



Kybernetische Systeme im Wandel der Zeit

Vier Jahrzehnte Entwicklung, auch aus meiner subjektiven beruflichen Perspektive

Prof. Dr. -Ing. Vesselin lossifov
Professur für Mikrocomputertechnik
Hochschule für Technik und Wirtschaft (HTW) Berlin
FB1 Energie und Information
Vesselin.lossifov@htw-berlin.de | <http://www.f1.htw-berlin.de>



1. Technische Kybernetik – Trends und Charakterisierung

Definition (allgemein)

Die Technische Kybernetik ist eine interdisziplinäre Ingenieurwissenschaft an der Schnittstelle zu den System- und Naturwissenschaften, welche sich mit der Modellierung, Simulation, Regelung und Steuerung technischer Systeme befasst. Sie stellt ein Teilgebiet der Kybernetik dar und basiert größtenteils auf Methoden der mathematischen System- und Automatentheorie sowie der Regelungstechnik.

Quelle: https://de.wikipedia.org/wiki/Technische_Kybernetik



Definition (Gründerzeit)

Der Begriff der „Kybernetik“ wurde erstmals 1947 vom US-amerikanischen Mathematiker Norbert Wiener eingeführt. Wiener hatte erkannt, dass es im Bezug auf die Interaktion mit der Umwelt sowie auf die damit verbundene Regelung oftmals die gleichen Prinzipien sind, nach denen sowohl technische, soziologische oder wirtschaftliche Systeme als auch lebende Organismen funktionieren. Basierend auf dem griechischen Begriff **κυβερνήτης** („Steuermann“) prägte Wiener für diese Erkenntnis und die damit verbundene Wissenschaft die Bezeichnung „**Cybernetics**“. Der chinesische Wissenschaftler H. S. Tsien griff die Ideen Wiensers auf und führte 1954 im Rahmen seiner Arbeit in der Luft- und Raumfahrtforschung den Begriff „**Engineering Cybernetics**“ ein, auf welchem die heutige deutsche Bezeichnung "Technische Kybernetik" beruht.

Quelle: https://de.wikipedia.org/wiki/Technische_Kybernetik



Definition (heute)

Kybernetische Systeme, heute Cyber-Physical Systems, bilden die enge Verbindung eingebetteter Steuer- und Regelungssysteme zur Überwachung und Steuerung physikalischer Vorgänge mittels Sensoren und Aktoren über Kommunikationseinrichtungen mit den globalen digitalen Netzen, dem „Cyberspace“. Dieser Typus von Systemen ermöglicht über Wirkketten eine Verbindung zwischen Vorgängen der physischen Realität und den heute verfügbaren digitalen Netzinfrastrukturen. Dies erlaubt vielfältige Applikationen mit hohem wirtschaftlichen Potential und mit starker Innovationskraft.

Quelle: CYBER-PHYSICAL SYSTEMS - INNOVATION DURCH SOFTWARE- INTENSIVE EINGEBETTETE SYSTEME

http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/acatech_diskutiert/acatech_diskutiert_CPS_einseitig_01.pdf



2. Geschichte der Technischen Kybernetik an der TU (früher TH) Ilmenau



Campus TU Ilmenau

Bild 1 Campus TU Ilmenau, Quelle: <https://commons.wikimedia.org/w/index.php?curid=6870205>

Technische Hochschule Ilmenau, Sektion Technische und Biomedizinische Kybernetik.
Wissenschaftsbereich Informationsverarbeitung Katalog der Deutschen Nationalbibliothek.

Quelle: <https://portal.dnb.de/opac.htm?sessionid=EE433D03AFF981180ADE067FA132036.prod-worker0?method=simpleSearch&reset=true&cqlMode=true&query=ubRef%3D954549724&selectedCategory=any>



2.2 Die Gründerväter der Technischen Kybernetik an der TH Ilmenau



Bild 2 Quelle: <https://www.tu-ilmenau.de/it-tet/informationen/mitarbeiter/philippow/>, https://de.wikipedia.org/wiki/Eugen_Philippow

Eugen S. Philippow (* 9. Mai 1917 in Sofia; † 2. September 1991 in Ilmenau) war ein bulgarischer Ingenieur und Hochschullehrer. Er trug maßgeblich zum Aufbau und zur Entwicklung von Lehre und Forschung des Fachgebiets Theoretische Elektrotechnik an der heutigen TU Ilmenau bei.

Taschenbuch **Elektrotechnik** in sechs Bänder:

Quelle: <http://www.zvab.com/buch-suchen/titel/taschenbuch-elektrotechnik-in-sechs/autor/eugen-philippow/>



Bild 3 Quelle: https://de.wikipedia.org/wiki/Karl_Reinisch

Karl Reinisch (* 21. August 1921 in Dresden; † 24. Januar 2007 in Ilmenau) war ein deutscher Elektroingenieur und Professor für Regelungstechnik in Ilmenau. Unter seiner Leitung wurden tragfähige Grundlagen für die Automatisierungs- und Systemtechnik als Bestandteile der Technischen Kybernetik an der Technischen Universität Ilmenau geschaffen. Auch war er langjährig bei der internationalen Föderation für Wissenschaft und Technik von Automatisierung und Regelung, der IFAC aktiv. Student von **Heinrich Barkhausen** und Georg Mierdel.

