General Purpose Input Output Ports
GPIO-Ports ARM Cortex M4 - TM4C1294

Vorlesung Mikroprozessortechnik

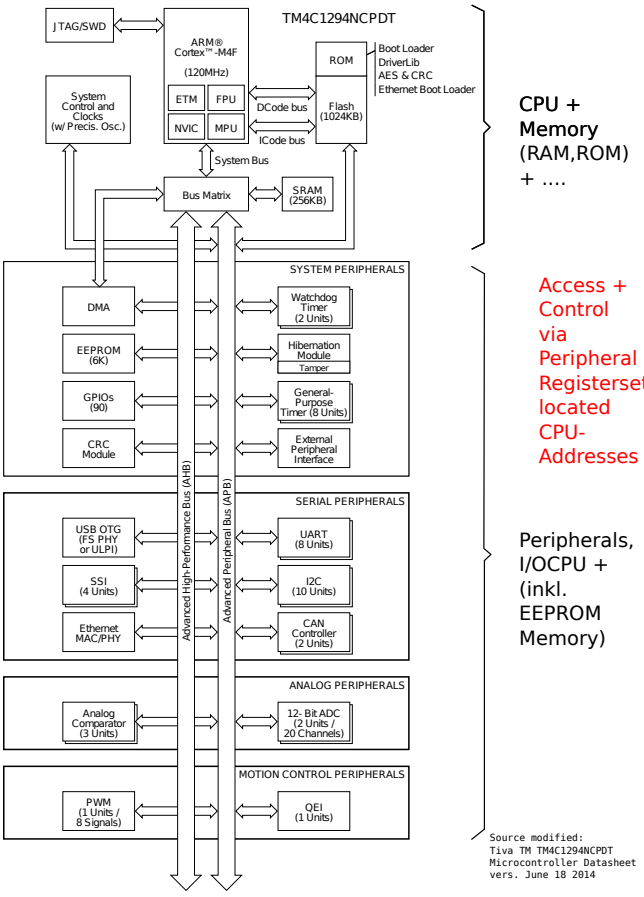
HAW Hamburg

29. Dezember 2017

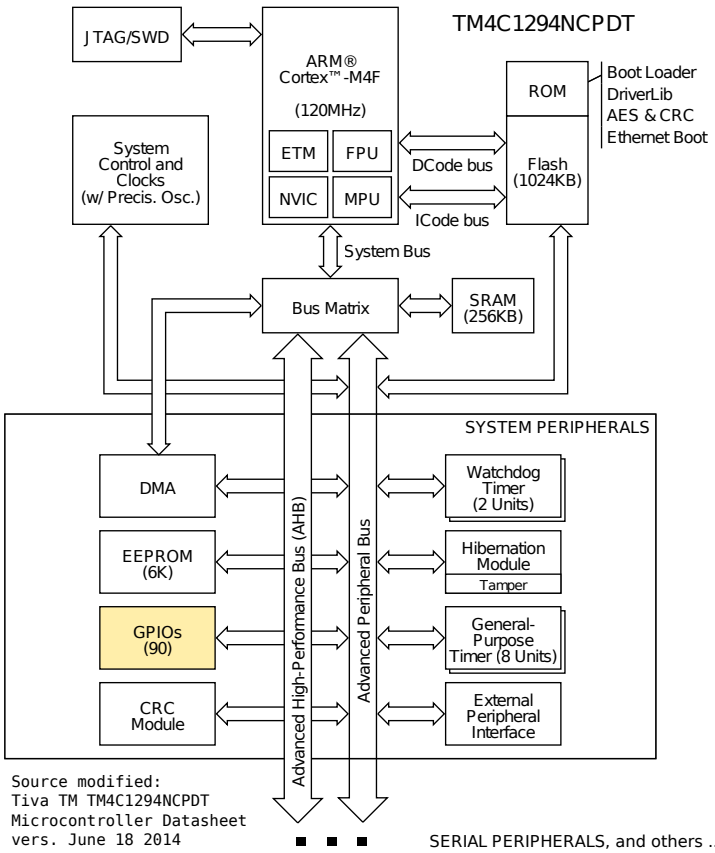
Wiederholung: Bestandteile des Prozessorsystems

- CPU
  - Dataprocessor DP: General Purpose Register Set + ALU + Status Register
  - Control Unit CU: Instruction Decoding Logic/Control, Instruction Register IR, Program Counter PC
- Memory
  - Read Only Memory ROM (Flash und EEPROM im Controller auf dem Chip)
  - Random Access Memory RAM (SRAM im Controller auf dem Chip)
- Bus System
  - Control Bus
  - Address Bus
  - Data Bus
- Input/Output Peripheral Units (I/O)
  - Parallel I/O $\Longrightarrow$ GPIO-Ports
  - Serial I/O $\Longrightarrow$ UART
  - Analog Output DAC, Analog Input ADC ...

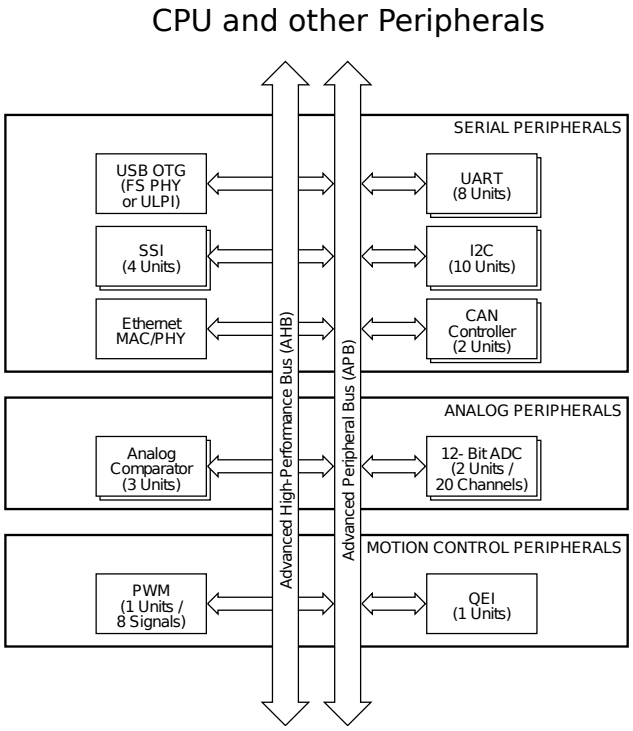
Block Diagram Controller TM4C1294 (I)



Block Diagram Controller TM4C1294 (II)

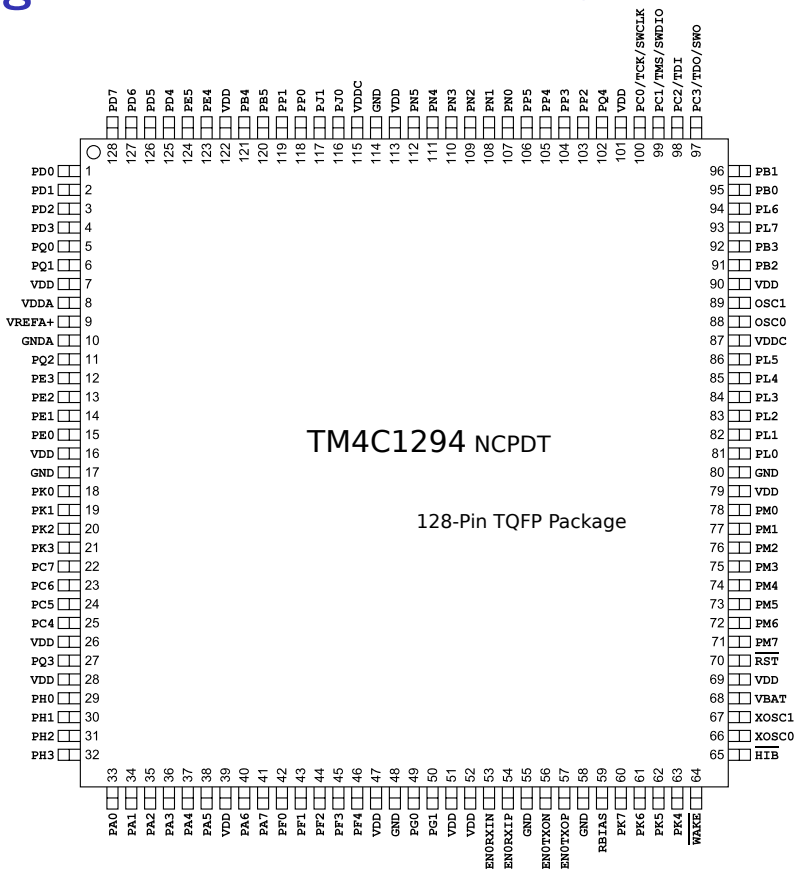


Block Diagram Controller TM4C1294 (III)



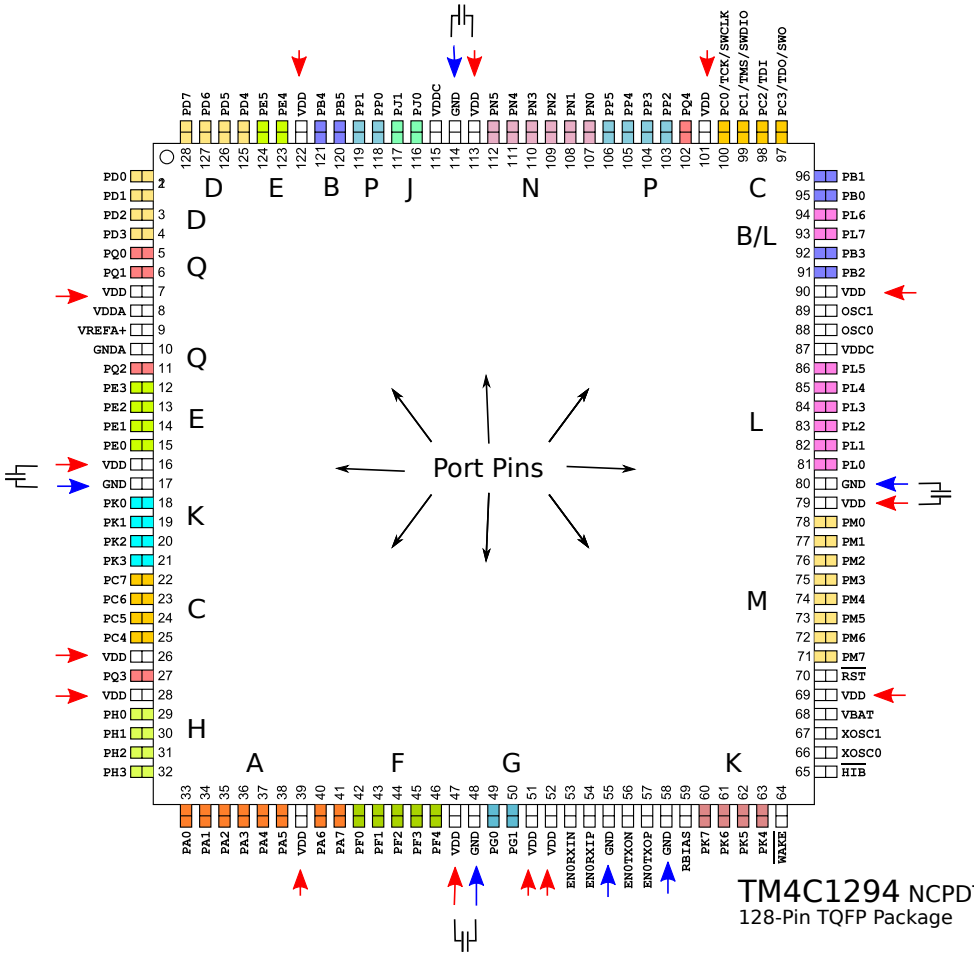
Source modified:
Tiva TM TM4C1294NCPDT
Microcontroller Datasheet
vers. June 18 2014

