

RTS6: Conception et programmation de Systèmes Embarqués

cours 3: périphériques et radio CC2500

Antoine Fraboulet, Fabrice Jumel, Lionel Morel, Tanguy Risset
antoine.fraboulet@insa-lyon.fr
Lab CITI, INSA de Lyon



- p. 1/48



TI eZ430-RF2500



TI eZ430-RF2500

● TI eZ430-RF2500

● Schématique du module

● Présentation

● Schémas

Schéma FET

Brochage et E/S

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

E/S analogiques

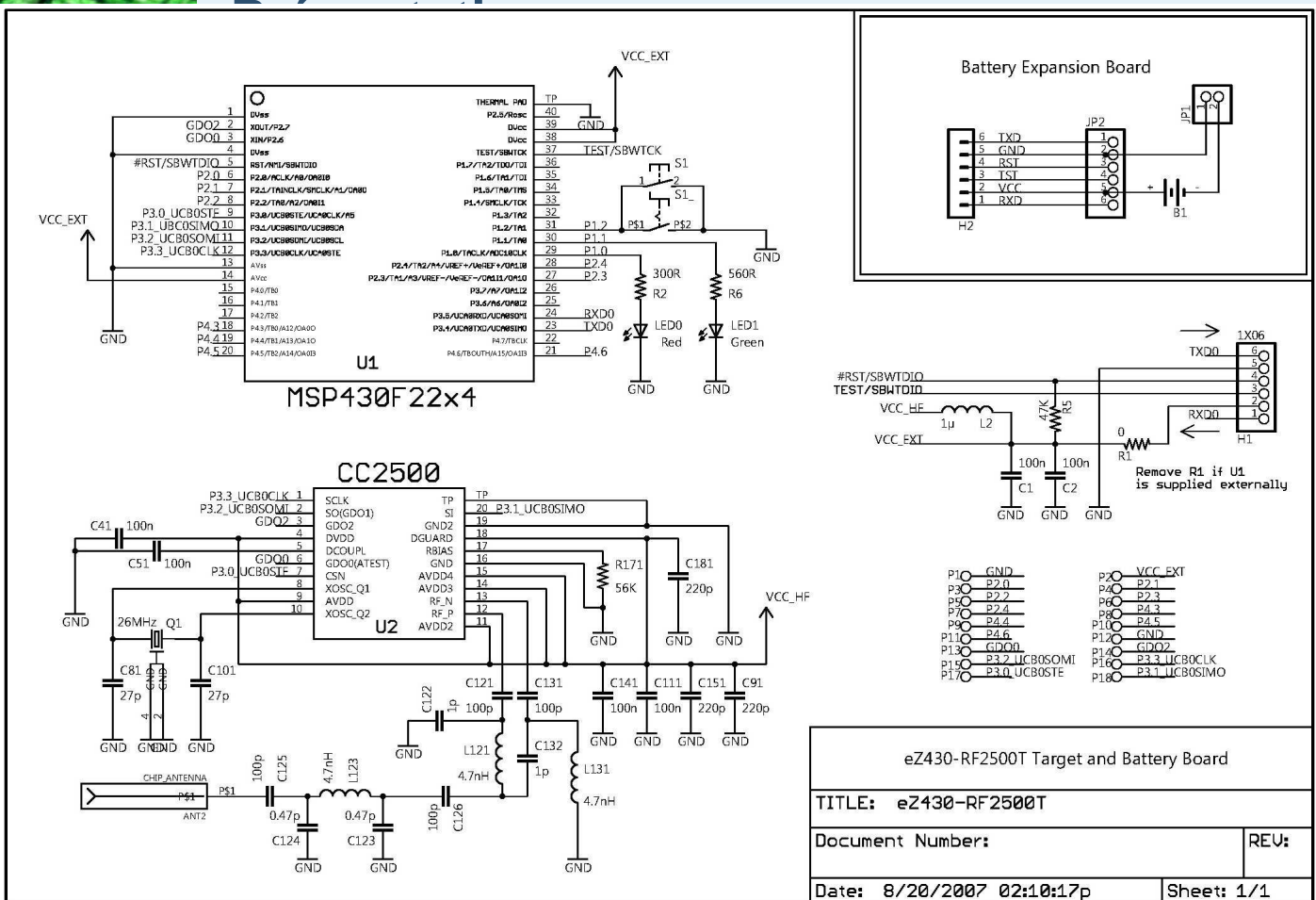
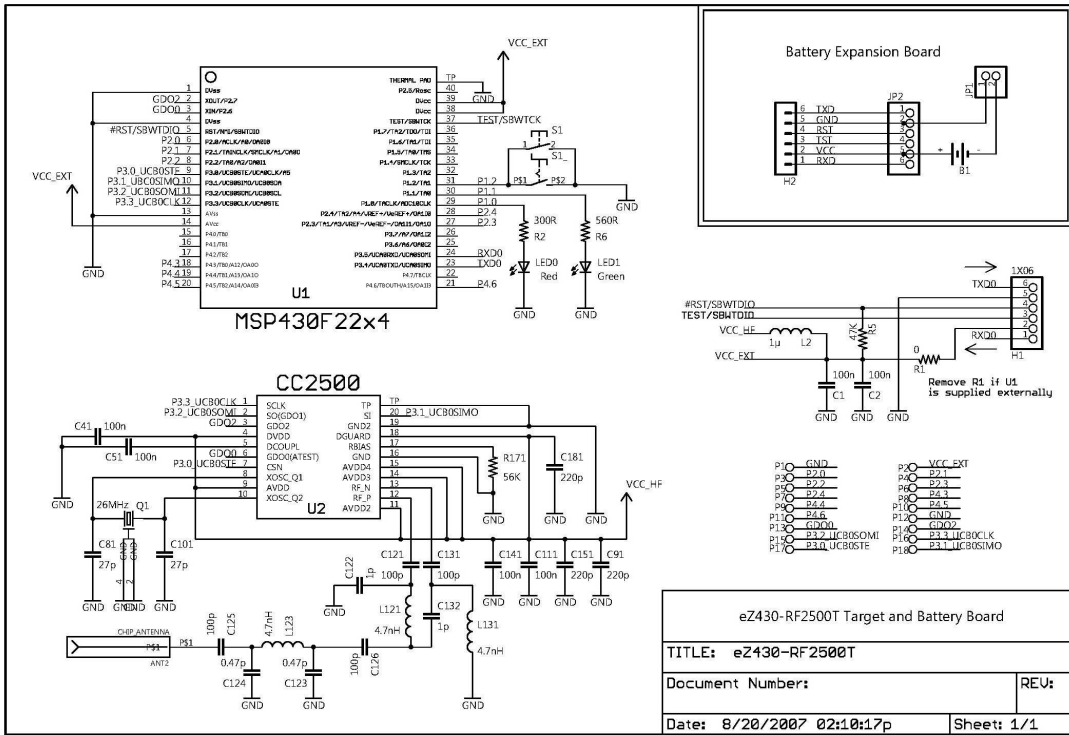
Pilotes de périphériques