Kybernetische Grundlagen und Beschreibung kontinuierlicher Systeme, Karl Reinisch.

Quelle: <http://d-nb.info/750088915>

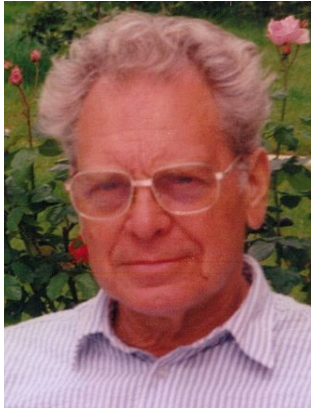


Bild 4 Quelle: https://de.wikipedia.org/wiki/Hans-Joachim_Zander

Hans-Joachim Zander (* 27. August 1933 in Memel, heute Klaipeda, Litauen) ist ein deutscher Ingenieur und Hochschullehrer. **Er gehört zu den Pionieren der Steuerungstechnik.** Die Arbeitsschwerpunkte in den 1970er Jahren waren Entwicklungen zu einem umfassenden Programmsystem für den rechnergestützten Entwurf digitaler Steuerungen und wurden auf Robotron-Großrechner implementiert. Mit dem Aufkommen der Prozessor- und SPS-Technik zeichnete sich der Wandel von der verbindungsprogrammierten Steuerung (VPS) hin zur speicherprogrammierbaren Steuerung (SPS). Das Interesse am rechnergestützten Entwurf digitaler Steuerungen wandelte zum Schaltkreisentwurf der Mikroelektronik.



2.3 Studium der Technischen Kybernetik an der TU Ilmenau, BA/MA Technische Kybernetik und Systemtheorie

Das Studium vermittelt Grundlagen aus den Bereichen der Automatisierung, Modellbildung, Analyse, Systemtheorie, Regelung und Optimierung von technischen Prozessen. Eine mathematische Grundausbildung, die sich aus dem Bachelorstudiengang der Mathematik ableitet, und eine breitgefächerte Auswahl von Grundlagen der ingenieur- und naturwissenschaftlichen Disziplinen legen hierzu den Grundstock.

Quelle: <https://www.tu-ilmenau.de/studieninteressierte/studieren/bachelor/technische-kybernetik-und-systemtheorie/>



3. Kybernetische Systeme heute

Ein **Cyber-physisches System**, engl. „**Cyber-physical systems**“, bezeichnet den Verbund informatischer, softwaretechnischer Komponenten mit mechanischen und elektronischen Teilen, die über eine Dateninfrastruktur, wie z.B. das Internet, kommunizieren. Die Ausbildung von cyber-physischen Systemen entsteht aus der Vernetzung eingebetteter Systeme durch drahtgebundene oder drahtlose Kommunikationsnetze. Die Begriffsbildung folgt dem Bedarf an einer neuen theoretischen Grundlage für die Erforschung und Entwicklung großer, verteilter, komplexer Systeme, wie zum Beispiel der Weiterentwicklung des deutschlandweiten Stromnetzes, hin zu einem intelligenten Stromnetz, oder die Konstruktion neuartiger Industrieproduktionsanlagen, die sich hochdynamisch an die jeweiligen Produktionserfordernisse anpassen können.

Quelle: https://de.wikipedia.org/wiki/Cyber-physisches_System



Cyber-physische Systeme stehen für die Verbindung von physikalischer und informationstechnischer Welt

- von eingebetteten Systemen, Anwendungssystemen und Infrastrukturen – Steuerungen im Fahrzeug, intelligente Kreuzungen, Verkehrsmanagementsysteme, Kommunikationsnetze und ihre Verknüpfungen mit dem Internet,
- auf Basis ihrer Vernetzung und Integration,
- und der Mensch-Technik-Interaktion in Anwendungsprozessen.

Der Begriff "Cyber-physische Systeme" wird definiert in:

- VDI GMA Fachausschuss 7.20 „Cyber Physical Systems“
<http://www.vdi.de/technik/fachthemen/mess-und-automatisierungstechnik/fachbereiche/anwendungsfelder-der-automation/gma-fa-720-cyber-physical-systems/>
- Cyber-physische Systeme und Industrie 4.0
<http://i40.iosb.fraunhofer.de/FA7.21%Begriffe%20-%20Industrie%204.0#cyber-physical-system>



Abgeleitet aus folgenden Quellen, die hier oft zitiert werden:

- **acatech POSITION Cyber-Physical Systems.** Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion.
http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Cyber-Physical-Systems/acatech_POSITION_CPS_web.pdf
- **Cyber-Physical Systems** – Innovation durch Software.Intensive Eingebettete Systeme.
http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/acatech_diskutiert/acatech_diskutiert_CPS_einseitig_01.pdf
- **VDI/VDE GMA Thesen** und Handlungsfelder - **Cyber-Physical Systems:** Chancen und Nutzen aus Sicht der Automation, April 2013 „Integrierte Forschungsagenda Cyber-Physical Systems“, Acatech 2012.
https://www.vdi.de/uploads/media/Stellungnahme_Cyber-Physical_Systems.pdf
- Broy 2010, Cyber-Physical Systems – Wissenschaftliche Herausforderungen bei der Entwicklung.
- Rainer Drath: Industrie 4.0 – Cyber Physical Production Systems. The next industrial revolution? Keynote at ETFA 2013, Cagliari, Italy, 2013.



