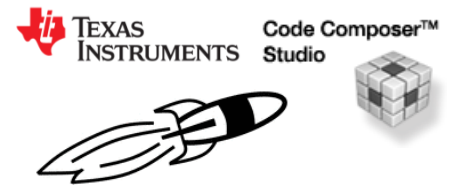




Embedded Systems 2



TIVA: HW/FW-Beschreibung & Erklärungen

Contents

1	Grundstruktur	2
2	Registerbezeichnungen beim TM4C1294NCPDT	3
2.1	Beispiel Direction Register of PortA	3
3	Detaillierter Aufbau	3
3.1	Beispiel GPIO - PortA	3
3.1.1	Struktur	3
3.1.2	Abfolge Registerkonfiguration	4
3.2	Beispiel ADC – ADC0	4
3.2.1	Abfolge Registerkonfiguration	4



1 Grundstruktur

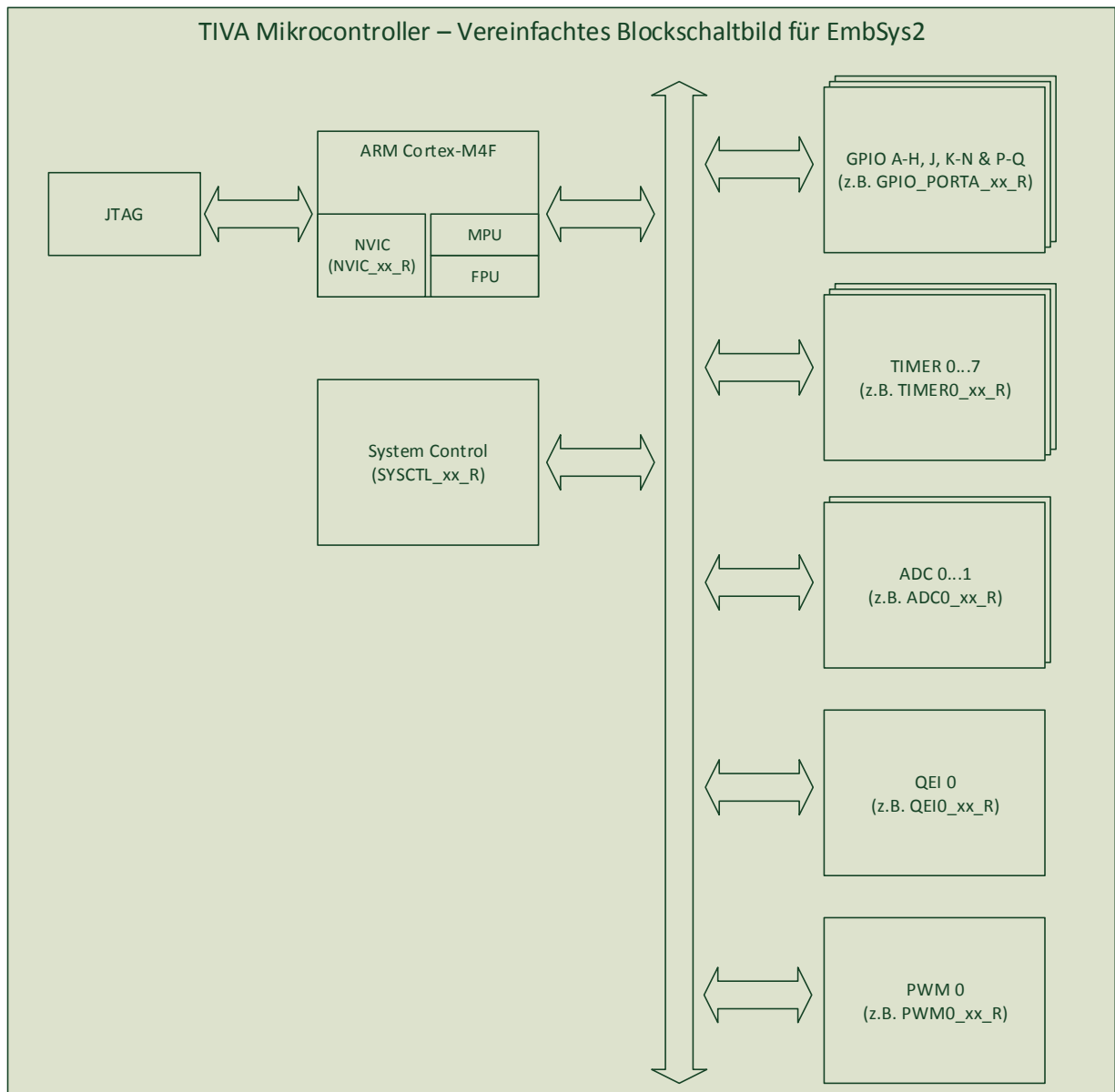


Abbildung 1 TIVA Mikrocontroller - Vereinfachtes Blockschaftbild



2 Registerbezeichnungen beim TM4C1294NCPDT

Der Aufbau der Registerbezeichnungen im Code Composer Studio ist wie folgt:

2.1 Beispiel Direction Register of PortA

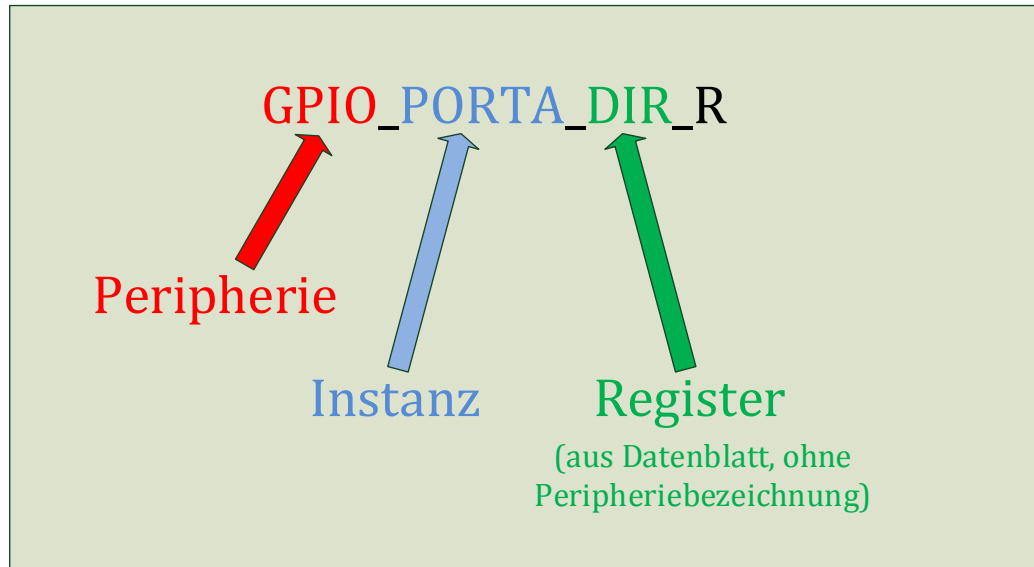


Abbildung 2 Registerbezeichnungen - TM4C1294ncpd.t.h

3 Detaillierter Aufbau

3.1 Beispiel GPIO - PortA

3.1.1 Struktur

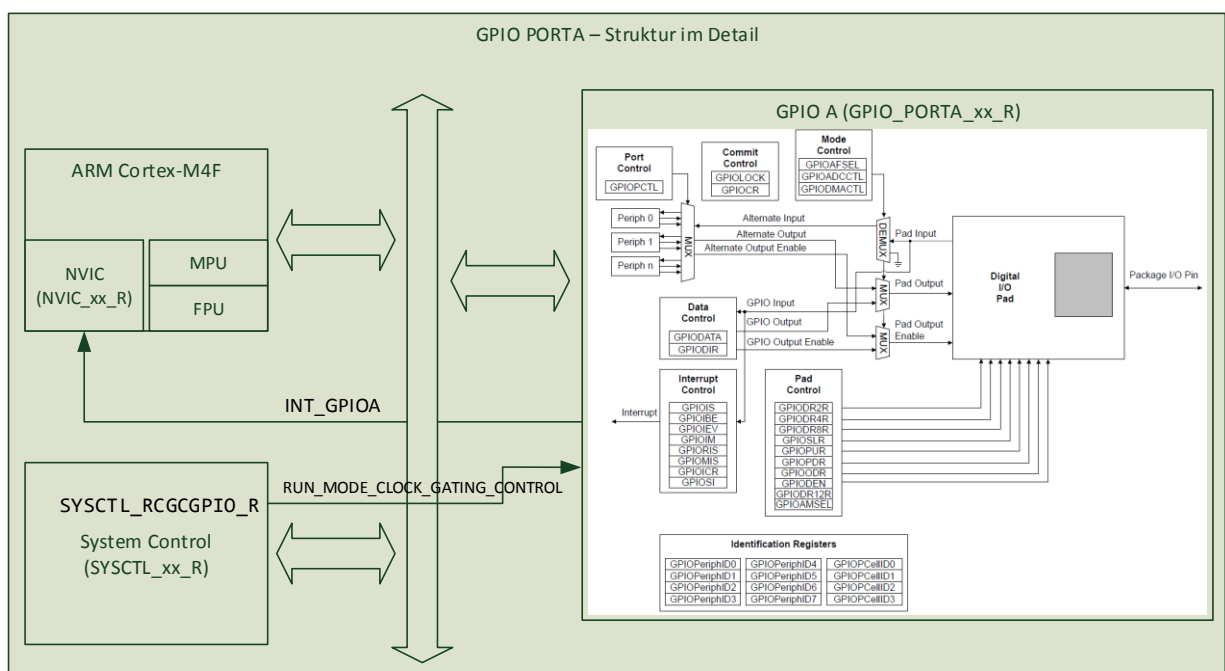


Abbildung 3 GPIO - im Detail



3.1.2 Abfolge Registerkonfiguration

Wichtige Schritte und dessen Abfolge:

1. Run Mode Clock Gating Control (**SYSCTL_RCGCGPIO_R**) für den Port Clock setzten. Dadurch wird der Clock zur Peripherie durchgeschaltet und die Register sind nun les-/schreibbar
2. Festlegen der Richtung (**GPIO_PORTA_DIR_R**). Eine „1“ ist ein Ausgang, eine „0“ ein Eingang (default).
3. Modus festlegen (**GPIO_PORTA_AFSEL_R**) für das Digital I/O Pad
 - a. Falls Pin für eine andere Peripherie benutzt werden soll, noch den Port Control Mux konfigurieren (**GPIO_PORTA_PCTL_R**)
4. Evtl. Pad konfigurieren gemäss Tabelle „GPIO Pad Configuration Examples“ im User Manual, meist genügen die default Einstellungen
 - a. Stromstärke (**GPIO_PORTA_DR2R_R**, **GPIO_PORTA_DR4R_R**, **GPIO_PORTA_DR8R_R**)
 - b. Pullup-Widerstand (**GPIO_PORTA_PUR_R**)
 - c. Pulldown-Widerstand (**GPIO_PORTA_PDR_R**)
 - d. Open Drain-Ausgang (**GPIO_PORTA_ODR_R**)
 - e. Slew rate (**GPIO_PORTA_SLR_R**)
5. Um als digital I/O zu verwenden, Bit im digital Enable Register setzten (**GPIO_PORTA_DEN_R**)
6. Falls Interrupt-Funktionalität benötigt wird
 - a. Interrupt-Sense (**GPIO_PORTA_IS_R**)