Pinning Controllers TM4C1294

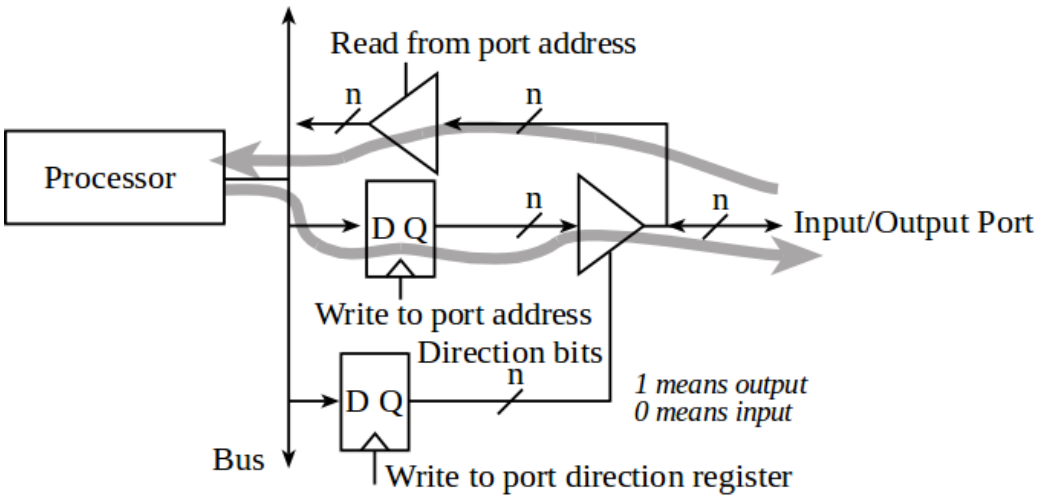


TM4C1294NCPDT microcontroller pin diagram
Source: www.ti.com Tiva TM TM4C1294NCPDT microcontroller data sheet rev. B. June 18, 2014

Portpins des Controllers TM4C1294



Grundfunktion eines I/O Ports



Common Simplified Bidirectional I/O Port

Quelle: Bard, Valvano and Yerraballi, Texasedu.org

General Purpose Input/Output (GPIO) TM4C1294

- 15 GPIO Ports als 'Blöcke' mit der Breite von 8 Pads/Pins
- Port A,B,C,D,E,F,G,H
- kein Port I
- Port J,K,L,M,N
- kein Port O
- Port P,Q
- Nicht alle Pads/Pins sind physikalisch verfügbar, jedoch immer 8-Bit breit in den Port-Registers 'logisch' angelegt
- beim Laborsystem sind nutzbar mit 2mm Buchse + LED :
- Port C Pin 5 und Pin 4
- Port D Pin 5 bis Pin 0
- Port K Pin 7 bis Pin 0
- Port L Pin 6 bis Pin 0
- Port M Pin 8 bis Pin 0
- Port P Pin 1 und Pin 0

GPIO-Ports beim TM4C1294

- Flexibles Pin-Multiplexing erlaubt die Benutzung der Pads/Pins als digitales General Purpose I/O oder für verschiedene andere Peripherie-Funktionen.
- Elektrisches Verhalten der GPIO-Pads ist konfigurierbar [1],z.B.:
- Weak Pull-up oder Pull-down Widerstände (10-40 KΩ) → Register GPIOPUR und GPIOPDR
- Open Drain Schaltung → Register GPIOODR Open Drain Select
- maximaler Ausgangsstrom ('Treiberstärke')
- 2-mA, 4-mA, and 8-mA Pad Drive, bis zu vier Pads als 18-mA Senke
- Slew Rate Control → Register GPIOSLR
- Als GPIO-Interrupt Quellen nutzbar, generell je PORT 1 Quelle / Port P und Q Pinweise

[1] (Ausnahmen bei PJ1)

Exkurs: Bezeichungen von Betriebspannungen

Vcc = Spannung an den Kollektoren, bei bipolaren ICs positive Versorgungsspannung, bei TTL +5V,"common collector voltage"

Vdd = positive Versorgungsspannung von MOS/CMOS Schaltkreisen (Vdd FET-Drains), heute oft 3.3V oder niedriger

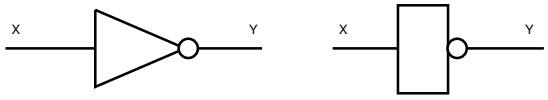
Vee = Spannung an den Emittern, negative Versorgungsspannung z. B. bei ECL-ICs

Vss = negative Versorgungsspannung von MOS/CMOS Schaltkreisen, oft identisch mit GND (Vss FET-Sources)

GND = Ground siehe Vss

Exkurs: CMOS-Logik

Symbole: | global üblich: US ANSI 91-1984 | deutscher Sprachraum: IEC 60617-12 | binäre Logik

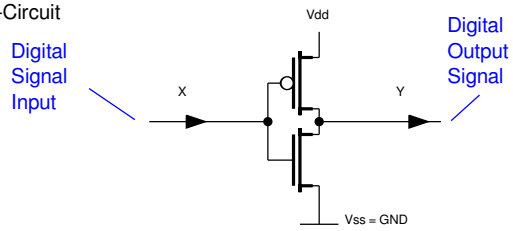


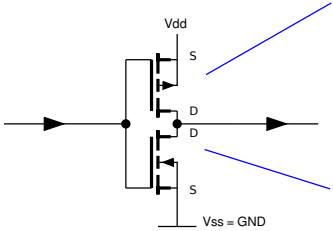
$Y = \overline{X}$

$Y = \neg X$

Y = NOT(X)

CMOS-Circuit





PMOS FET *)
Source-Drain Strecke als Pull-up-Path ist:
- **sperrend** (sehr hochohmig) bei positiver Gatespannung
- **leitend** (niederohmig) bei Gatespannung Null oder negativ

NMOS FET *)
Source-Drain Strecke als Pull-Down-Path ist:
- **leitend** (niederohmig) bei positiver Gatespannung
- **sperrend** (sehr hochohmig) bei Gatespannung Null od. negativ

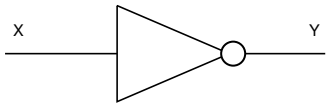
*) genauer Anreicherungstypen

CMOS - Logik:
Complementary
Metal-**O**xide-**S**emiconductor

= paarweise Kombination von PMOS-FET und NMOS-FET für Logik-Funktionen

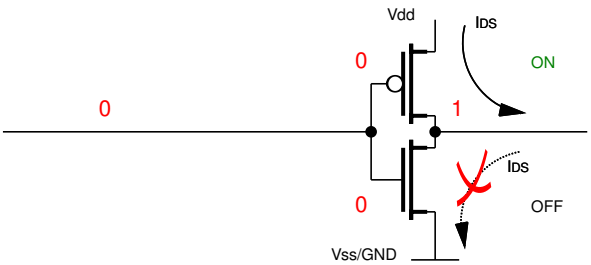
Exkurs: Grundfunktion CMOS-Inverter

Symbol



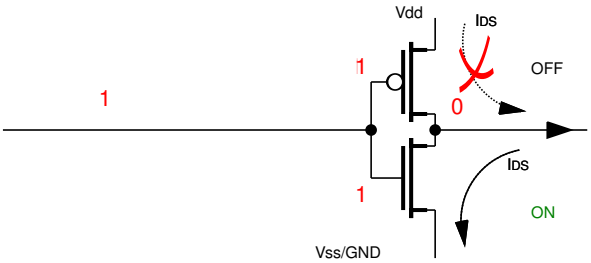
Digital Signal
Input 0

Digital Output
Signal 1

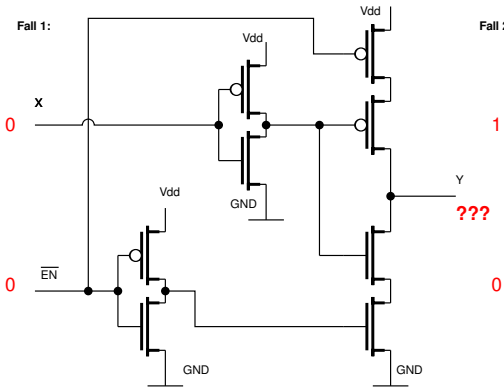


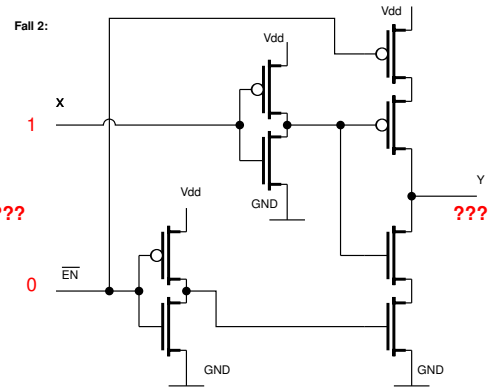
Digital Signal
Input 1

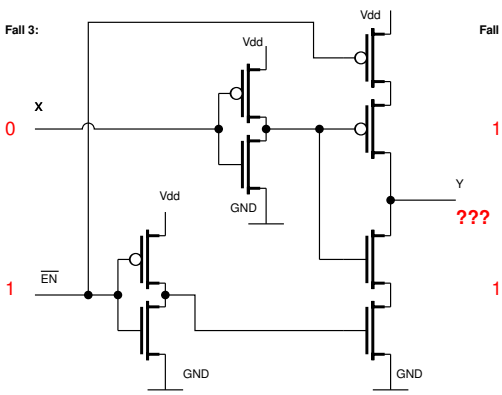
Digital Output
Signal 0

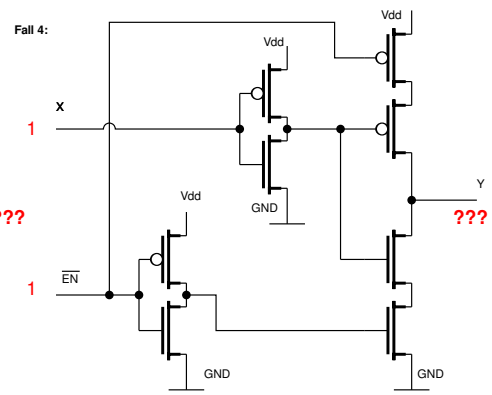


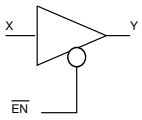
Arbeitsblatt: Bestimmen Sie den Ausgangswert ???











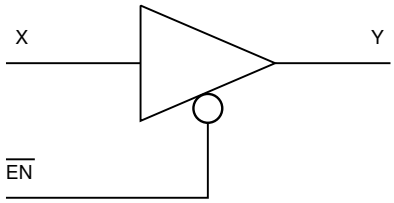
Aufgabe:

a) Ermitteln Sie die Ausgangswerte Y (???) der vier Fälle, in dem Sie die Transistoren "schalten". Notieren Sie den Eingangswert und Schaltzustand (on/off) an **jedem** Transistor.

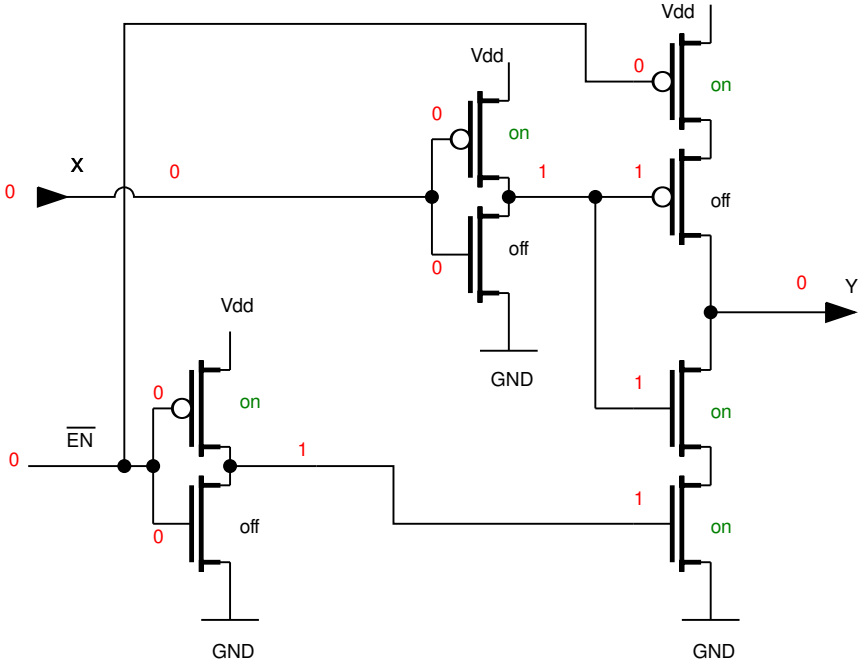
b) Vervollständigen Sie die Wahrheitstabelle.