3.1 Charakterisierung von Cyber-physischen Systemen

Die Informations- und Kommunikationstechnologie (IKT) sticht seit ihrem Bestehen durch eine anhaltende Folge schneller technologischer Fortschritte hervor. Immer stärker miniaturisierte integrierte Schaltungen sowie das anhaltend rapide Wachstum von Rechenleistung und Bandbreite in Netzwerken sind nur die am stärksten ins Auge fallenden Phänomene. IT ist allgegenwärtig, die Vision vom **Ubiquitous Computing** ist Wirklichkeit geworden.

Im Zuge dieses Trends entstehen offene, vernetzte, flexibel agierende und interaktive Systeme, die die physikalische Welt nahtlos mit der virtuellen Welt der Informationstechnik verknüpfen. Softwareintensive Systeme und Geräte wandeln sich durch ihre vielfältige Vernetzung und die Einbindung von Daten und Diensten aus globalen Netzen zunehmend zu integrierten, übergreifenden Diensten und Lösungen, die in allen Lebensbereichen genutzt werden.

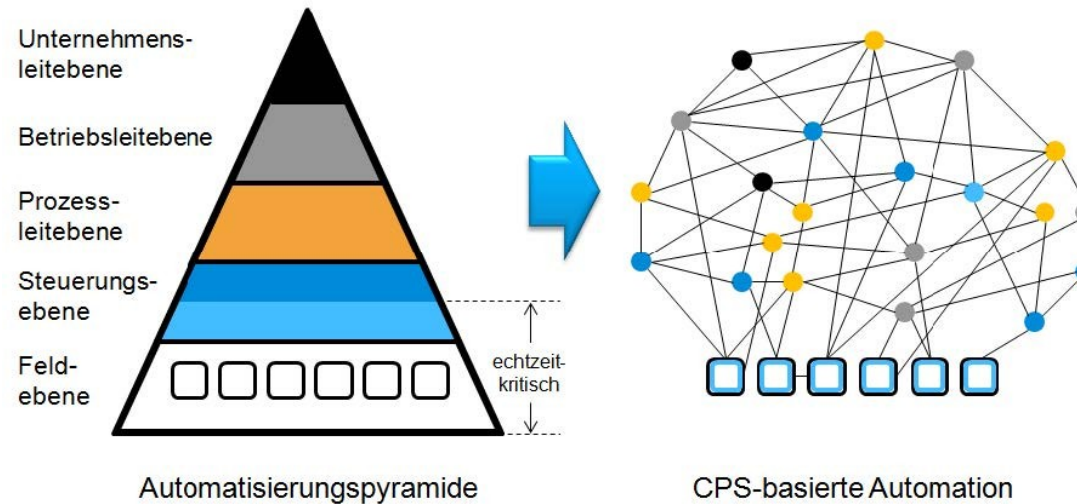


Bild 5 Auflösung der hierarchischen Automatisierungspyramide durch CPS mit verteilten Diensten

Quelle: https://www.vdi.de/uploads/media/Stellungnahme_Cyber-Physical_Systems.pdf



3.2 Ineinandergreifende Anwendungsbereiche und verwandte Disziplinen

Aufgrund ihres systemwissenschaftlichen Ansatzes und der interdisziplinären Ausrichtung finden Kybernetische Systeme, heute Cyber-Physical Systems, in vielen Bereichen Anwendung, u.a. in der

- Kraftfahrzeugtechnik (u.a. Antiblockiersystem, Fahrdynamikregelung, autonomes Fahren)
- Luft- und Raumfahrttechnik
- Automatisierungstechnik
- Verfahrenstechnik
- Informatik und Nachrichtentechnik
- Elektrotechnik
- Energietechnik und Energieversorgung
- Robotik
- Verkehrstelematik, insbesondere Verkehrsmanagement
- Produktionstechnik
- Umwelttechnik
- Medizintechnik



4. Technologien zur Realisierung der besonderen Fähigkeiten von Cyber-physischen Systemen

Zur Realisierung von Cyber-physischen Systemen bedarf es einer Fülle von Technologien und Engineering-Verfahren. Betrachtet wird hier die Technologie des "Low-Power" von-Neumann Betrieb und Programmierung von Embedded Systems, die spezifisch zur Realisierung der besonderen Fähigkeiten und Eigenschaften von Cyber-physischen Systemen sowie ihrer Erforschung beiträgt.

