



Ein Hardwareüberblick zum ATmega 2560

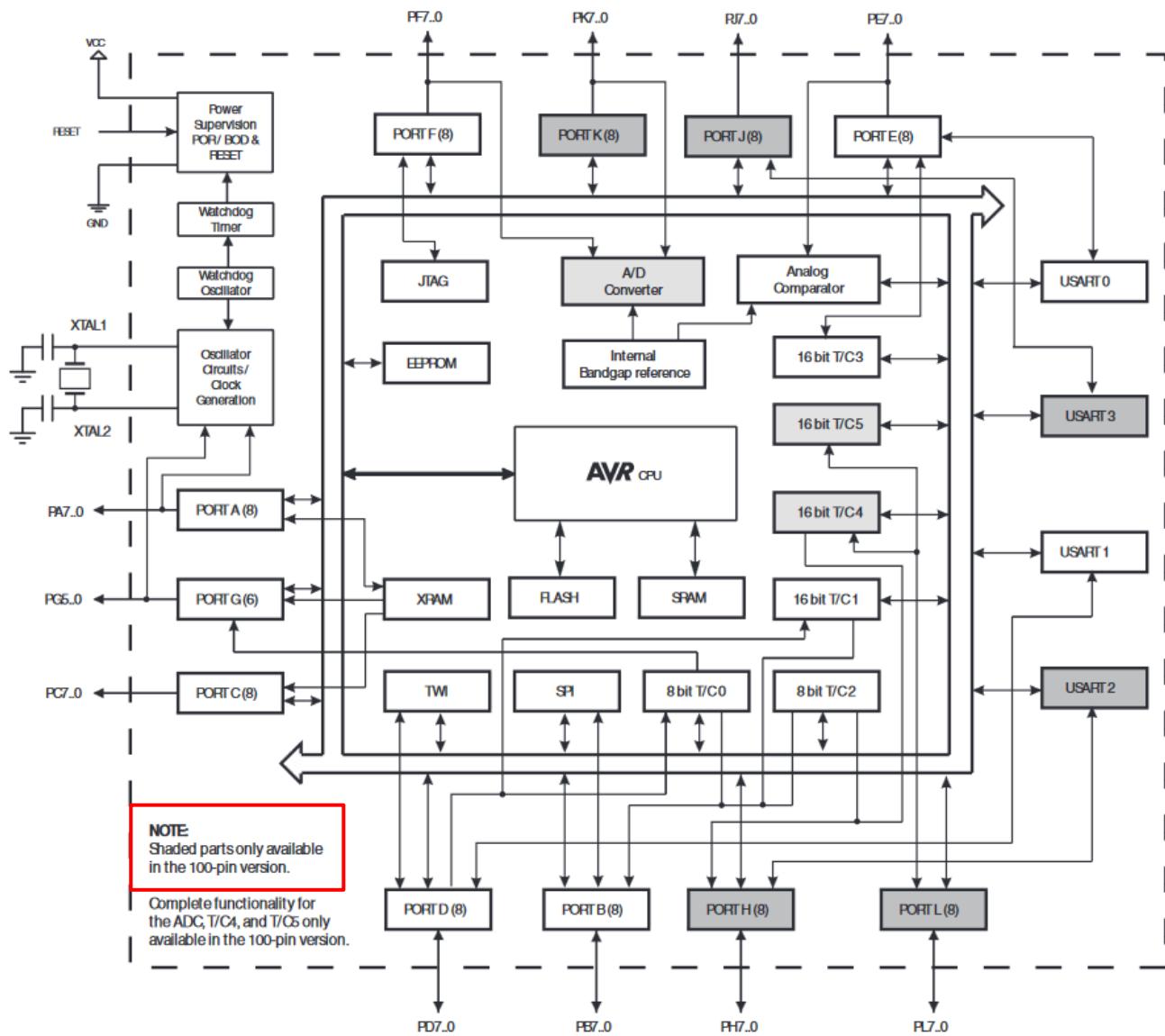
In diesem Arbeitsblatt werden Sie sich etwas näher mit der Hardware des verwendeten Mikrocontrollers ATmega 2560 beschäftigen. Die Angaben und Zitate sind -wenn nicht anders angegeben- nachfolgend genannten PDFs entnommen, welche Sie so im Internet finden können^{1 2}:

[Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_Summary](http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_Summary.pdf)

[Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet](http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf)

„The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhancedRISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the systemdesigner to optimize power consumption versus processing speed.“

Blockschaltbild des ATmega 2560



1 http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_Summary.pdf abgerufen am 1.4.2019

2 [https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf) abgerufen am 1.4.2019

