

## **iCon© 2.0** „Intelligent Connector“

... ist die Bezeichnung für eine Softwarelösung, die die Adobe© Flash Entwicklungsumgebungen Flex© , Flash© oder Open Source Flash Entwicklungsumgebungen (MTASC) für Embedded Systems nutzbar machen soll. Version 2.0 ist verfügbar auf: Exm32 Plattform von MSC (Mipsi), Tonfunk© BMCU Mopad (Arm) und iPAN7 (Keith & Koep GmbH).

Über iCon© 2.0 kann die Embedded Hardware direkt über eine Klassenbibliothek angesprochen werden. Hierdurch werden die normalerweise existierenden Einschränkungen des Flash Players beseitigt.

### **Vorteile**

- Schnellere Softwareentwicklung grafisch anspruchsvoller, ergonomischer Applikationen
- Trennung von Design und Programmierung im Entwicklungsprozess
- Projektmanagement orientiert sich mehr am Wasserfall-Modell als am V-Modell. Dies ermöglicht eine frühere Einbindung des Kunden schon in der Spezifikationsphase, durch schnellere Design Vorschläge zur Bedienungsoberfläche.
- Steigerung der Attraktivität von Embedded OEM Produkten für potentielle Kundenkreise

### **Voraussetzungen**

- Windows© CE 5.0  
Adobe© Flash Player 7 SDK
- Windows© CE 6.0 Release 3  
Adobe© Flash Player 3.1 / 2.1 oder Flash Player Browser Plugin
- Windows Compact 7  
Flash Player Plugin 10.1 (AS 3.0)
- **iCon© 2.0** bestehend aus conn.exe und iCon\_20.swc (Flash Komponente)

### **Konzept**

biobedded systems GmbH liefert iCon© 2.0 im Rahmen eines Lizenzabkommens nur an OEM Partner aus. Die Lizenz ist eine Bulk Lizenz und wird einzeln vereinbart. Die Lösung wird im Rahmen eines Partnervertrages seitens des OEM auf Messen und zur Kundenpräsentation vorgestellt. biobedded systems GmbH verfolgt zwei strategische Ziele: Zum einen den Verkauf von Bulk Lizenzen, zum anderen die Akquise von Anwendungssoftwareprojekten.

Der Adobe© Flash Player ist eine Komponente die zur Zeit in Millionen von Geräten installiert ist. Aus diesem Grunde kann Flash vom Sicherheitsstandard her nur eine „Kernansprache“ von lokalen Ressourcen anbieten. Die üblicherweise vom Betriebssystem mittels individueller Treiber angesprochenen Hardware-Komponenten von OEM Embedded Produkten werden

nicht unterstützt. Hier schafft iCon© einen Konnektor, der diese Komponenten der Flash Entwicklungsumgebung direkt anbindet.

Der Entwickler kann somit ohne Kenntnisse von C++ und den damit verbundenen Frameworks wie NET 2.0 usw. auf Embedded CE Systemen „native“ in Adobe© Flash arbeiten.

Die Zusammenarbeit mit entsprechenden interessierten OEMs benötigt für die Portierung und Anpassung von iCon© einen professionellen Ansprechpartner der im Hause des OEMs für die Generierung der BSPs zuständig ist. Die Schnittstellenspezifikation der individuellen Treiber muss biobedded systems GmbH mitgeteilt werden.



Für Universitäten und Schulen ist iCon© zu besonderen Konditionen erhältlich.

#### **Zurzeit existierende Befehle**

```
name_ikon_instanz = new Icon(timer:Number,ip:String);
```

- instanziiert iCon
- timer in ms bestimmt wie häufig das Timer Event kommt
- ip ist die ip Nummer für das Zielsystem, wenn der Flash Player nicht lokal ist

Definiert ein Object, welches auf name\_ikon\_instanz "listened":

```
myObj = {};
```

verbinde myObj mit name\_ikon\_instanz:

- interpretiere Events mittels myObj.messages
- name\_ikon\_instanz.addEventListener("message", myObj.message)
- name\_ikon\_instanz.addEventListener("data", myObj.data)
- name\_ikon\_instanz.addEventListener("command", myObj.command);

name\_icon\_instanz.sendCommand(Kommando:String):  
testet myObj.command

**Methoden**

name\_icon\_instanz.cursor(status:Boolean):  
schaltet Cursor ein/aus

name\_icon\_instanz.sip(status:Boolean):  
schaltet sip ein/aus

name\_icon\_instanz.classtrace(status:Boolean):  
schaltet trace des Icon ein/aus

name\_icon\_instanz.version():String:  
liefert iCon Version und iCon Version Komponente  
liefert i2c Modul Version mit Kennung zurück: <x>Vi2c:xxx.xxx</x>

name\_icon\_instanz.discon():  
trennt iCon

name\_icon\_instanz.init():  
verbindet iCon

name\_icon\_instanz.check\_key(name:String,firma:String,key:String):  
führt Key Check durch...