$\overline{EN}$	X	Y
0	0	. . .
0	1	. . .
1	0	. . .
1	1	. . .

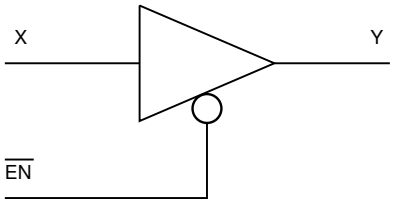
Tristate-Gate Fall 1



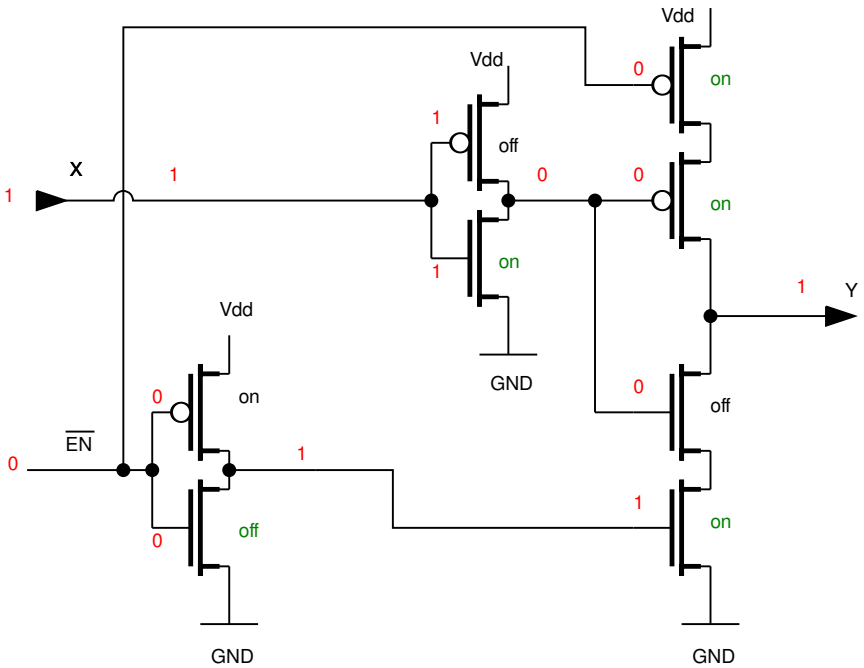
$\overline{EN}$	X	Y
0	0	0
0	1	1
0	1	high ohmic
1	1	high ohmic



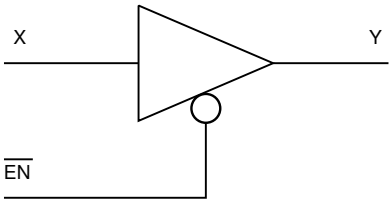
Tristate-Gate Fall 2



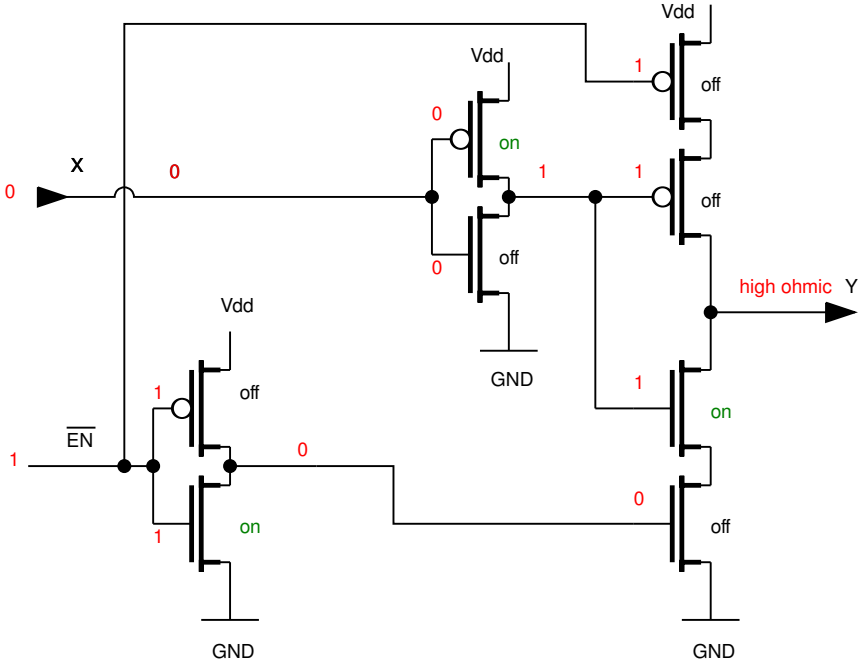
$\overline{EN}$	X	Y
0	0	0
0	1	1
0	1	high ohmic
1	1	high ohmic



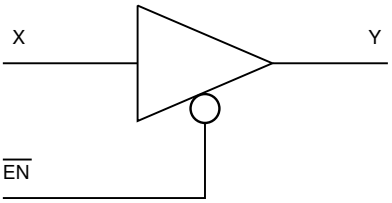
Tristate-Gate Fall 3



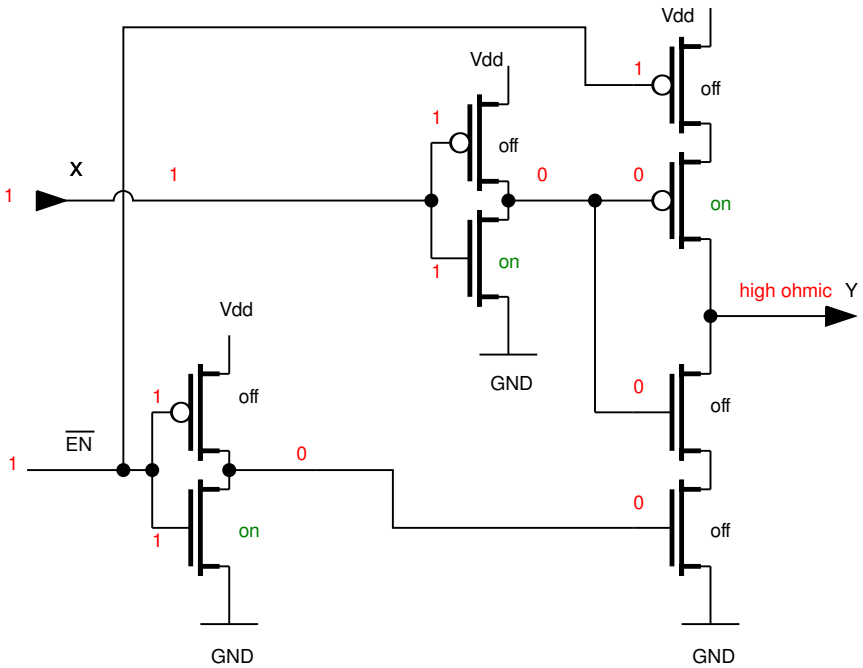
$\overline{EN}$	X	Y
0	0	0
0	1	1
1	0	high ohmic
1	1	high ohmic



Tristate-Gate Fall 4

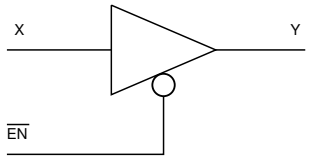


$\overline{EN}$	X	Y
0	0	1
0	1	0
1	0	high ohmic
1	1	high ohmic

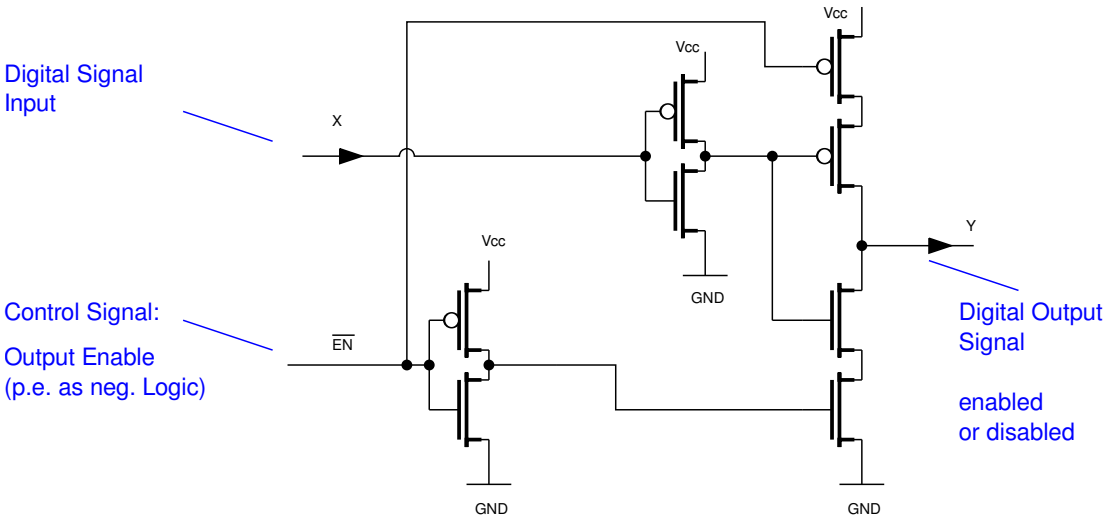


Tristate-Gate mit drittem Logikzustand ermöglicht das 'Abschalten eines Ausgangs'