4.1 Sensor- und Aktortechnologie

Sensoren und Aktoren dienen der Beobachtung und Beeinflussung der physikalischen Umwelt von Cyber-physischen Systemen. Sensoren nehmen physikalische oder chemische Eigenschaften wie Temperatur, Feuchtigkeit, Schall oder die stoffliche Beschaffenheit der Umgebung qualitativ oder quantitativ auf und wandeln die so erhaltenen Messwerte in eine digital weiterverarbeitbare Form um. Aktoren setzen digitale Stellwerte in physikalische Größen wie Druck, Temperatur oder mechanische Bewegung um und beeinflussen so die Umgebung.



4.2 Verteilte stabile Regelungen

Verteilte Regelungen sind Regelkreise oder -netze, bei denen die signalverarbeitenden Komponenten nicht zentral, sondern geographisch verteilt und gegebenenfalls sogar hierarchisch angeordnet sind. Hierdurch erlauben sie die Beherrschung komplexerer verteilter Systeme, für die zentrale Regelung nicht möglich ist, und insbesondere die für Cyber-Physical Systems notwendige Beeinflussung der Umwelt, abhängig von Sensorwerten durch Interaktion verteilter Teilsysteme.

4.3 Leistungssteigerungen durch neue Technologien (Auszug):

- Zusätzliche **Hardwarefunktionen für Spezialaufgaben** (Multimedia, Grafik, Videoanalyse, Bildverarbeitung, Echtzeit, IT-Sicherheit) führen zu Chips mit dedizierten Kernen für spezielle Aufgaben.
- Eingebettete Systeme ermöglichen noch stärker verteilte, eingebettete Steuerungen und Regelungen auf **geringen Stromverbrauch** durch **energieeffiziente Programmierung** und Kommunikationsprotokolle zu optimieren.
- Neue Methoden der Energiegewinnung (**Energy Harvesting**) erlauben CPS ohne externe Energiezufuhr.



5. Grundsätze, Möglichkeiten und neue Aufgaben bei der HW/SW-Entwicklung mit "Low-Power" von-Neumann Prozessorbetrieb und -Programmierung

Embedded Systems, Funkmodule, Sensoren etc. müssen für einen anwendungsgebundenen, minimalen oder gar un stetigen und schwachen Energiefluss (Energy Harvesting), der aber ihre Funktionalität dennoch sichern muss, ausgelegt werden.

Die Low-Power Fähigkeit einer Systemlösung wird bestimmt durch geeignete Betriebsarten für die Umsetzung von funktionalen Energiesparzuständen des Systems, Auswahl und/oder Entwicklung neuartiger Algorithmen zur Datenanalyse und Systemsteuerung mit möglichst geringem Bedarf an Ressourcen (Hardware/Software) und Anwendung neuartiger Konzepte zur Energiegewinnung. Der Softwareentwurf für Embedded Systems mit limitierten Energieressourcen soll somit auf die Randbedingung einer **doppelten Echtzeitfähigkeit** des Gesamtsystems ausgerichtet werden.



5.1 Wesentliche zu berücksichtigende Punkte

- **Energiemanagement** - Es ist wichtig den Energiehaushalt des Embedded Systems, Funkmodule, Sensoren/Aktoren etc. auch zeitlich zu analysieren.
- **Low-Power Embedded Systems** mit einer limitierten Energiezufuhr haben eine limitierte Funktionalität als Funktion der Zeit, bestimmt durch die Umstände, dass alle Aufgaben in der Zeit der Energiepräsenz erledigt werden müssen. Die Herausforderung hier ist die dynamische Energiezuweisung. Es muss festgestellt werden, welche Computerressourcen zu welchem Zeitpunkt notwendig sind, um eine korrekte Funktionalität aus der Sicht der Anwendung zu gewährleisten und wie das System fehlertolerant und re-entrant unter schwankendem Energiefluss gestaltet werden kann.
- **Der Softwareentwurf für Embedded Systems** mit limitierten Energieressourcen muss auf der Randbedingung einer doppelten Echtzeitfähigkeit des Gesamtsystems ausgerichtet sein – **Echtzeitfähigkeit** bezogen auf die Anwendung und **Echtzeitfähigkeit** bezogen auf die zeitliche Verfügbarkeit der Energieversorgung.



Die zeitliche und algorithmische Analyse und Simulation der beiden Randbedingungen stellt hier die Herausforderung dar.

5.2 Zwei Lösungsansätze (wahlweise oder zusammen)

- **statische**, auch semiautomatisierte, semantikbasierte und durch Codeanalyse und Kenntnis der Low-Power Eigenschaften des Embedded Systems prädestinierte Lösung.
- **dynamische** Energiezuweisung den einzelnen Funktionsgruppen des Embedded Systems gesteuert von einem Betriebssystem.