Arbeitsblatt Nr. 05	Q2 Technikwissenschaft: Digitale Steuerungstechnik		B S G G
Datum:	Thema: Hardwareüberblick		
Seite 2 von 4	Name:		

Das auf den ersten Blick möglicherweise etwas unübersichtliche Blockschaltbild stellt die Vielfalt der möglichen Funktionalitäten dar. Bei dem verwendeten Modell stehen auch die grau schattierten Funktionsblöcke zur Verfügung. Schauen wir uns die Hardwareausstattung nun etwas näher an. Lassen Sie sich darauf ein, so schwer wie es aussieht, ist es nicht ;-)

Zuerst: Das gestrichelte Rechteck bildet quasi die Trennung zwischen dem Innenleben des Mikrocontrollers und der Außenwelt. Extern sind daher nur wenige Komponenten erforderlich.

Links oben erkennt man die extern zuführende Spannungsversorgung und die Möglichkeit für einen Reset des Mikrocontrollers (auf dem Funduino-Board der Taster).

Links in der Mitte ist der Quarzoszillator abgebildet, der den Systemtakt liefert. Bei dem verwendeten Funduino-Board bzw. dem enthaltenen Mikrocontroller beträgt dessen Frequenz 16 MHz. Und das war es auch schon mit den erforderlichen Komponenten!

Schaut man sich das Funduino-Board etwas genauer an, entdeckt man noch einen weiteren kleinen Chip: Das ist ein zweiter kleiner Mikrocontroller namens ATmega 16U2³, der die Anbindung des Funduino-Boards an einen USB-Anschluss kontrolliert und das Flashen des ATmega 2560 ermöglicht. Ansonsten sind auch noch einige Bauteile hinsichtlich der Stabilisierung der Spannungsversorgung per Klinkenbuchse enthalten.

Einen PDF-Schaltplan für das komplette Arduino MEGA 2560 Board findet sich unter der Webadresse https://www.arduino.cc/en/uploads/Main/arduino-mega2560_R3-schematic.pdf.

Okay, zurück zum ATmega 2560...

Digitale I/O-Pins

Was bisher schon ansatzweise genutzt wurde sind die digitalen Ports. Der ATmega 2560 verfügt über 11 Ports (von Port A bis Port L, wobei es keinen Port I gibt) mit insgesamt 86 digitalen I/O-Pins. Wer jetzt stutzt: Außer Port G sind alle Ports 8-Bit breit und bidirektional; d.h. sie können sowohl als Ein- wie auch als Ausgang verwendet werden. Port G hat nur sechs Bits.

Jeder Port ist mit dem internen Bussystem (symbolisiert durch das Rechteck mit den beiden Pfeilspitzen) verbunden. Allerdings hat jeder einzelne Port noch besondere Funktionalitäten, die man im ausführlichen Datenblatt beschrieben findet.

Serielle Schnittstellen

Weiterhin enthält der ATmega 2560 vier serielle Schnittstellen (USART0 bis USART3) zur Kommunikation mit der Außenwelt. Hierbei ist USART ein Akronym für „Universal Serial Asynchronous Receiver Transmitter“. Ein USART ist eine serielle Schnittstelle, mit deren Hilfe binäre Daten zeitlich hintereinander über eine Leitung gesendet und über eine zweite Leitung empfangen werden können. Rechnersystem können so miteinander kommunizieren.

Diese vier USART-Funktionsblöcke im obigen Blockschaltbild sind sowohl mit dem internen Systembus des Mikrocontrollers, aber auch mit bestimmten Ports verbunden. Zum Beispiel ist der USART0 mit dem Port E verbunden, angedeutet durch die Linie mit zwei Pfeilspitzen.

Dies ist nun eine der besonderen Funktionalitäten des Port E. Zwei Bits von Port E können anstatt als digitale I/O-Pins alternativ als serielle Schnittstelle (Tx/D und Rx/D, Sende- und Empfangsleitung) verwendet werden. Das gleiche gilt analog für die drei anderen USARTs, die mit anderen Ports verbunden sind.

Es gibt noch einen Funktionsblock SPI (Serial Peripheral Interface) und TWI (Two Wire Interface), der es dem Mikrocontroller erlaubt mit externen Geräten zu kommunizieren.