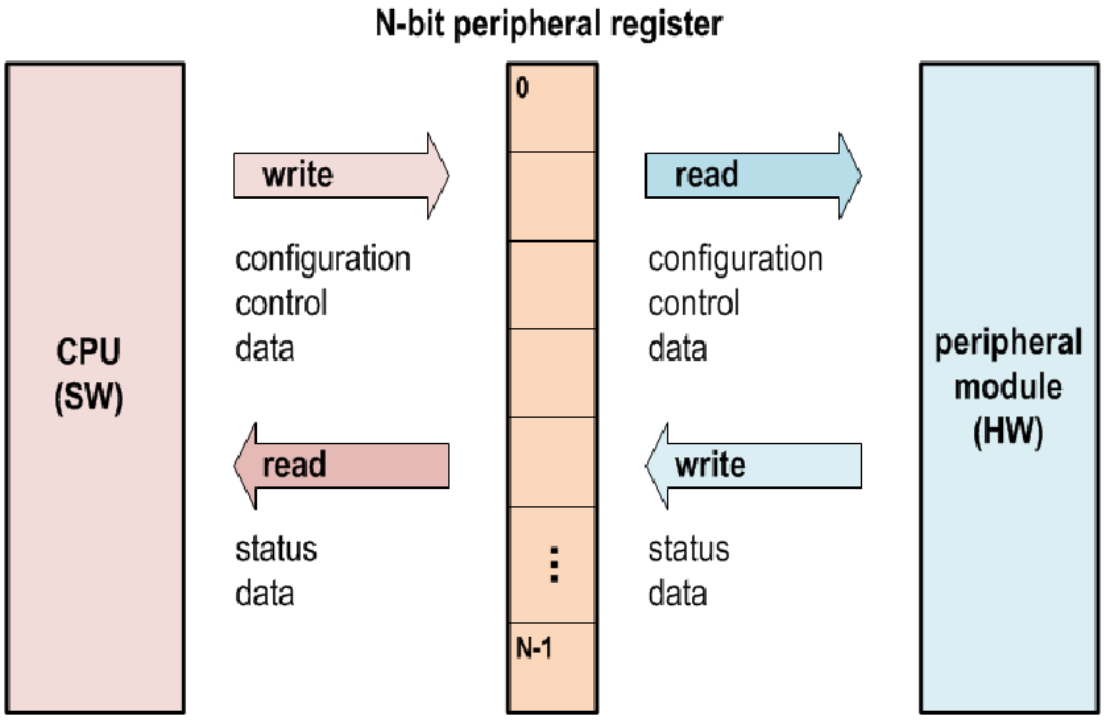
Symbol



Circuit

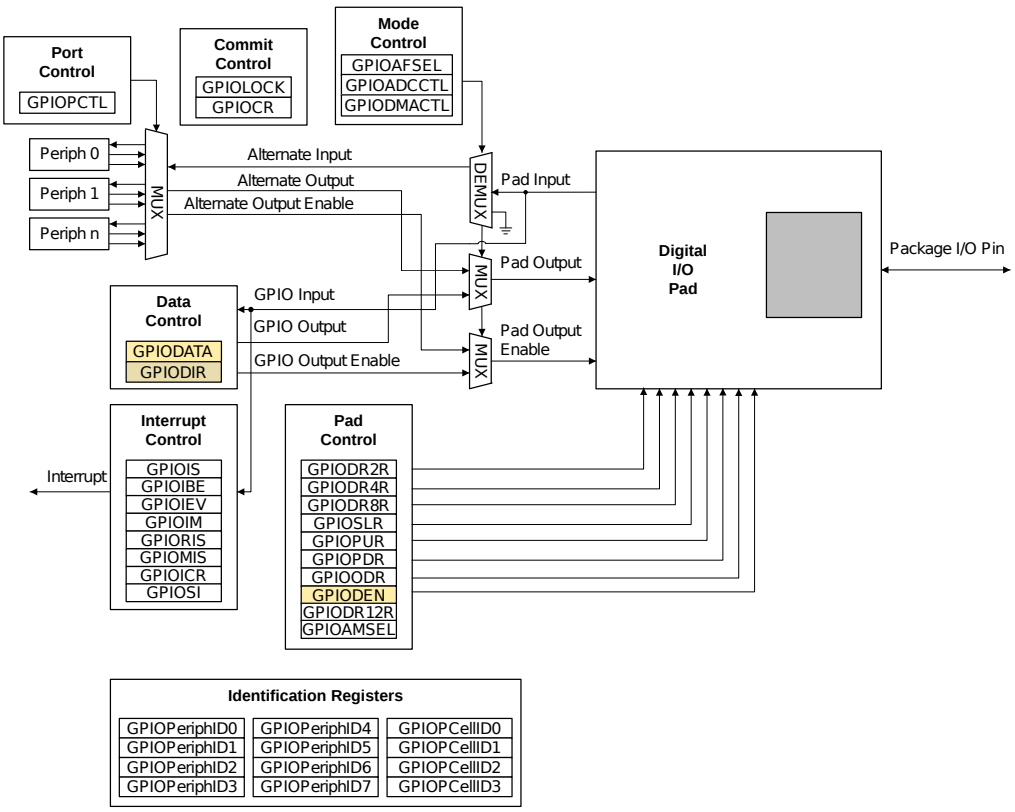


Zwei Seiten der Peripherie-Register



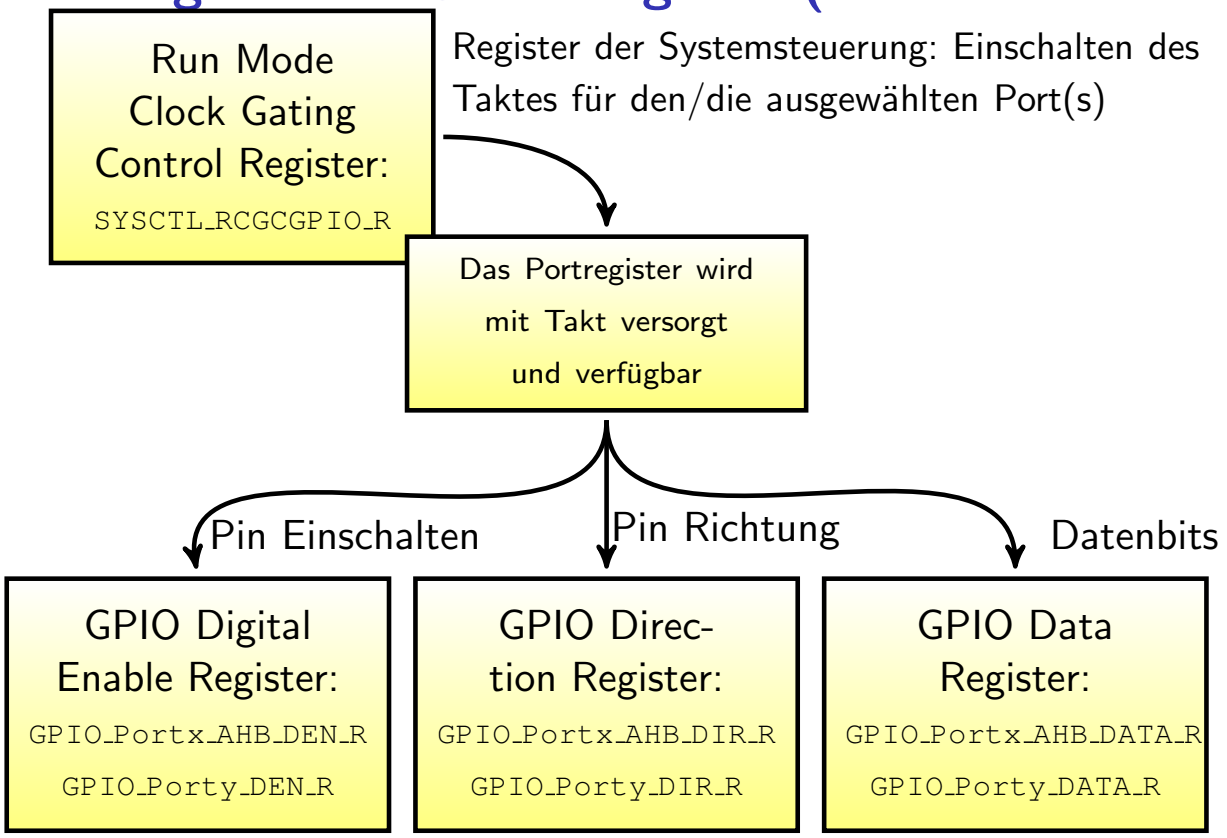
Quelle Lutz Leutelt HAW Vorlesungsunterlagen

Digital Only I/O Pads



Source Texas Instruments, Datasheet Tiva TM TM4C1294NCPDT Microcontroller June 18,

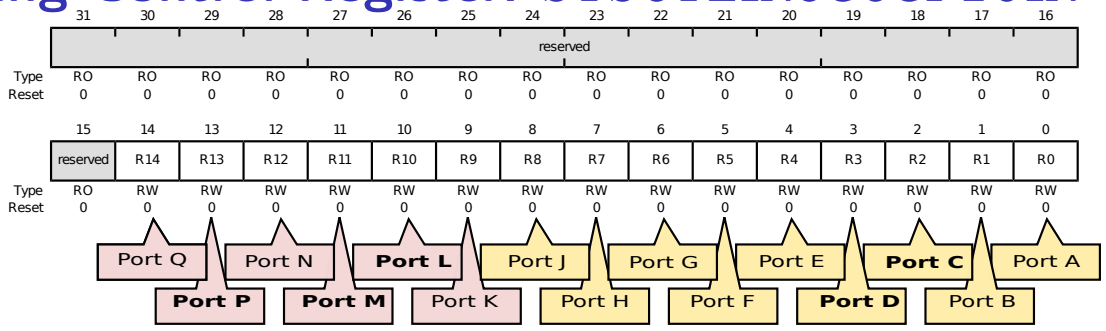
Nutzung der GPIO-Portregister (Minimalvariante)



x = A,B,C,D,E,F,G,H,J : AHB Ports (Port I not available)

y = K,L,M,N,P,Q: APB Ports (non AHB Ports) (Port O not available)

General-Purpose Input/Output Run Mode Clock Gating Control Register: SYSCTL_RCGCGPIO_R



Bit/Field	Name	Type	Reset	Description
31:15	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
14	R14	RW	0	GPIO Port Q Run Mode Clock Gating Control Value Description 0 GPIO Port Q is disabled. 1 Enable and provide a clock to GPIO Port Q in Run mode.
0	R0	RW	0	GPIO Port A Run Mode Clock Gating Control Value Description 0 GPIO Port A is disabled. 1 Enable and provide a clock to GPIO Port A in Run mode.

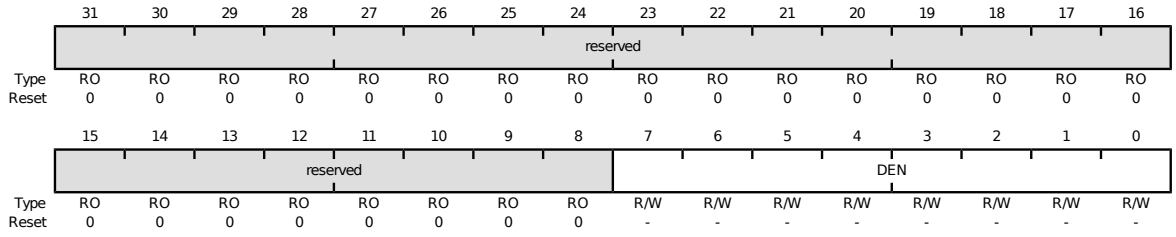
Source Datasheet Texas Instruments TM4C1294 Microcontroller, 18. June 2014, page 382ff, modified

GPIO Digital Enable Register:

GPIO_Portx_AHB_DEN_R (x = A,B,C,D,E,F,G,H,J)

GPIO_Porty_DEN_R (y = K,L,M,N,P,Q)

GPIO Digital Enable Register (GPIO_PORTx_DEN_R)



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DEN	R/W	-	Digital Enable Value Description 0 The digital functions for the corresponding pin are disabled. 1 The digital functions for the corresponding pin are enabled. The reset value for this register is 0x0000.0000 for GPIO ports .

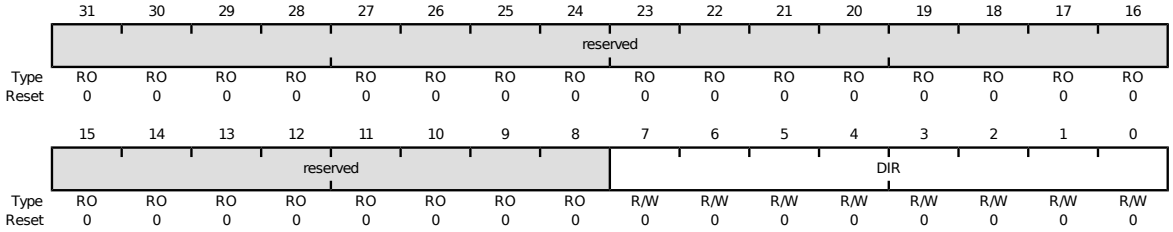
Source Datasheet Texas Instruments TM4C1294 Microcontroller, 18. June 2014, p. 782, modified

GPIO Direction Register:

GPIO_Portx_AHB_DIR_R (x = A,B,C,D,E,F,G,H,J)

GPIO_Porty_DIR_R (y = K,L,M,N,P,Q)

GPIO Direction Register (GPIO_PORTx_DIR_R)



Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DIR	R/W	0x00	GPIO Data Direction

Value	Description
0	Corresponding pin is an input.
1	Corresponding pins is an output.