An dieser Stelle freut es mich besonders festzustellen, dass der wissenschaftliche Schwerpunkt meiner Dissertation, „**Methoden der dynamischen Priorisierung von Prozessen in Echtzeit-Systemen**“, verteidigt vor **34 Jahren im März 1982**, hier wieder die wissenschaftliche Herausforderung darstellt. Diesmal soll statisch und/oder dynamisch in Echtzeit entschieden und unter Energy-Harvesting Bedingungen realisiert werden, welchem Prozess energiegespeiste Ressourcen und wie lange zur Verfügung gestellt werden sollen, so dass die Echtzeitfähigkeit gewährleistet wird (doppelte Echtzeitfähigkeit). Die aktuelle Forschung auf diesem Gebiet sucht und findet immer mehr Antworten auf dieser Herausforderung.



6. Low-Power Codeanalyse - ULP Advisor für MSP430 MC

Energieminimale Programmiermodelle und Techniken brauchen **Low-Power Profiling** Werkzeuge wie der **ULP Advisor von Texas Instruments** im **Code Composer Studio** für **MSP430** zum Einsatz, http://processors.wiki.ti.com/index.php/ULP_Advisor. Dadurch wird schon in der Entwicklungsphase möglich der Anwendung den Energieverbrauch des Systems abschätzen zu können (simuliertes Profiling). Durch die direkte Schnittstelle zu einem Messplatz kann der reale Verbrauch gegenüber dem simulierten verifiziert werden. So können unerwünschte „Verbraucher“ (z.B. Ruheströme, unbenutzte schwingende Eingänge, gegeneinander treibende Ausgänge) eines Systems identifiziert werden.

Der **Ultra-Low Power Advisor ist ein Werkzeug zur C-Code Analyse** bezüglich einer optimalen Nutzung der ultra-low power Möglichkeiten des MSP430 MC. Das Ziel ist, einen Code zu entwickeln, mit dem der anwendungsspezifische Energieverbrauch der Funktionsgruppen des MSP430 um jedes mögliche Nanoampere Strom reduziert werden kann. Zur Übersetzungszeit werden Codezeilen hervorgehoben, die nach eingehenden Empfehlungen des ULP bezüglich dem o.g. Ziel weiter optimiert werden können.



6.1 Einige Regeln zur Low-Power Programmierung als Empfehlungen vom ULP MSP430

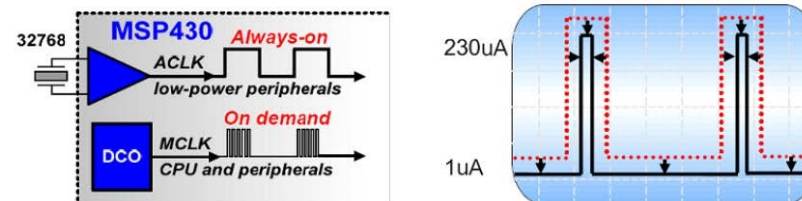
Rule 1.1 Nutzung des Low power Modus (LPM)

Die Nutzung des Low Power Modus (LPM) des Mikrocontrollers ist unbedingt zu empfehlen. Ein der wichtigsten Schritte für einen geringen Energieverbrauch besteht in der Minimierung der Zeit des Aktiven Modus des Mikrocontrollers und zur Maximierung der Zeit im Low Power Modus. E/A Funktionsgruppen können so konfiguriert werden, dass sie ohne die Intervention der CPU arbeiten können. Die CPU sollte nur kurz "geweckt" werden, um kritische Tasks auszuführen um anschließend schnell wieder in das Low Power Modus versetzt zu werden.

```
void main (void) {  
    while(1)  
    {  
        _bis_SR_register(LPM3_bits + GIE); // LPM0 mit Interrupts aktiv  
    }  
}
```

Low-Power Operation

- ◆ Power-efficient MSP430 apps:
 - ◆ Minimize instantaneous current draw
 - ◆ Maximize time spent in low power modes
- ◆ The MSP430 is inherently low-power, but your design has a big impact on power efficiency
- ◆ Proper low-power design techniques make the difference



“Instant on” clock

Operation ...
44

FLL + DCO statt PLL: die Innovation für Low Power
Phase Locked Loop, Frequency Locked Loop, Digital Controlled Oscillator
 $f_{\text{System}} = N \times f_{\text{Quarz}}$

Bild 6 Low-Power Operation MSP 430, Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad



Low-Power Modes

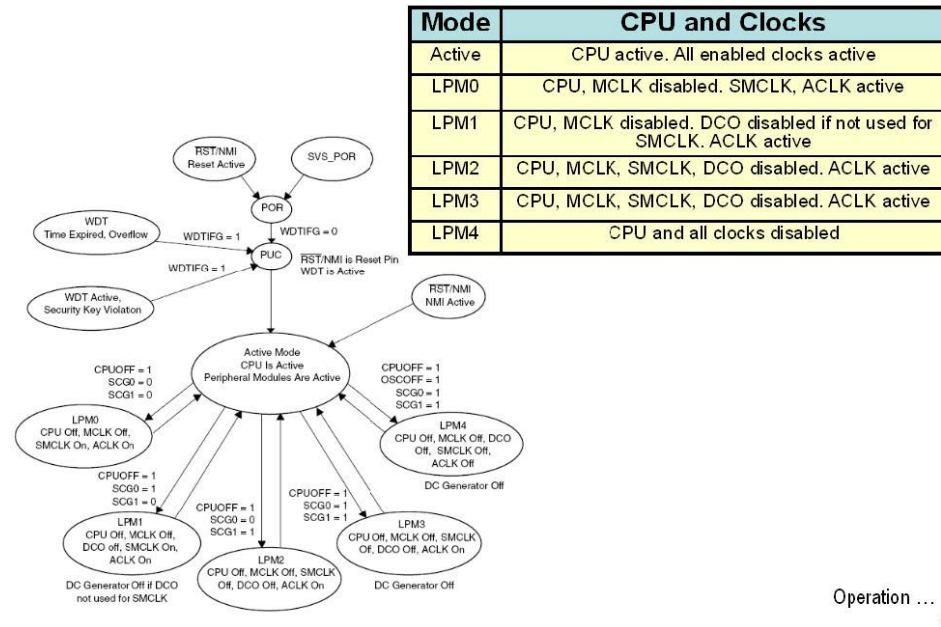


Bild 7 Low-Power Modi MSP 430, Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad



Rule 3.1 Nutzung von Interrupt-Service Routinen anstatt Flag Polling

Anstatt die CPU in eine Warteschleife verweilen zu lassen, um periodisch ein Statusbit in einem Algorithmus zu prüfen, sollte die CPU in „**Low Power Modus**“ (LPM) versetzt werden, um per Interrupt das erwartete Ereignis anzuzeigen, kurz mit einer Interrupt-Service Routine im „**Aktiv-Modus**“ zu verarbeiten und wieder in den LPM versetzt zu werden.

```
// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10 (void)
{
    ADC10CTL0 &= ~ADC10IFG;           // Löschen Interrupt Flag
    ADC10CTL0 &= ~ENC;                // Deaktivieren der ADC Umwandlung
    ADC10CTL0 &= ~(REFON + ADC10ON); // Auswahl und ADC10 deaktivieren
    tempRaw = ADC10MEM;              // Lesen des umgewandelten Wertes
    __bic_SR_register_on_exit(LPM3_bits);
}
```



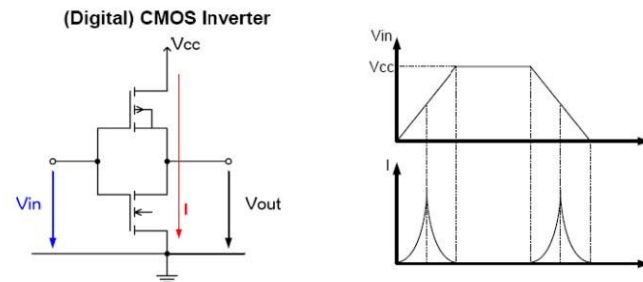
Rule 4.1 Abstützen von ungenutzten GPIO Pins gegen Vcc/Gcc mit einem Pull-up Widerstand

Ungenutzte GPIO Pins führen zu Leckverlusten der Transistoren und zur allgemeinen Erhöhung des Stromverbrauchs in der Baugruppe. Programmtechnisch müssen die Ausgänge so definiert oder gegen Betriebsspannung und/oder Masse gestützt werden, dass diese Leckverluste vermieden werden.

```
P1DIR = 0xFF; //0xFF bedeutet Programmierung auf "Ausgabe"  
P1OUT = 0x00; //bedeutet "LOW"-Spannungspegel, bei nicht abgestützten GPIO Pins  
P1OUT = 0xFF; //bedeutet "High"-Spannungspegel, bei abgestützten GPIO Pins
```

Unused Pin Termination

- ◆ Digital input pins subject to shoot-through current
 - ◆ Input voltages between V_{IL} and V_{IH} cause shoot-through if input is allowed to “float” (left unconnected)
- ◆ Port I/Os should
 - ◆ Driven as outputs
 - ◆ Be driven to V_{CC} or ground by an external device
 - ◆ Have a pull-up/down resistor



Lab...
48

Bild 8 Ungenutzte GPIO Pins, Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad



Rule 5.1 Rechen- und somit energieintensive Operationen vermeiden: Modulo, Dividieren, Fließkomma

Modulo- und Division-Operationen außerhalb von Schleifen als Kehrwert-Produkte berechnen lassen und diese dann in der Schleife als Multiplikation nutzen. Fließkomma-Operationen im 32-Bit IEEE754 Standard mit Festkomma-Bibliotheken ersetzen. Genauigkeit der Anwendung anpassen.