- p. 2/48

Schématique du module

Module externe à détacher.

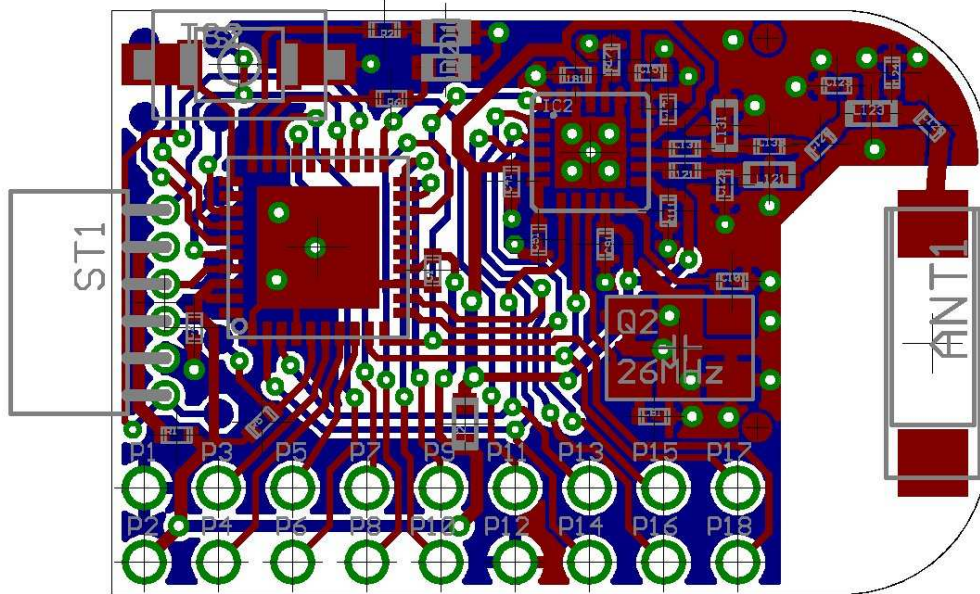
- TI eZ430-RF2500
 - TI eZ430-RF2500
 - Schématique du module
 - Présentation
 - Schémas
-
- Schéma FET
-
- Brochage et E/S
-
- Exemple
-
- Introduction
-
- Gestion du temps
-
- Interruptions
-
- Timers
-
- Ports série
-
- E/S analogiques
-
- Pilotes de périphériques