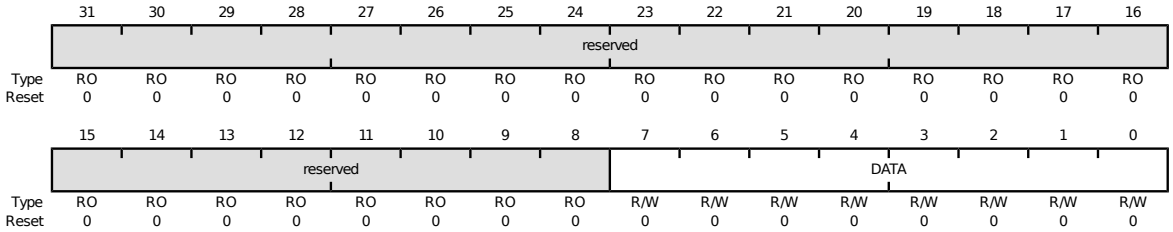
Texas Instruments TM4C1294 Microcontroller, 18. June 2014, p. 760, modified

GPIO Data Register:

GPIO_Portx_AHB_DATA_R (x = A,B,C,D,E,F,G,H,J)

GPIO_Porty_DATA_R (y = K,L,M,N,P,Q)

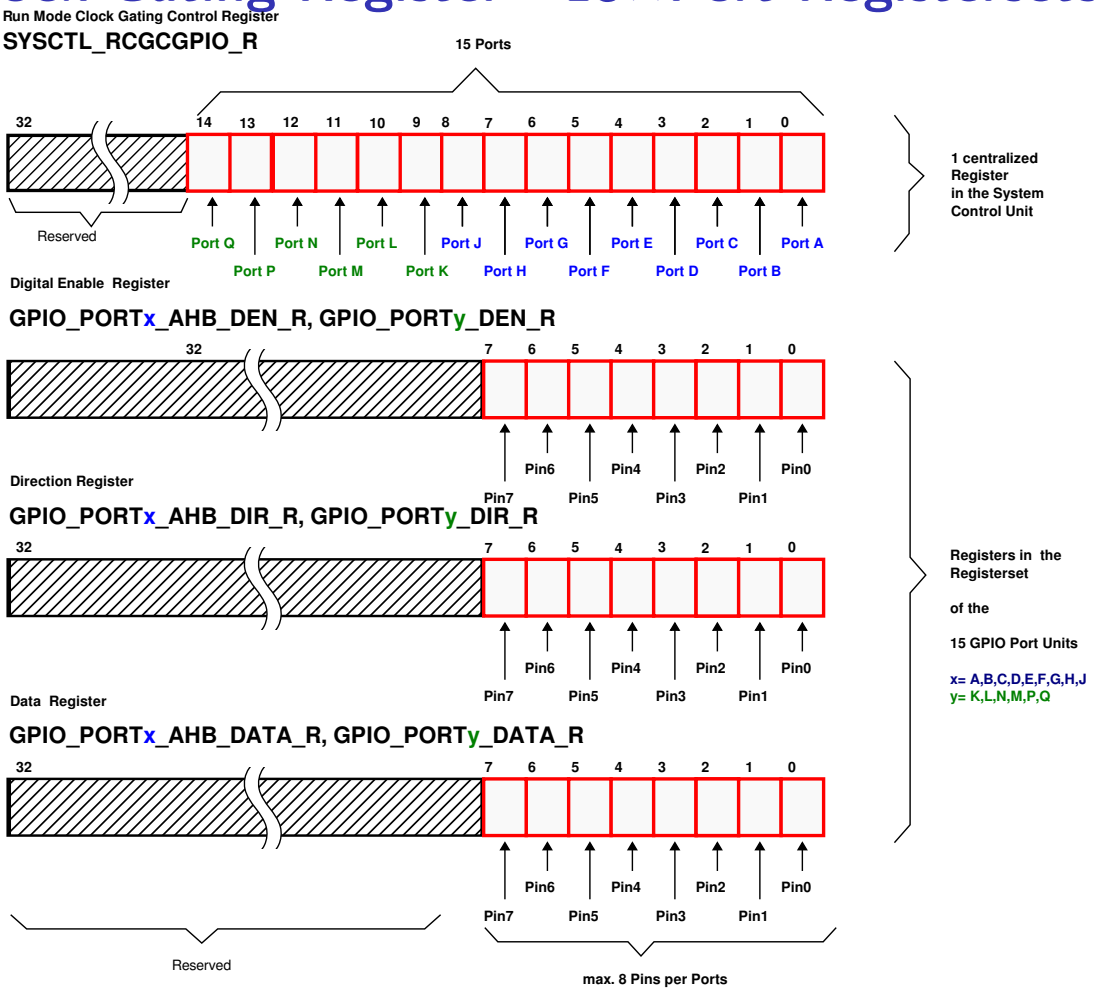
GPIO Data Register (GPIO_Portx_DATA_R)



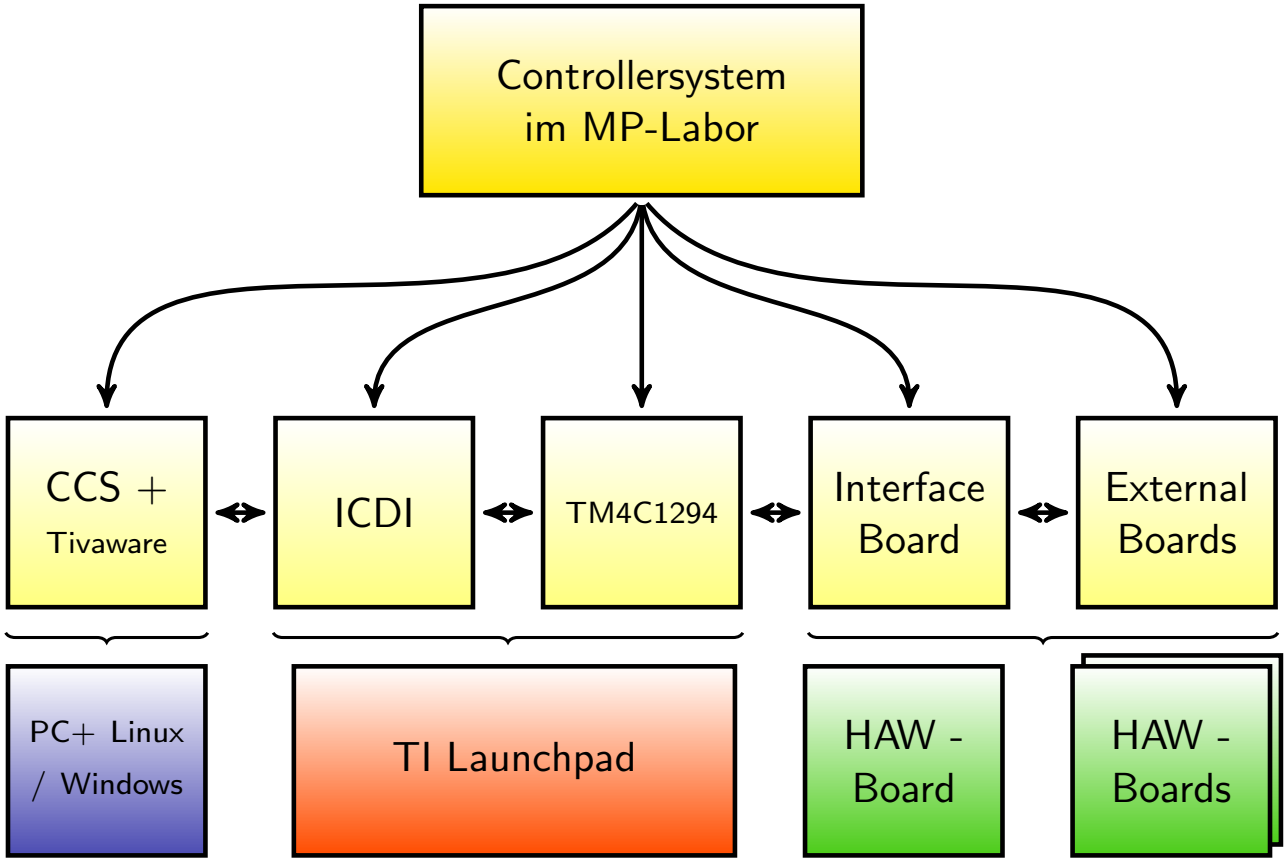
Bit/Field	Name	Type	Reset	Description
31:8	reserved	RO	0x0000.00	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.
7:0	DATA	R/W	0x00	GPIO Data This register is virtually mapped to 256 locations in the address space. To facilitate the reading and writing of data to these registers by independent drivers, the data read from and written to the registers are masked by the eight address lines [9:2]. Reads from this register return its current state. Writes to this register only affect bits that are not masked by ADDR[9:2] and are configured as outputs.

Texas Instruments TM4C1294 Microcontroller, 18. June 2014, p. 759, modified

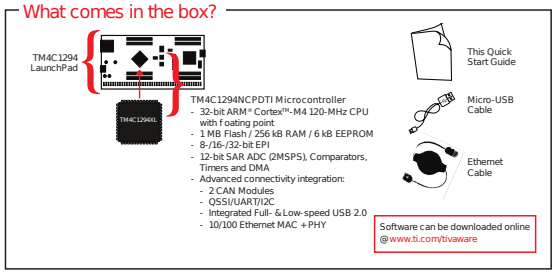
1 Clock Gating Register - 15×Port Registersets

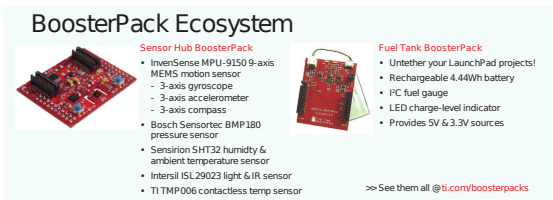


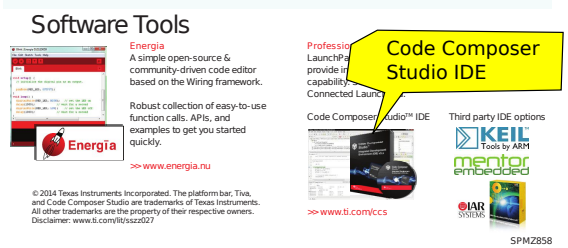
Controllerumgebung im MP-Labor

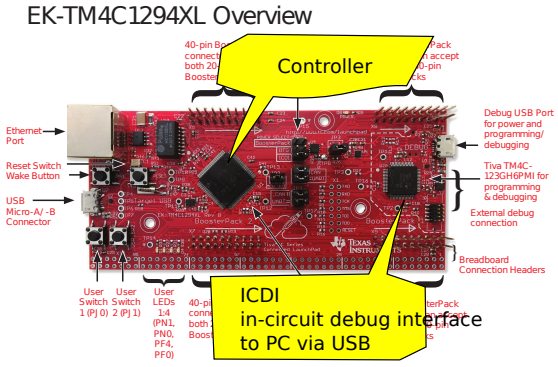


'Connected Launch Pad TM4C1294' als Basis









Source modified: spmz8258 www.ti.com

Code Composer Studio IDE

- IDE = Integrierte Entwicklungsumgebung
  - Quelltexteingabe
  - Compiler
  - Debugger
  - Download vom Host-PC auf den Controller
- Basiert auf Eclipse (als universelle IDE)
- Umgebung für Prozessoren/ Mikrocontroller von Texas Instruments
  - ARM Cortex M, R, A
  - MSP430
  - TMS320CXXX DSP's

Tivaware

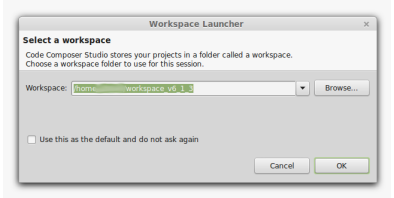
- Marketing-bezeichung / Trademark: "Tiva ..."[2]
- Tivaware = Firmware-Sammlung angelehnt den an Cortex Microcontroller Software Interface Standard CMSIS - der Fa. ARM
  - Ziel: einheitliche Schnittstellen durch 'schwache' Hardwareabstraktion
  - einheitliche Namenskonventionen für Register durch Makros
  - Konfiguration und Kommunikation mit Peripherie Units
  - Address-Konfiguration für die Typen der Controllerfamilie über Makros in Headerfiles (z.B. Registeradressen)
- Quellcodes der Firmware und Beispielprogramme werden mitgeliefert
  - Ausgangspunkt/Beispiele für eigene Programme, 'Nachschlagewerk' zur Ergänzung der Dokumentation
  - Direct Register Access Model (überwiegend im Fach MP)
  - Software Driver Model (nur im Ausnahmefall im Fach MP, regelmäßig im Fach MC)