Rule 5.1 Code Example

```
int i, x = 20000, y = 3, y_KW, z = 0;
y_KW = 1/y;                               // Bildung des Kehrwertes von "y"
for (i = 5000; i>0; i--)                   // Schleifenvariable runterzählen
{
    /* application code */
    z+ = x * y_KW;
    // In instead of: z = x/y;
}
```



Rule 13.1 Runterzählen von Schleifenvariablen

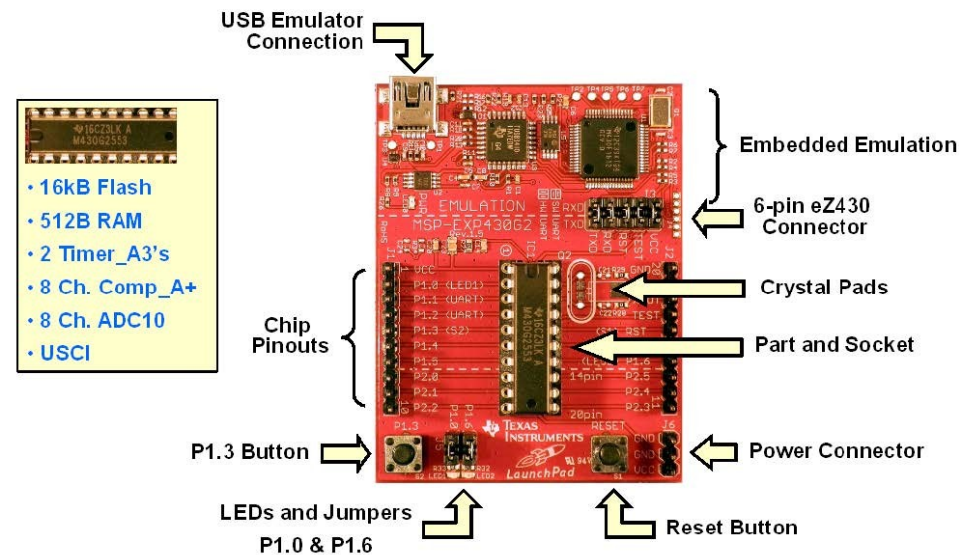
Im MSP430 Assemblercode basiert der Conditional Branch auf dem Vergleich einer Variablen/Register mit einem Wert unterschiedlich Null auf zwei Befehle: Compare und Conditional Branch. Branching & comparing bezogen auf einem Vergleich mit Null lässt sich mit nur einer spezifischen Operation, BNE realisieren. So kann mit einer runterzählenden Schleifenvariablen ein Befehl pro Iteration sparen gegenüber einer hochzählenden Schleifenvariable. Und somit auch Zeit (**5000* 62,5 nS**) und Energie.

Rule 13.1 Code Example

```
int i;
P1OUT |= 0x01;           // Set P1.0 LED on
for (i = 5000; i>0; i--) // Count down loop
// In instead of: (i = 0; i <5000; i++)
{
    /* Execute your application code */
}
```

6.2. Low-Power Code Optimierung: Ein Beispiel mit der ADC-Baugruppe des MSP430 Mikrocontrollers

LaunchPad Development Board



Lab ...
10

Bild 9 Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad

Mit Hilfe des ADC werden die Signalwerte eines Temperatursensors digitalisiert. Dieser Prozess kann mit der CPU im „**Active mode**“, also **klassisch**, in einer WHILE-Schleife des ADC oder **Low-Power** mit Interrupt-Triggerung des ADC, Timer und DMA im „**Low-Power-Mode**“ der CPU realisiert werden.

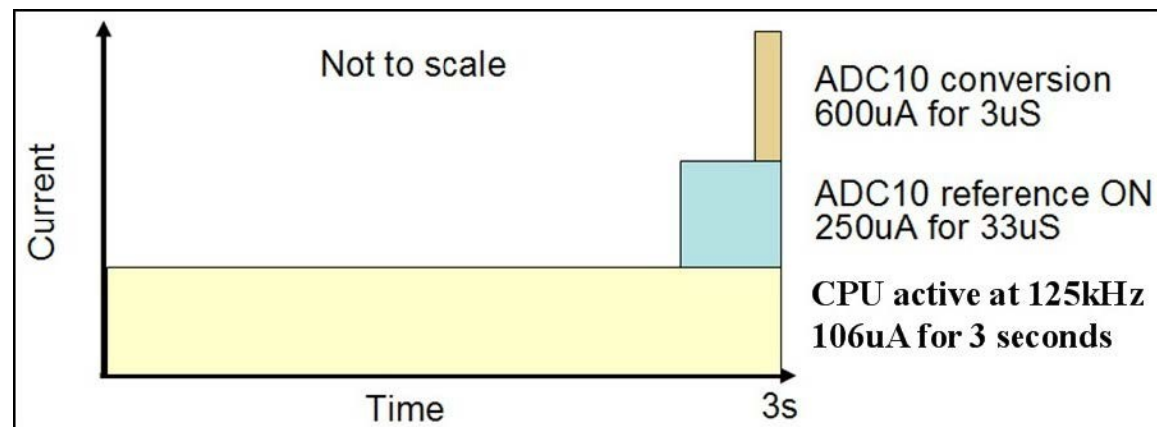


Bild 10 Stromverbrauch des MSP430 LaunchPad mit ADC-Einsatz im „Active mode“

Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad

Der Stromverbrauch des MSP430 beim Einsatz des ADC mit der CPU im „**Active mode**“, also klassisch, in einer WHILE-Schleife des ADC aus dem Programmbeispiel, hat etwa einen Verlauf wie in Bild 10. Von der CPU im „Active mode“ werden über die ganze Zeit der ADC Aktivität **106 μA** Strom konsumiert. Die ADC allein konsumiert in der aktiven Phase von **33 μs** für jedes Sampling **250 μA** . Der Strom für das eigentliche Sampling der ADC ist **600 μA** für **3 μs** .