name\_icon\_instanz.auth(key:String,sn:String,name:String,firma:String):  
führt Authentifizierung durch

name\_icon\_instanz.writefile(fileName:String,file:String):  
schreibt Text file in Textdatei fileName.txt in \Flashspeicherpfad\daten

name\_icon\_instanz.appendfile(fileName:String,file:String):  
hängt Text file an Textdatei fileName.txt in \Flashspeicherpfad\daten an.

name\_icon\_instanz.readfile(fileName:String):  
liest Textdatei fileName.txt aus \Flashspeicherpfad\daten.

name\_icon\_instanz.set\_volume(vol:Number):  
stellt Audio Lautstaerke ein 0-101

name\_icon\_instanz.get\_volume():  
gibt Lautstärke Wert in % mit Kennung zurück <A>ddd</A>

name\_icon\_instanz.audio(Boolean:status):  
schaltet Audio Ein oder Aus

name\_icon\_instanz.suspend():  
schaltet in den Suspend Modus

name\_icon\_instanz.set\_backlight(Number:lightness):  
lightness von 0 bis 100 steuert Hintergrundbeleuchtungshelligkeit

name\_icon\_instanz.get\_backlight():Number:  
holt lightness aus Registry  
Kennung "Hddd"

name\_icon\_instanz.setdate(year:Number,month:Number,day:Number):  
setzt System Datum UTF

name\_icon\_instanz.settime(hour:Number,minute:Number):  
setzt System Zeit UTF

name\_icon\_instanz.savedefault():  
speichert Parameter in default.ini

name\_icon\_instanz.get\_default():  
liest "default.ini".

name\_icon\_instanz.reset():  
führt Soft Reset des Rechners durch

name\_icon\_instanz.save\_registry():  
sichert Registry in Flashspeicher

name\_icon\_instanz.menu\_off():  
IE in den Kiosk Modus

name\_icon\_instanz.win\_come():  
IE Window Mode

name\_icon\_instanz.quitIE():  
Quit IE

name\_icon\_instanz.sendCom(data:String):  
sendet String über Com Schnittstelle

name\_icon\_instanz.emg(Command:String):  
Command = "Start" Initialisiert EMG Messung I2C  
Command = "Read" Gibt EMG Messwert mit Kennung <e>xxxx</e> zurück  
Command = "Stop" Stoppt Messung un FIR auf dem Modul  
Command = "InitRead" Gibt EMG Messwert mit Kennung <e>xxxx</e> zurück  
sperrt Funk Task, Initialisiert EMG, Holt Messwert und gibt Funk Task frei  
Command = "FunkStop" Stoppt Funktask

Command = "FunkStart" Gibt Funktask frei FunkNitProFitLady  
Command = "FunkNitProFitLady" Initialisiert Funkmodul für ProFitLady

name\_icon\_instanz.akkustatus():  
gibt Akkustatus mit Kennung zurück: <a>A,0,ddd</a> ddd = Akkustatus in %

name\_icon\_instanz.updatetest():  
fragt nach updates auf ftp server  
gibt Liste der upzudatenden Dateien aus

name\_icon\_instanz.update():  
dated Dateien Up und startet icon neu  
download.ini wird an daten/update kopiert  
Dateien werden gemäß download.ini vom FTP Server an daten/update kopiert  
iCon Server wird heruntergefahren  
update\_server wird gestartet  
update wird gemäß update.ini durchgeführt und update.log wird beschrieben  
update server wird heruntergefahren  
iCon server wird gestartet

name\_icon\_instanz.rnd(nt:Number,delay:Number):  
obsolet!  
nt = Anzahl der Random Numbers; delay = Zeit zwischen den Samples in Millisekunden (delay = 0 zur Zeit alle 80 ms ein Sample)  
gibt rnd aus 200 Trials zurück mit Kennung "4" ADC3 wird gemessen

name\_icon\_instanz.rnd\_kali():  
obsolet!  
geliefert wird <5>Kalibrierung gestartet</5> wenn Kalibrierung gestartet ist  
und <5>Kalibrierung OK</5> wenn Kalibrierung erfolgt ist

name\_icon\_instanz.get\_registration\_data():  
ruft Registrierungsdaten ab  
geliefert wird <k>r,3,software//biobedded,kunde,ssssssssss,</k> ssssssss = Inhalt des Keys zurück  
geliefert wird <k>r,3,software//biobedded,therapeut,ssssssssss,</k> ssssssss = Inhalt des Keys zurück  
geliefert wird <k>r,3,software//biobedded,id,ssssssssss,</k> ssssssss = Inhalt des Keys zurück  
geliefert wird <k>r,3,software//biobedded,registriert,ssssssssss,</k> ssssssss = Inhalt des Keys zurück  
oder "Reg Key NOK lesen!"