[2]Der Marketingname Tiva von der Firma Texas Instruments, löst den Marktetingnamen Stellaris genutzt für eine Controllerfamilie und Firmware

Code Composer Studio: Startprozedur

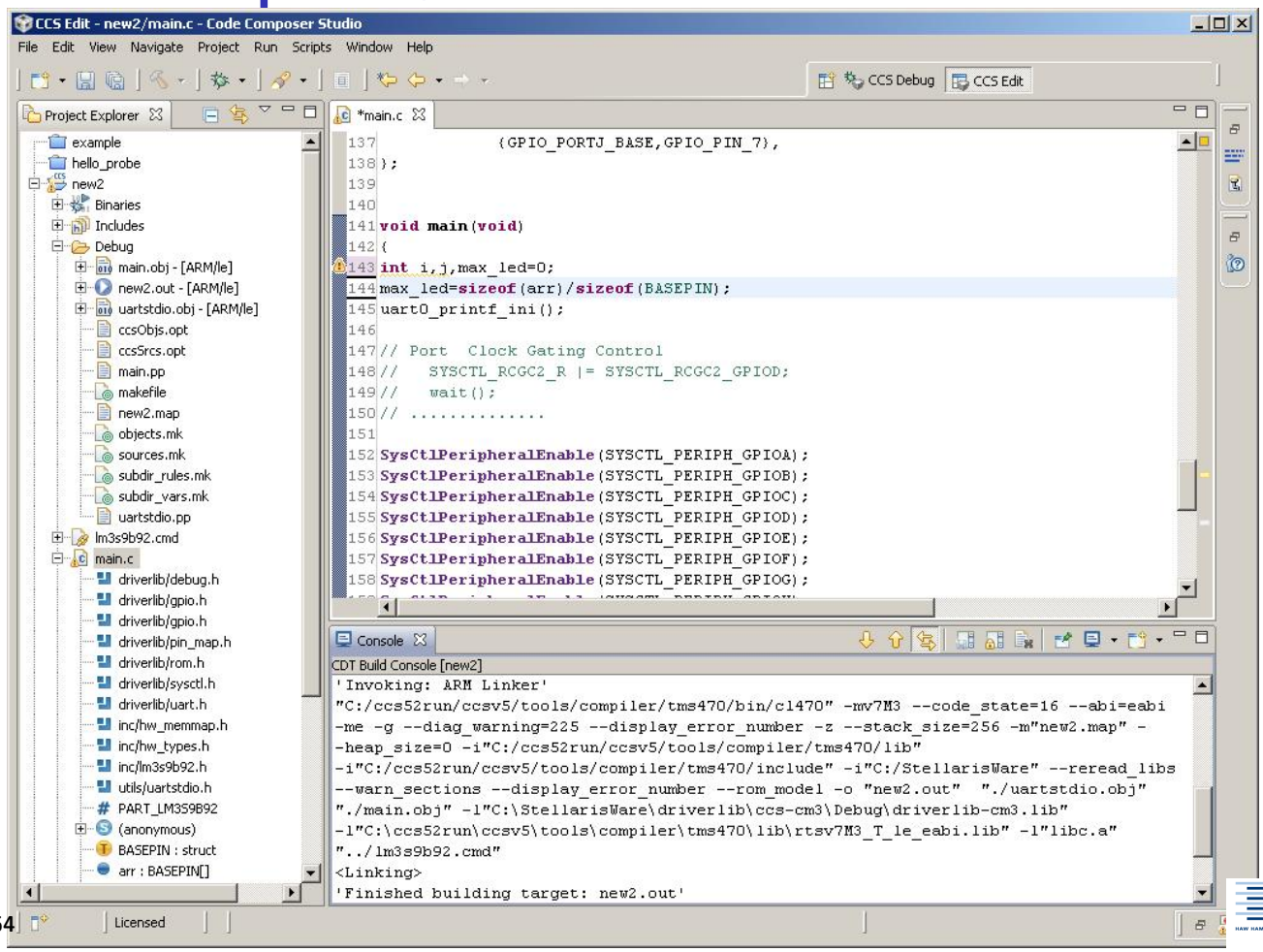
Start Icon - Mouseclick

Welcome Icon - Time to wait

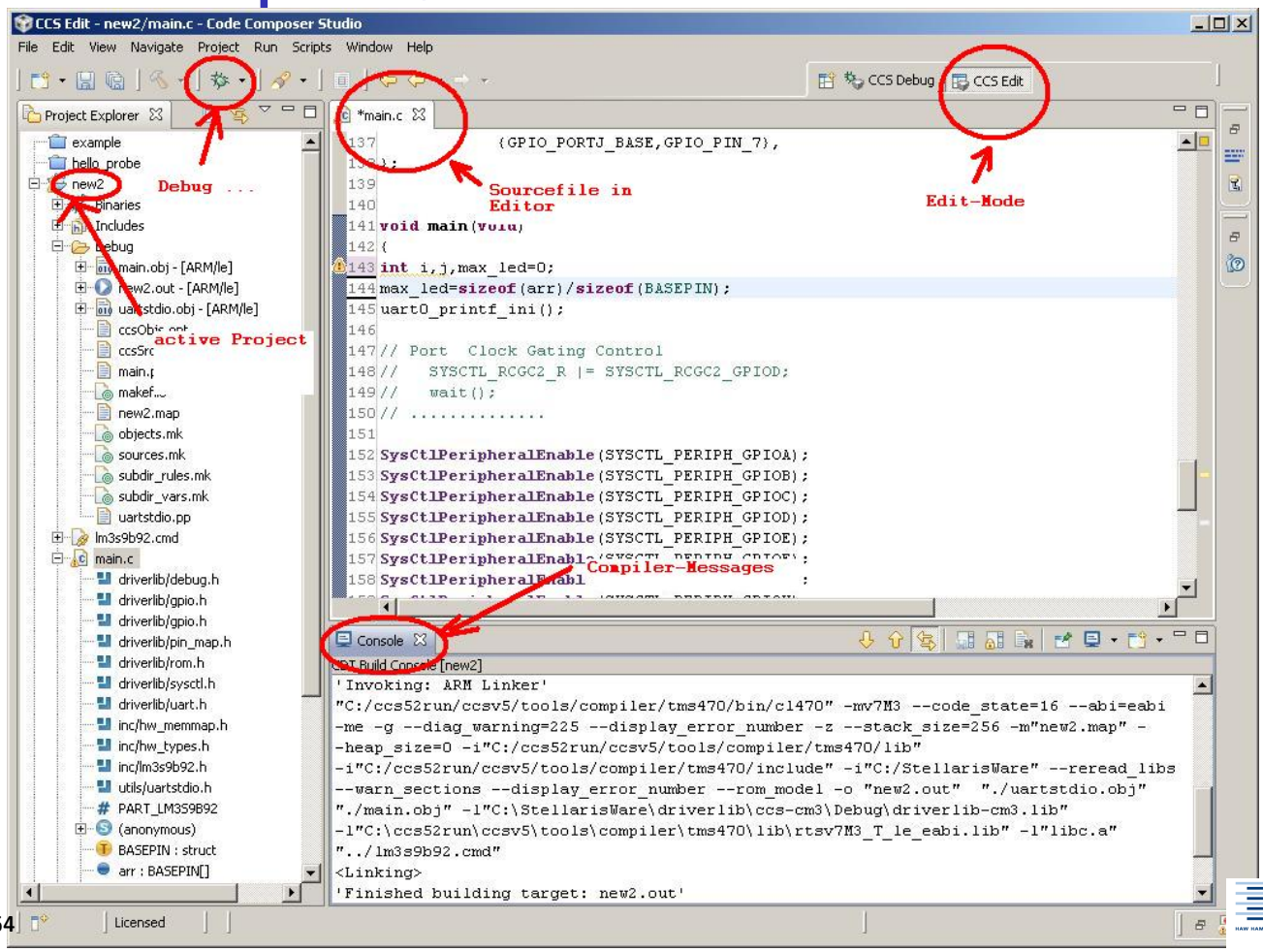


Select Workspace Directory for Projects

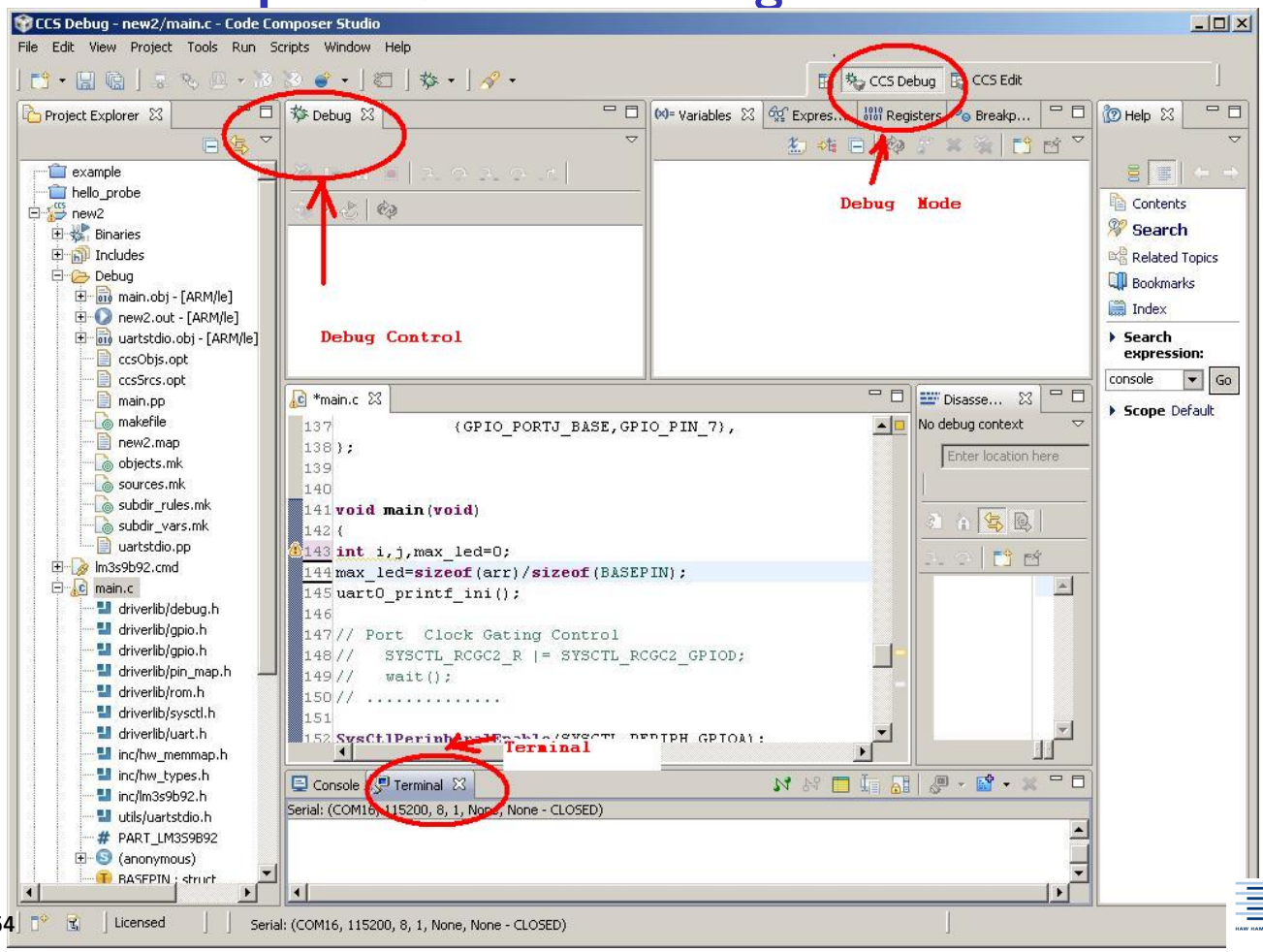
Code Composer Studio



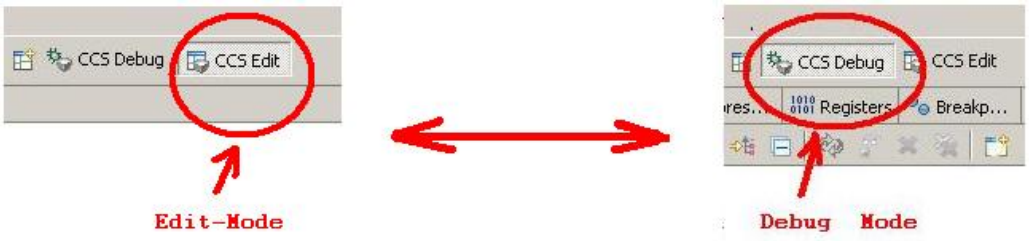
Code Composer Studio - Edit Mode



Code Composer Studio - Debug Mode



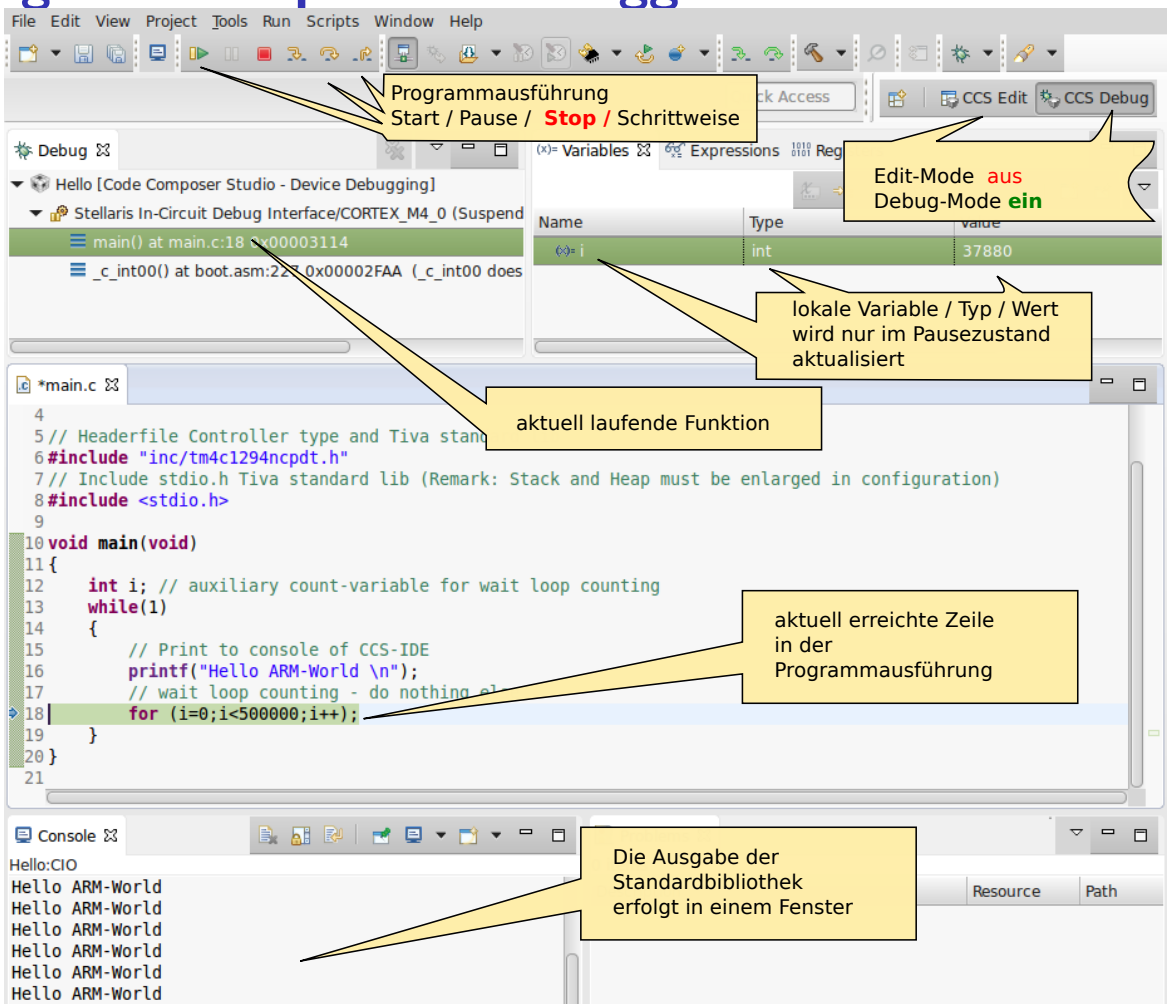
CCS: Edit-Mode und Debug-Mode



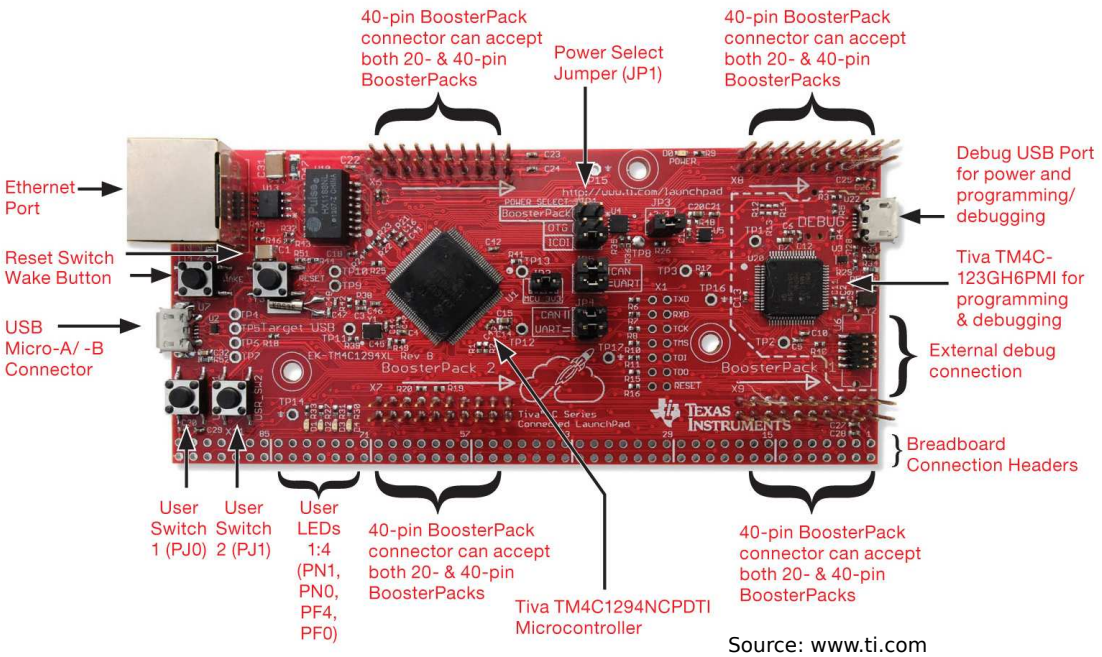
- Im Edit-Mode wird das Quellprogram editiert, als Projekt übersetzt und von Syntax-Fehler befreit.
- Der Übergang in den Debug-Mode ist mit einen Download über das ICDI-Modul in den Speicher des Controllers verbunden.
- Im Debug-Mode kann das Programm im Controller schrittweise oder komplett ausgeführt werden.
- Für Programmänderungen sollte die Ausführung des Programms beendet, der Debug-Mode verlassen werden. Es kann dann erneut im Edit-Mode editiert werden ... usw.

```
1  //*****************************************************************************
2  // Simple Hello World Program for ARM + Tivaware Lib + CCS
3  //*****************************************************************************
4
5  // Headerfile Controller type
6  #include "inc/tm4c1294ncpdt.h"
7  // Include stdio.h Tiva standard lib (Remark: Stack and Heap must be enlarged in