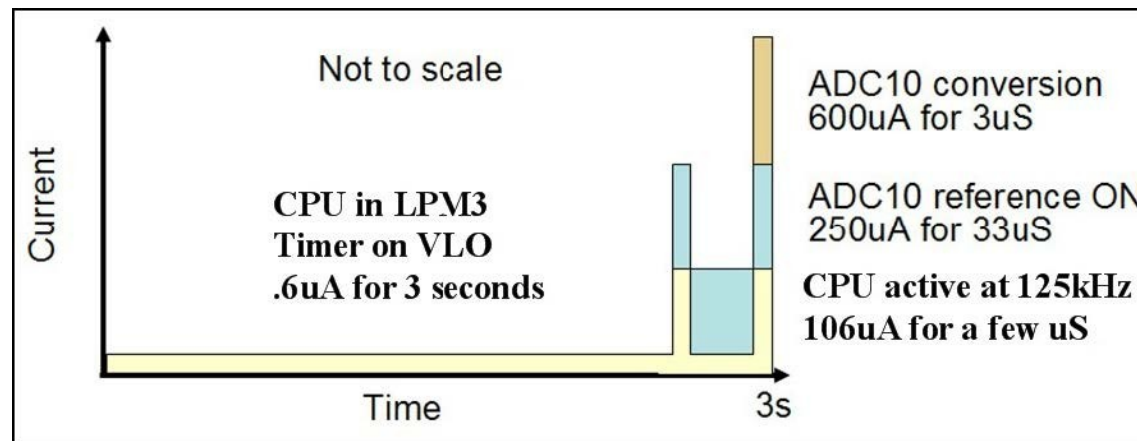


Bild 11 Stromverbrauch des MSP430 LaunchPad mit ADC-Einsatz im „LPM“

Quelle: Texas Instruments, Getting Started with the MSP430 LaunchPad



Im weiterentwickelten Beispiel wird die Zeit der CPU im „**Active mode**“ mit Hilfe des Ultra-Low Power Advisors MSP 430 reduziert. Das Sampling des ADC durch eine **WHILE-Schleife** wird von einem CPU **Low-Power-Mode** und **Interrupt-Service-Routinen** orientierten Sampling ersetzt. Das Ergebnis des Stromverbrauchs ist auf Bild 11 zu sehen. Der Stromverbrauch der CPU im Low-Power-Mode ist jetzt **0.6 μA , also ca. $1.8 \cdot 10^2$ mal geringer!** Das ist ein großer Unterschied! Die CPU verweilt die meiste Zeit der Samplingperiode in Low-Power-Mode mit einem Stromverbrauch von **600nA** anstatt **106 μA** . In einer anwendungsorientierten Architektur des Mikrocontrollers sind auch weitaus geringere Stromwerte mit einer dem Anwendungsfall angepassten I/O-Baugruppe erreichbar. Die Batterielaufzeit könnte von Monaten auf Jahren verlängert oder Energy Harvesting erst möglich gemacht werden..

Es lohnt sich also, Low-Power Programmierung als Anwender zu beherrschen und die HW des ES dieser Zielstellung anzupassen. Eine neue Möglichkeit und Herausforderung in der neuen CPS-Welt.



Schlussbemerkungen

- Kybernetische Systeme auf der Grundlage von **energieminimalen Embedded Systems, Energy Harvesting** mit **energieminimalen Funkprotokollen** und Techniken der Datengewinnung daraus können heute als **Ubiquitous Computing** Wirklichkeit werden.
- Beim Einsatz solcher Systemen ist der **Technische Kybernetiker** heute gefordert, neben der Beherrschung von Prozessmess-, Regel- und Steueralgorithmen die analoge und digitale Schaltungstechnik, die Funkprotokolle, die Rechnerarchitektur von Embedded Systems und ihre low-power Programmierung zu beherrschen.
- Die Forschung liefert immer neue Herausforderungen, etwa bei Methoden der Informationsgewinnung und -einspeisung aus/in einer „**Wolke**“, **Wireless Sensor Networks**, von Embedded Systems/Energy Harvesting Knoten.

Wie sollte der Technische Kybernetiker das schaffen?



„Der Computer ist das bemerkenswerteste Werkzeug, das wir je bekommen haben. Er ist so wie ein Fahrrad für unseren Geist.“

Steve Jobs (1955-2011)

Lasset unseren Geist auf diesem „Fahrrad“ schwingen und mit ihm in die Zukunft der energieminimalen und -nachhaltigen kybernetischen Systemen radeln!



VIELEN DANK!