

DATENBLATT

iCon® – Intelligent Connector Softwarelösung, die die Adobe® Flash Entwicklungsumgebungen Flex®, Flash® oder Open Source Flash Ent- wicklungsumgebungen (MTASC) für Embedded Systems nutzbar macht

Version 2.0 ist verfügbar auf: Exm32 Plattform von MSC (Mipsi), Ton-
funk© BMCU Mopad (Arm) und iPAN7 (Keith & Koep GmbH).

Über iCon© 2.0 kann die Embedded Hardware direkt über eine Klas-
senbibliothek angesprochen werden. Hierdurch werden die norma-
lerweise 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 Bedienungs Oberfläche.
- 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

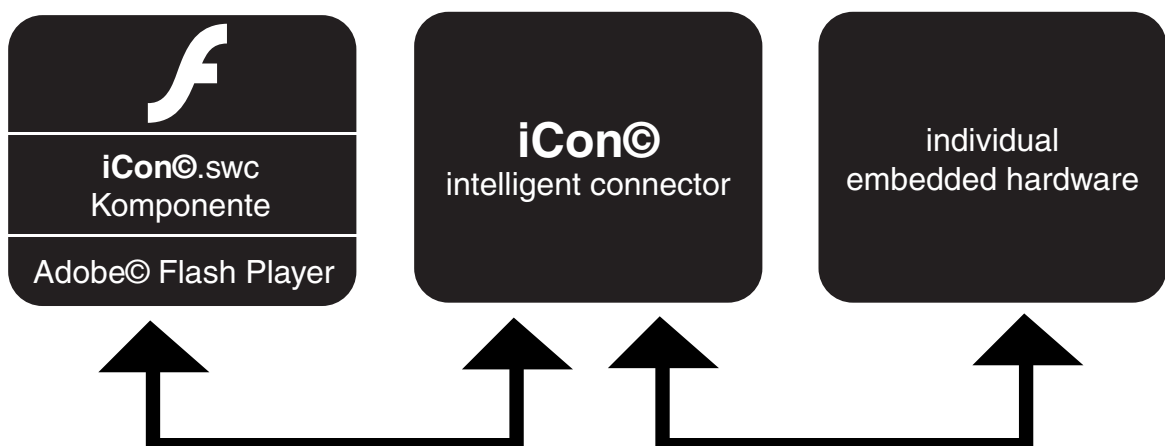
biobedded systems GmbH liefert iCon© 2.0 im Rahmen eines Lizenz-
abkommens nur an OEM Partner aus. Die Lizenz ist eine Bulk Lizenz
und wird einzeln vereinbart. Die Lösung wird im Rahmen eines Part-
nervertrages seitens des OEM auf Messen und zur Kundenpräsen-
tation vorgestellt. biobedded systems GmbH 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 Sicher-
heitsstandard her nur eine „Kernansprache“ von lokalen Ressourcen
anbieten. Die üblicherweise vom Betriebssystem mittels individueller
Treiber angesprochenen Hardware-Komponenten von OEM Embed-
ded 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 Systeme
„native“ in Adobe© Flash arbeiten.

Die Zusammenarbeit mit entsprechenden interessierten OEMs benö-
tigt 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.





FIRMENSITZ
Tersteegenstraße 15
D-46045 Oberhausen

GESCHÄFTSSITZ
Am Wiesenbusch 1
D-45966 Gladbeck

HANDELSREGISTER
HRB 3584
GmbH, AG Oberhausen

UMSATZSTEUER ID
DE209026417

GESCHÄFTSFÜHRER
Dipl.-Ing. Andreas Hilburg
Dipl.-Ing. Matthias Krzizan

KONTAKTDATEN
+49 2043 945 115
info@biobedded.de
www.biobedded.de

ZERTIFIZIERUNG
PN-EN ISO 13485

DATENBLATT

BEFEHLE

name_icon_instanz = new Icon(timer:Number,ip:String):	<ul style="list-style-type: none"> • 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_icon_instanz „listened“: myObj = {};
verbinde myObj mit name_icon_instanz:	<ul style="list-style-type: none"> • interpretiere Events mittels myObj.messages • name_icon_instanz.addEventListener(„message“, myObj.message) • name_icon_instanz.addEventListener(„data“, myObj.data) • name_icon_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 Lautstärke ein 0-101
name_icon_instanz.get_volume():	gibt Lautstärke Wert in % mit Kennung zurück <A>ddd
name_icon_instanz.audio(Boolean:status):	schaltet Audio ein/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

DATENBLATT

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 FunkInitProFitLady Command = „FunkInitProFitLady“ Initialisiert Funkmodul für ProFitLady
name_icon_instanz.akkuStatus():	gibt Akkustatus mit Kennung zurück: <a>A,0,ddd 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).



FIRMENSITZ
Tersteegenstraße 15
D-46045 Oberhausen

GESCHÄFTSSITZ
Am Wiesenbusch 1
D-45966 Gladbeck

HANDELSREGISTER
HRB 3584
GmbH, AG Oberhausen

UMSATZSTEUER ID
DE209026417

GESCHÄFTSFÜHRER
Dipl.-Ing. Andreas Hilburg
Dipl.-Ing. Matthias Krzizan

KONTAKTDATEN
+49 2043 945 115
info@biobedded.de
www.biobedded.de

ZERTIFIZIERUNG
PN-EN ISO 13485

DATENBLATT

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 Netzwerktest 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.