 1
- Parity: None
- Buttons: Close COM, Scan COMs

FS232 Digital I/O Lines:

- DDD Pin 1 (IN)
- DTR Pin 4 (OUT)
- DSR Pin 6 (IN)
- RTS Pin 7 (OUT)
- DTS Pin 8 (IN)
- RING Pin 9 (IN)

Binary Interface:

Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0

Binary Input: 97 (LEDs: Bit7 red, Bit6 red, Bit5 red, Bit4 red, Bit3 red, Bit2 red, Bit1 red, Bit0 red)

Binary Output: 97 (LEDs: Bit7 green, Bit6 green, Bit5 green, Bit4 green, Bit3 green, Bit2 green, Bit1 green, Bit0 green)

Buttons: Send, Clear-Protocols, Port Open

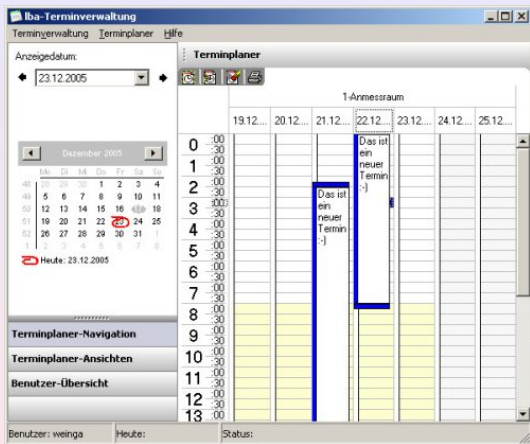
Protocol Display: Clear-Protocols, Protokoll Display, Binary LED Interface

Log:

```

Received Byte: DEC: 81  BIN: 01010001
Sending Byte:  DEC: 83  BIN: 01010011
Received Byte: DEC: 83  BIN: 01010011
Sending Byte:  DEC: 91  BIN: 01101011
Received Byte: DEC: 91  BIN: 01101011
Sending Byte:  DEC: 93  BIN: 01111011
Received Byte: DEC: 93  BIN: 01111011
Sending Byte:  DEC: 94  BIN: 01111110
Received Byte: DEC: 94  BIN: 01111110
Sending Byte:  DEC: 96  BIN: 01100000
Received Byte: DEC: 96  BIN: 01100000
Sending Byte:  DEC: 97  BIN: 01100001
Received Byte: DEC: 97  BIN: 01100001
Sending Byte:  DEC: 97  BIN: 01100001
Received Byte: DEC: 97  BIN: 01100001
Sending Byte:  DEC: 97  BIN: 01100001
Received Byte: DEC: 97  BIN: 01100001
Received Byte: DEC: 97  BIN: 01100001
    
```

PC-Softwareentwicklung Beispiele-Terminverwaltung



Frontend für ein existierendes Terminverwaltungssystem
Fa. SOP Hilmbauer & Mauberger GmbH & CoKG

<http://www.sop.co.at>

PC-Softwareentwicklung Beispiel-SDT

The screenshot displays the 'Software Development Tool V2.3.1' interface. The main window shows the assembly code for 'uCSoftware.sdp'. The code includes a loop structure for handling commands, with labels like 'doCommand', 'noCommandP', 'noCommandS', 'noCommandV', and 'noCommandZ'. It also features a main program section starting with 'HAUPTPROGRAMM' and a 'RET' instruction at the end.

```

doCommand:
    CJNE  A, #80, noCommandP
        CALL doCommandP
    noCommandP:
    CJNE  A, #83, noCommandS
        CALL doCommandS
    noCommandS:
    CJNE  A, #86, noCommandV
        CALL doCommandV
    noCommandV:
    CJNE  A, #90, noCommandZ
        CALL doCommandZ
    noCommandZ:
RET

;#####>HAUPTPROGRAMM<#####
HAUPTPROGRAMM: ;##..HAUPTPROGRAMM: Her
    CALL  rs232Get      : Res
                    : C=1

    JNC   noReceivedByte : jump
        MOV  receivedByte, A
        CALL doCommand
    noReceivedByte:

JMP  HAUPTPROGRAMM
END
    
```

The interface also includes a project tree on the left, a 'Lesezeichen' (Bookmarks) panel on the right, and a status bar at the bottom showing 'Line 0: Col 0' and 'Line 0 to 54'. A logo for 'Weinga-Unity Technology for Embedded Systems' is visible in the bottom right corner of the tool window.

Dokumentation

Erstellung von Dokumentationen

- Source-Code in Form von Kommentaren im Code
- Bedienungsanleitung in HTML oder PDF
- Minutengenaue Tätigkeitserfassung für die Abrechnung
- Alle relevanten Daten und Recherchen werden zusammengestellt und bei Projektende mittels CD übergeben

Dokumentation

Erstellung von Dokumentationen

- Source-Code in Form von Kommentaren im Code
- Bedienungsanleitung in HTML oder PDF
- Minutengenaue Tätigkeitserfassung für die Abrechnung
- Alle relevanten Daten und Recherchen werden zusammengestellt und bei Projektende mittels CD übergeben

Verwendete Tools

- \LaTeX für die Bedienungsanleitungen und Zusammenstellungen

Arbeitsweise - Verrechnung

Arbeitsweise

- Es wird versucht, möglichst kostenlose bzw. OpenSource Programme und Bibliotheken zu verwenden.
- Vermeidung möglicher Abhängigkeiten (möglichst standalone EXE-Programme)

Arbeitsweise - Verrechnung

Arbeitsweise

- Es wird versucht, möglichst kostenlose bzw. OpenSource Programme und Bibliotheken zu verwenden.
- Vermeidung möglicher Abhängigkeiten (möglichst standalone EXE-Programme)

Verrechnung

- Es wird der Zeitaufwand verrechnet, wobei Abschätzungen des Aufwandes möglich sind.
- Ist eine Abschätzung nicht möglich, wird dies offen geäußert.
- Der Zeitaufwand wird minutengenau dokumentiert.

Kontaktinformationen



Weichinger Klaus
A-3303 Greinsfurth - Feldstr. 16
www.weinga-unity.at.tt
snaky.1@gmx.at