       configuration)
8  #include <stdio.h>
9
10 void main(void)
11 {
12     int i; // auxiliary count-variable for wait loop counting
13     while(1)
14     {
15         // Print to console of CCS-IDE
16         printf("Hello ARM-World \n");
17         // wait loop counting - do nothing else
18         for (i=0;i<500000;i++);
19     }
20 }
```


Programmbeispiel im Debugger

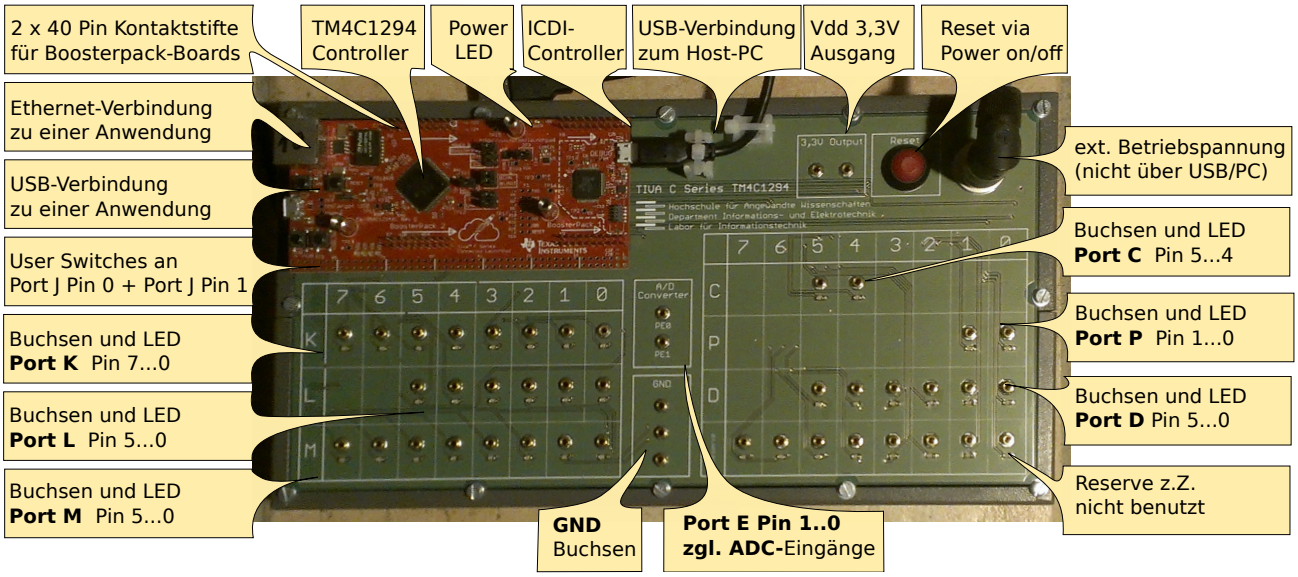


Texas Instruments Launchpad



Source: www.ti.com
TivaTM C Series TM4C1294 Connected LaunchPad Evaluation Kit spmu3675.pf

Anschlüsse des HAW-Interface-Boards



HAW-Interface-Board: 7 v. 15 Ports / 34 v. 90 Pins

Ports	Buchsen	Makro
Port A,B	-	nicht herausgeführt
Port C	Pins 5 und 4	`GPIO_PortC_AHB_{Registername}_R`
Port D	Pins 5 bis 0	`GPIO_PortD_AHB_{Registername}_R`
Port E	Pins 1 und 0	`GPIO_PortE_AHB_{Registername}_R`
Port F,G,H	-	nicht herausgeführt
Port G, H, J	-	nicht herausgeführt
Port K	Pins 7 bis 0	`GPIO_PortK_{Registername}_R`
Port L	Pins 5 bis 0	`GPIO_PortD_{Registername}_R`
Port M	Pins 7 bis 0	`GPIO_PortM_{Registername}_R`
Port N,	-	nicht herausgeführt
Port P	Pins 1 und 0	`GPIO_PortP_{Registername}_R`
Port Q	-	nicht herausgeführt

- `{Registername}= DEN,DIR,DATA` und 29 weitere
- Alle Ports sind im TM4C1294 an den AHB-Bus (Advanced High Performance Bus) angeschlossen. Bei anderen Typen wird zwischen dem Anschluß an dem AHB und dem langsamen APB (High Peripheral Bus) unterschieden. Die Makros sind aus Software-kompatibilitäts Gründen unterschiedlich belassen.