name\_icon\_instanz.registrate(kunde:String,id:String):  
es wird die Registrierung für den Kunden und die Geräte ID erzeugt  
geliefert wird <u>j,Create Registrierung FTP Dir "/kunde/id/" OK!</u> für OK  
geliefert wird <u>j,Create Registrierung FTP Dir "/kunde/id/" NOK!</u> falls Verzeichnis existiert

name\_icon\_instanz.keytoggle():  
toggelt Tastatur Sprachen Layout (dasselbe wie Shift Alt Tastenkombination)

name\_icon\_instanz.write\_reg\_key(key:String):  
key in registry schreiben, falls korrekt wird Authentifizierung durchgeführt

name\_icon\_instanz.wlanStatus():  
Fragt Wlan Signalqualität ab (abhängig von Hardware Version).  
Gibt <u>q,Q</u> zurück mit Q = (-100)-(0). 0 = beste Signalqualität.

name\_icon\_instanz.wlan\_reset():  
Zeigt Betriebssystem Fenster "Wlan". Hier kann der Anwender die Wlan Parameter setzen/bearbeiten.  
Gibt Kennung <u>r,Reset Wifi adapter OK </u> oder <u>r,Reser Wifi adapter not found </u> zurück

name\_icon\_instanz.ttlout(ttl:String):  
Gibt String an TTL Ausgänge aus (abhängig von Hardware Version).  
ttl - z.B. "11111111" schaltet alle Ausgänge auf On

name\_icon\_instanz.adc(Kanal:Number):  
Fragt Analog Digital Wandler ab.  
(Kanal 2 ist belegt mit Touch Screen Werten)  
Gibt Daten mit Kennung {0-3} zurück. Daten {0-512}.

name\_icon\_instanz.rnd\_xx(Anzahl:Number):  
Gibt Auftrag für Random Nummern heraus.  
Diese Funktion ist nur für das Rauschmodul Version 2.  
Anzahl - so viel werden pro Sekunde erzeugt.  
Liefert mit Kennung <C>inhalt</C>. Inhalt siehe (see Kommunikationsprotokoll Noisegenerator V1.1.pdf)

name\_icon\_instanz.start\_kalibrierung\_xx(Anzahl:Number):  
Gibt Auftrag für Kalibrierung heraus.  
Diese Funktion ist nur für das Rauschmodul Version 2.  
Anzahl - so viel werden für die Kalibrierung erzeugt.  
Liefert mit Kennung <C>inhalt</C>. Inhalt siehe (see Kommunikationsprotokoll Noisegenerator V1.1.pdf)

name\_icon\_instanz.get\_gauss\_array():  
Gibt Auftrag für Gauss Array heraus.  
Diese Funktion ist nur für das Rauschmodul Version 2.  
Liefert mit Kennung <C>inhalt</C>. Inhalt siehe (see Kommunikationsprotokoll Noisegenerator V1.1.pdf)

name\_icon\_instanz.netzwerk\_dialog():

Zeigt Betriebssystem Fenster "Netzwerk Dialog". Hier kann der Anwender die lokale IP bearbeiten.

name\_icon\_instanz.nt\_util\_dialog():

Zeigt Anwendungs Fenster "Netzwerk Tools". Hier kann der Anwender die verschiedene Netzwerkttest Programme aufrufen.

name\_icon\_instanz.emg\_delete():

löscht den Inhalt des Verzeichnisses /emg

name\_icon\_instanz.druck\_delete():

löscht den Inhalt des Verzeichnisses /druck

name\_icon\_instanz.training\_delete():

löscht den Inhalt des Verzeichnisses /training

name\_icon\_instanz.all\_delete():

löscht den Inhalt aller Verzeichnisse

name\_icon\_instanz.all\_copy():

kopiert den Inhalt aller Verzeichnisse auf die SD Card

name\_icon\_instanz.wlan\_connect():

Führt einen Konnekt des WLAN durch. Die Parameter werden aus den Voreinstellungen benutzt.

Gibt Kennung <u>c,Wifi connect! to: SSID OK </u> zurück.