³ <https://www.microchip.com/wwwproducts/en/ATMEGA16U2>

© Uwe Homm Version vom 1. April 2019 D:\Users\Uwe\Documents\(\Schule)\Lehrgang_Neu\(\BG TI)\(12BGTI-Q2) Digitale Steuerungstechnik\Skript\Mikrocontroller\05_Hardwareüberblick.odt

Arbeitsblatt Nr. 05	Q2 Technikwissenschaft: Digitale Steuerungstechnik	
Datum:	Thema: Hardwareüberblick	
Seite 3 von 4	Name:	

B
S
G
G

Timer/Counter

Eine weitere, bei Mikrocontrollern häufig anzutreffende Funktionalität, sind sogenannte Timer/Counter, kurz: T/C. Hier von gibt es ebenfalls mehrere, nämlich insgesamt sechs T/C. Davon sind zwei 8-Bit T/C und vier 16-Bit T/C.

Timer/Counter lässt sich als programmierbare „Zeitgeber/Zähler“ verstehen. Zähler kennen Sie bereits. Diese Zähler sind eben 8- oder 16-Bit-Zähler; können also von 0 bis 255 (2^8) oder von 0 bis 65535 (2^{16}) zählen. Die Zähler können vor- und rückwärts zählen; je nach Programmierung. Man kann diese T/C nun so programmieren, dass sie z.B. einen intern festgelegten Pin bei einem bestimmten Zählerstand schalten.

Das Wichtigste hierbei ist: Diese T/C arbeiten nach der Programmierung völlig unabhängig von irgendwelcher Software, die der Mikrocontroller gerade ausführt! Man konfiguriert sie (z.B. im Initialisierungsteil des Assembler-Programms) und dann arbeiten sie völlig unabhängig. Das Blinken einer LED kann somit auch durch einen Timer/Counter realisiert werden, ohne dass es irgendwelche Warteschleifen gibt, in die man den Mikrocontroller schickt und währenddessen er eigentlich nichts macht. Das ist aber nur eine mögliche Nutzung... es gibt noch viele weitere Anwendungen.

Diese T/C0 bis T/C5 sind nun ebenfalls mit einigen der Ports verbunden und stellen damit eine weitere besondere Funktionalität dar. So ist z.B. Port L mit T/C4 und T/C5 verbunden.

Analog-Digital-Converter

Der ATmega 2560 enthält einen 10-Bit Analog-Digital-Converter (ADC), der mittels Multiplexing mit bis zu 16 analogen Signalquellen verbunden werden kann. 10-Bit-ADC bedeutet, dass ein analoges Spannungssignal zwischen 0 V und einem festgelegten Maximum in 1024 Stufen aufgelöst wird. Sei als Beispiel das Maximum 2,56 V. Dann beträgt jede Stufe 2,56 V geteilt durch 1024 gleich 2,5 mV. Jeder angelegte Spannungswert wird nun durch eine 10-bittige Zahl dargestellt, im Bereich von 0 bis 3FF (0 bis 1023). Dieser Maximalwert kann entweder der internen Referenzspannungsquelle (Bandgap Reference) entnommen oder von außen (Pin 98 des Mikrocontrollers) als externe Referenzspannung zugeführt werden.

Auch hier sind diese 16 Analog-Eingänge eine Sonderfunktion; in diesem Fall von Port F und Port K. Diese Anschlüsse können dann nicht als digitale I/O-Pins genutzt werden.

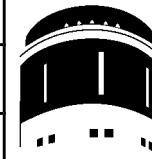
Analog Comparator

Ein Comparator ist ein Vergleicher. Hier werden analoge Spannungswerte verglichen und ein Ausgang wird entsprechend geschaltet, um bestimmte Funktionen auszulösen.

Interner Speicher

Im Mikrocontroller ATmega 2560 sind mehrere Speicherelemente integriert. Es gibt

- einen 256KByte großen Flash-Speicher, der das Maschinenprogramm aufnimmt. Da die Maschinenbefehle 16 oder 32 Bit umfassen, ist der Speicher in der Form 128K x 16 Bit organisiert.
- Ein 8 KByte großer SRAM (Static Random Access Memory). SRAM wird durch eine Spannung gepuffert und benötigt im Unterschied zu DRAM keinen Refresh. Das interne SRAM kann durch externes RAM erweitert werden. Hierzu dient der Funktionsblock XRAM
- Ein 8KByte großes EEPROM (Electrically Erasable Programmable Read Only Memory). Das ist ein elektrisch lösch- und programmierbarer Nur-Lese-Speicher, der z.B. Konfigurationsdaten dauerhaft speichern kann.