Simple GPIO Port Output for ARM Cortex M4 + Stellaris + CSS 1 of 1

```
1  //*****************************************************************************
2  // Simple Port Output with Macros for TM4C1294 + Tivaware Lib + CCS
3  //*****************************************************************************
4
5  #include <stdint.h>
6  #include "inc/tm4c1294ncpdt.h"
7
8  void main(void)
9  {
10     int i;                  // auxilary for wait loop counting
11     unsigned char c;        // auxilary short wait
12
13     SYSCTL_RCGCGPIO_R = 0x00000008;   // clock enable port D
14     c = 0;                 // short wait before access
15     GPIO_PORTD_AHB_DEN_R = 0x0F;      // digital I/O pins PD0 to PD3
16     GPIO_PORTD_AHB_DIR_R = 0x0F;      // define output direction I/O pins PD0 to PD3
17        while(1)
18        {
19          GPIO_PORTD_AHB_DATA_R = 0x01;    // output "1" at pin PD0, "0" at pins PD1,PD2,PD3
20          for (i=0;i<500000;i++);      // wait loop counting
21          GPIO_PORTD_AHB_DATA_R = 0x02;    // output "1" at pin PD1, "0" at pins PD0,PD2,PD3
22          for (i=0;i<500000;i++);      // wait loop counting
23          GPIO_PORTD_AHB_DATA_R = 0x04;    // output "1" at pin PD2, "0" at pins PD0,PD1,PD3
24          for (i=0;i<500000;i++);      // wait loop counting
25          GPIO_PORTD_AHB_DATA_R = 0x08;    // output "1" at pin PD3, "0" at pins PD0,PD1,PD2
26          for (i=0;i<500000;i++);      // wait loop counting
27       }
28 }
```

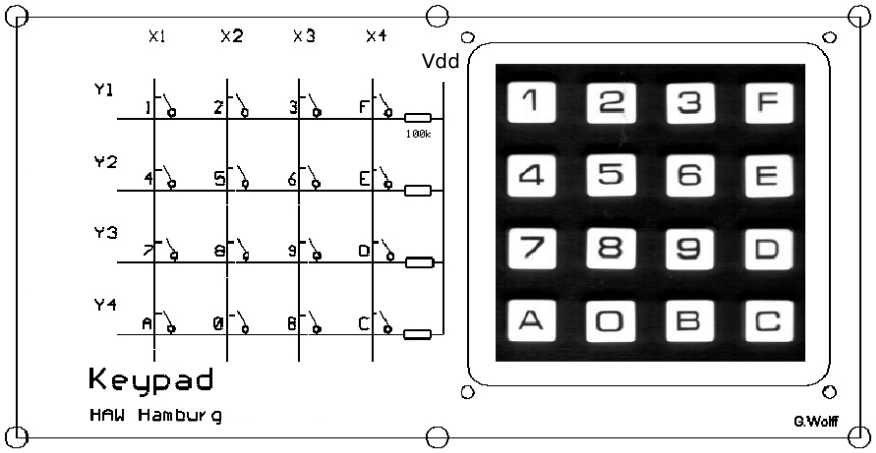

GPIO Port Output with diff. Macros for ARM Cortex M4 + Tivaware + CSS 1 of 1

```
1  //*****************************************************************************
2  // Port Output - Example with Two Ports and Bit-Shift Operations
3  //*****************************************************************************
4  #include <stdint.h> // Include for Type definition (unint32_t)
5  #include "inc/tm4c1294ncpdt.h" // Include for Register Makros
6
7  void main(void)
8  {
9      int i,j;                // auxilary for wait loop counting
10     unsigned char c;        // auxilary short wait
11     SYSCTL_RCGCGPIO_R = 0x000808;   // clock enable port M and D
12     c = 0;                          // short wait before access
13     // Configure Port D (with AHB in Makro for Registers)
14     GPIO_PORTD_AHB_DEN_R = 0x3F;    // digital I/O pins PD0 to PD6
15     GPIO_PORTD_AHB_DIR_R = 0x3F;    // define output direction I/O pins PD0 to PD6
16     // Configure Port M (with AHB in Makro for Registers)
17     GPIO_PORTM_DEN_R = 0xFC;        // digital I/O pins PM2 to PM7
18     GPIO_PORTM_DIR_R = 0xFC;        // define output direction I/O pins PM2 to PM7
19
20     while(1)                        // Endless loop
21     {
22      for(j=0;j<6;j++)               // Loop from 0 to 5
23      {
24      GPIO_PORTD_AHB_DATA_R = 0x01<<j; // output "1" at pin PD0, then PD1 ... until PD5
25                                       // operation: j steps left bit shift of constant 0x01
26      GPIO_PORTM_DATA_R     = 0x80>>j; // output "1" at pin PM7, then PM6 ... until PM2
27                                       // operation: j steps right bit shift of constant 0x80
28      for (i=0;i<500000;i++);          // wait loop counting
29      }
30     }
31 }
```

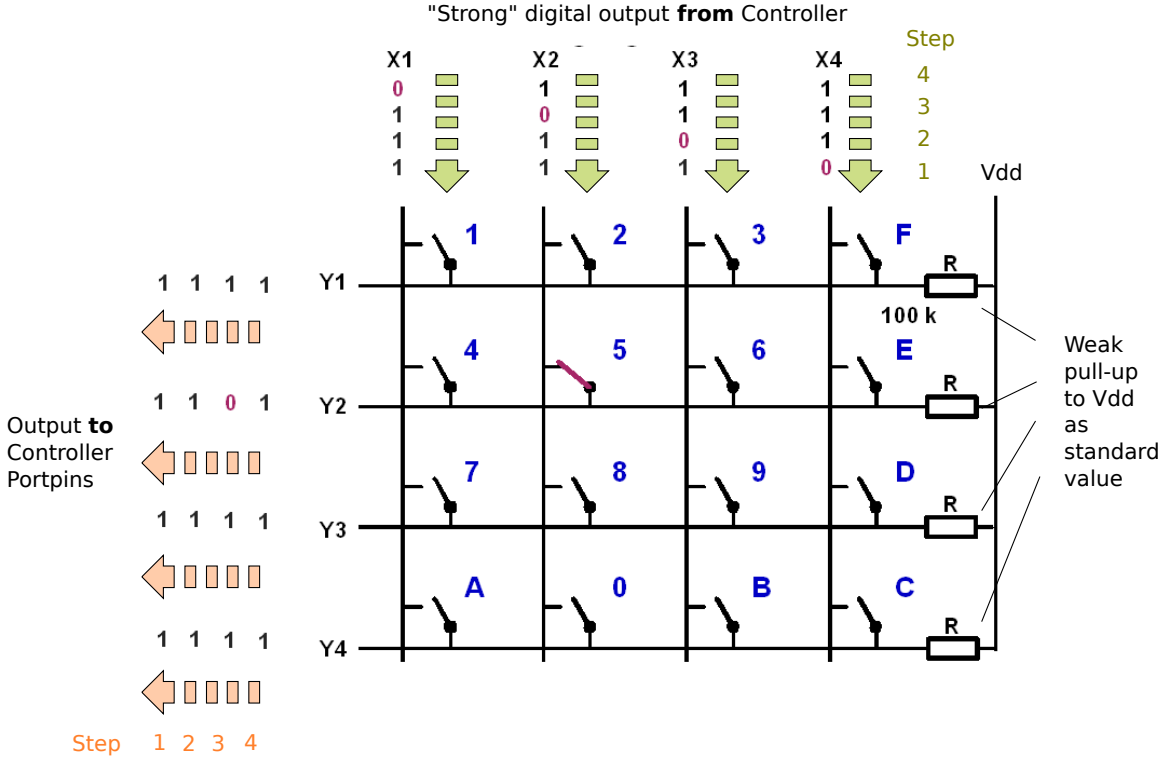

Simple GPIO Port Input for ARM Cortex M4 + Tivaware + CSS 1 of 1

```
1  //*****************************************************************************
2  // Simple GPIO Input Test  Program for ARM / Stellaris / CCS
3  // A pushbutton switch is connected to Pin D0
4  // if PD0 is switched to "1": PD4 set to "1" and PD5 is set to "0"
5  // if PD0 is switched to "0": PD5 set to "1" and PD4 is set to "0"
6  //*****************************************************************************
7
8  // Headerfiles for type definition and controller dependend macros
9  #include <stdint.h>
10 #include "inc/tm4c1294ncpdt.h"
11
12 void main(void)
13 {
14     int i; // auxilary for wait loop counting
15     unsigned char c; // auxilary short wait and input
16
17     SYSCTL_RCGCGPIO_R= 0x00000008; // clock enable port D
18     c = 0;  // short wait before access
19     GPIO_PORTD_DEN_R = 0x31;      // digital I/O pins PD0, PD4 and PD5
20     GPIO_PORTD_DIR_R = 0x30;      // define output direction I/O pins PD4 and PD5
21                                   // PD0 remains as input
22     while(1)
23     {
24         c=GPIO_PORTD_DATA_R;          // Read Port D complete and put to variable
25         if ((c & 0x01) == 0x01)       // Mask variable c with bitwise AND Bit0 and test it
26           GPIO_PORTD_DATA_R = 0x10;  // output "1" at pin PD4
27         else
28           GPIO_PORTD_DATA_R = 0x20;  // output "1" at pin PD5
29     }
30 }
```


Externes Board: Hexadezimale Tastenmatrix

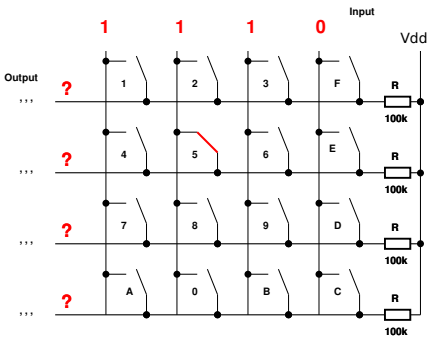


Tastenmatrix: Spalten auswählen - Zeilen abfragen

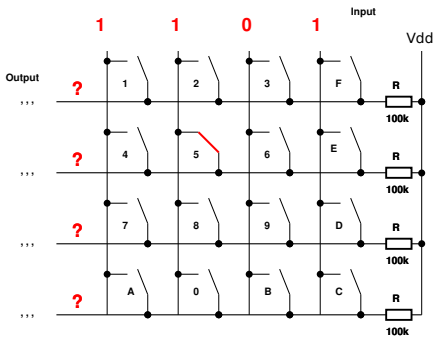


Arbeitsblatt: Tastenmatrix

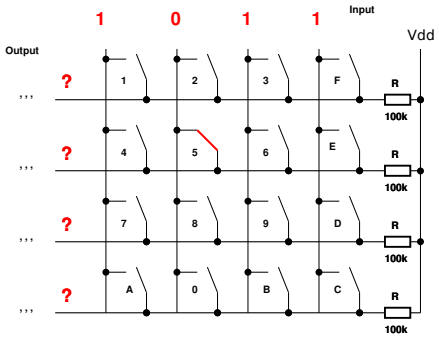
Step 1



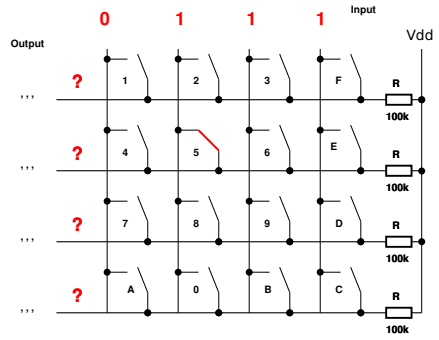
Step 2



Step 3



Step 4



Aufgabe: Schreiben Sie die Output-Werte der Tastatur für die vier Schrtte auf.