 - b. Falls beide Flanken auszuwerden sind (**GPIO_PORTA_IBE_R**)
 - c. Interrupt-Event (**GPIO_PORTA_EV_R**)
 - d. Interrupt-Maske (**GPIO_PORTA_IM_R**)
 - e. Interrupt der Peripherie beim NVIC aktivieren (**NVIC_ENn_R**)

3.2 Beispiel ADC – ADC0

3.2.1 Abfolge Registerkonfiguration

Wichtige Schritte und dessen Abfolge:

Dazugehörige GPIO – Peripherie konfigurieren (gemäss 3.1.2 Abfolge Registerkonfiguration)

- Speziell:
 - o Digital I/O Bit löschen (**GPIO_PORTx_DEN_R**)
 - o Alternative Funktion selektieren (**GPIO_PORTx_AFSEL_R**)
 - o Ausschalten der analogen Isolation (**GPIO_PORTx_AMSEL**) mittels eines „1“

ADC – Peripherie konfigurieren

1. Run Mode Clock Gating Control (**SYSCTL_RCGCADC_R**) für den ADC Clock setzten. Dadurch wird der Clock zur Peripherie durchgeschaltet und die Register sind nun les-/schreibbar
2. Festlegen der Sample-Sequenz Prioritäten (**ADC0_SSPRI_R**)
3. Sample Sequenz konfigurieren
 - a. Sequenz deaktivieren (**ADC0_TSSEL_R**)
 - b. Trigger Event festlegen (**ADC0_EMUX_R**)
 - c. Für jedes Sample der Sequenz die Eingangsquelle festlegen (**ADC0_SSMUXn_R** & **ADC0_SSEMUXn_R**)
 - d. Für jedes Sample der Sequenz die Kontrolle festlegen (beim letzten zusätzlich Ende markieren) (**ADC0_SSCTLn_R**)
4. Falls benötigt Interrupt konfigurieren
 - a. Interrupt-Maske festlegen (**ADC0_IM_R**)
 - b. Interrupt der Peripherie beim NVIC aktivieren (**NVIC_ENn_R**)
5. Logik der Sample Sequenz aktivieren (**ADC0_ACTSS_R**)