JTAG (Joint Test Action Group)

„Joint Test Action Group (kurz JTAG) ist ein häufig verwendetes Synonym für den IEEE-Standard 1149.1, der eine Methodik für das Testen und Debuggen integrierter Schaltungen, also Hardware auf Leiterplatten, beschreibt. [...] Zweck des Verfahrens ist es, integrierte Schaltungen (ICs) auf Funktion zu testen, während sie sich bereits in ihrer Arbeitsumgebung befinden, beispielsweise verlötet auf einer Platine. Dazu besitzt ein JTAG-fähiges IC Komponenten, die im Normalbetrieb vollkommen abgetrennt sind und somit die Funktion des Bauteils nicht stören. Erst durch Aktivierung der JTAG-Funktion an einem bestimmten Pin, dem *Test Mode Select Input*, kann mit Hilfe dieser zusätzlichen Komponenten das Hardwaresystem beeinflusst und dadurch kontrolliert werden.“⁴

Da war noch was...

Links oben entdeckt man noch einen sogenannten „Watchdog Timer“ und Watchdog Oscillator“. Damit kann der Mikrocontroller automatisch neu gestartet werden, wenn der Watchdog Timer (und das ist im Wesentlichen ein Zähler) einen bestimmten Zählerstand erreicht. Gezählt werden die Impulse des Watchdog Oscillator, der eine Frequenz von 128 kHz hat und den Watchdog Timer damit taktet.

Außerdem verfügt der Mikrocontroller über ein Power Management, um Energie zu sparen. Damit können nicht benötigte Funktionsblöcke „schlafen gelegt“ werden.

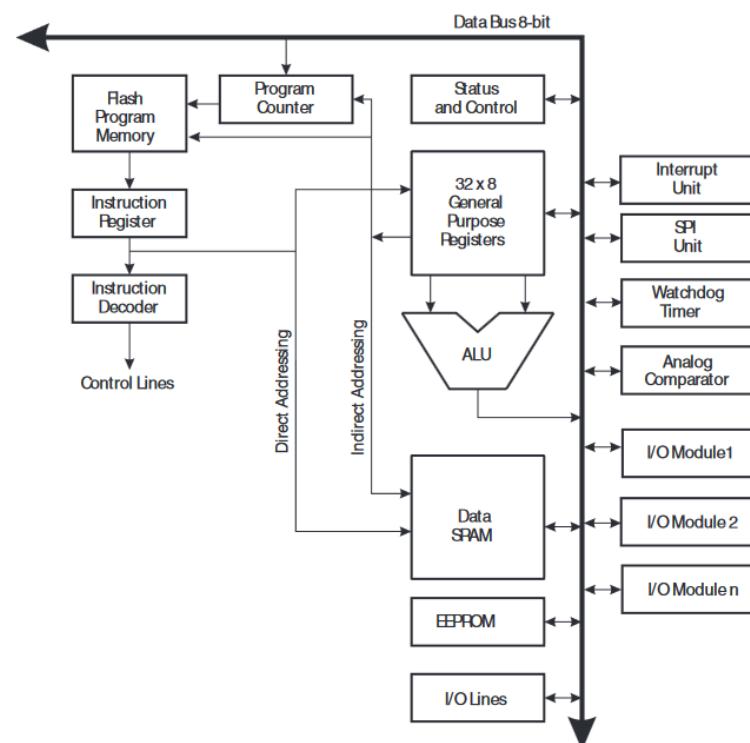
Architektur des Kern-Systems

Das folgende Bild zeigt die Kern-Architektur des Mikrocontrollers. Die zugrunde liegende Speicher-Architektur ist eine Harvard-Architektur, bei der Programm- und Datenspeicher voneinander getrennt sind (im Gegenteil zu einer von-Neumann-Architektur, bei der es keine solche Trennung gibt; siehe z.B. PCs).

„In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Re-programmable Flash memory.“

Bei den AVR Mikrocontrollern handelt es sich um sogenannte RISC-Prozessoren (Reduced Instruction Set Computer). Solche Prozessoren haben einen eingeschränkten Befehlssatz; d.h. es gibt keine Maschinenbefehle für komplexe Operationen, wie z.B. Fließkommaberechnungen. Das Gegenteil hierzu sind CISC-Prozessoren (Complex Instruction Set Computer). Bei RISC-Prozessoren werden die meisten Maschinenbefehle innerhalb ei-

nens Maschinenzyklus ausgeführt. Bei CISC-Prozessoren können dies bei komplexen Operationen auch sehr viele MZen sein. Ältere PC-Prozessoren bis zum 80386 waren CISC-Prozessoren.



⁴ Entnommen aus https://de.wikipedia.org/wiki/Joint_Test_Action_Group abgerufen am 1.4.2019
© Uwe Homm Version vom 1. April 2019 D:\Users\Uwe\Documents\(\Schule)\Lehrgang_Neu\(\BG TI)\(12BGTI-Q2) Digitale Steuerungstechnik\Skript\Mikrocontroller\05_Hardwareüberblick.odt