Schémas

- TI eZ430-RF2500
- TI eZ430-RF2500
- Schématique du module
- Présentation
- Schémas**

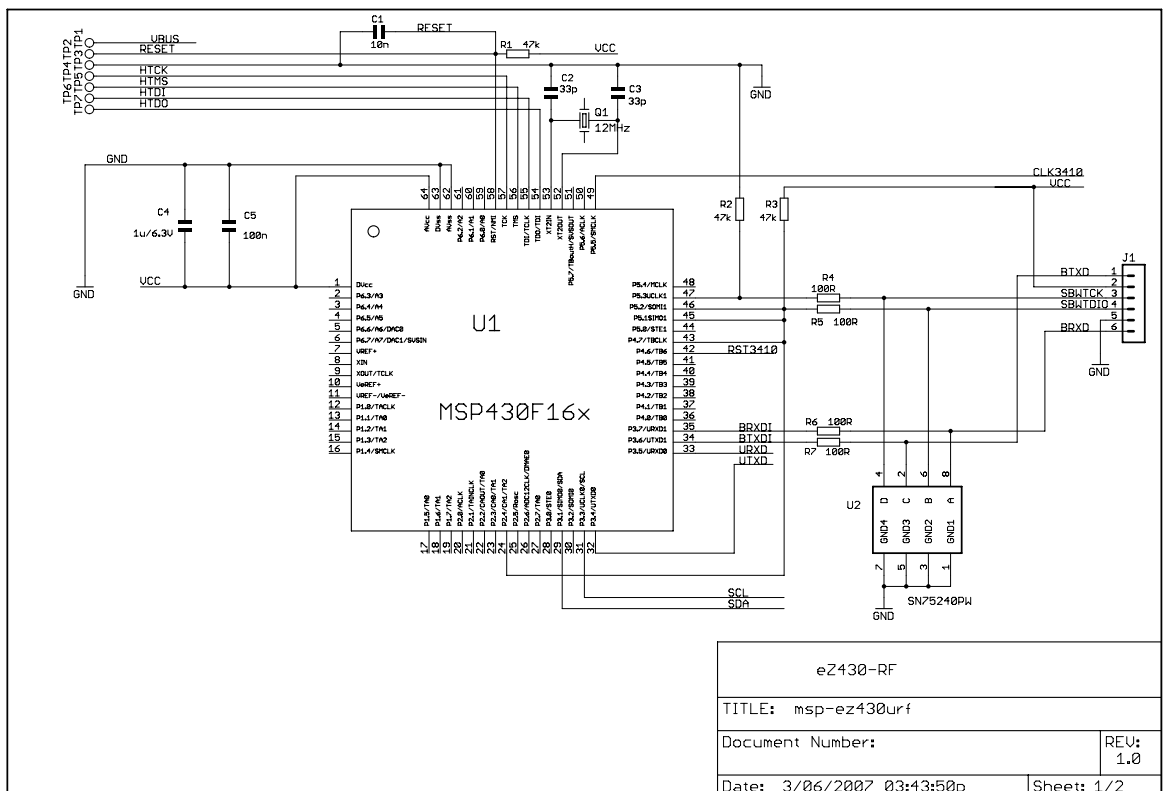
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques



Schémas partie USB (1)

- TI eZ430-RF2500
- Schéma FET
- Schémas partie USB (1)**
- Schémas partie USB (2)

- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques



Schémas partie USB (2)

TI eZ430-RF2500

Schéma FET

● Schémas partie USB (1)

● Schémas partie USB (2)

Brochage et E/S

Exemple

Introduction

Gestion du temps

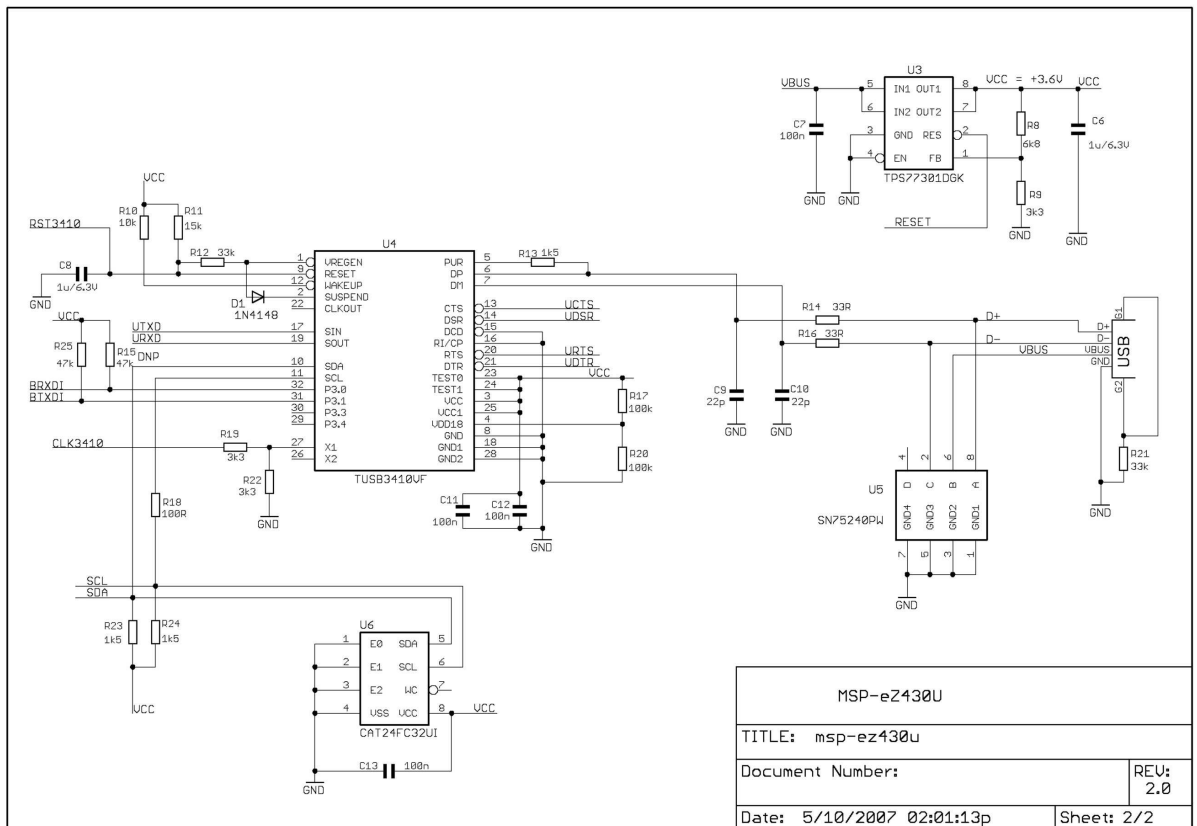
Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques



Brochage du MSP430f2274

TI eZ430-RF2500

Schéma FET

Brochage et E/S

● Brochage du MSP430f2274

● Pilotage des Entrées / Sorties

● Mapping mémoire des MSP430

● Exemple de port (MSP430f149)

● Exemple de port (MSP430f149)

Exemple

Introduction

Gestion du temps

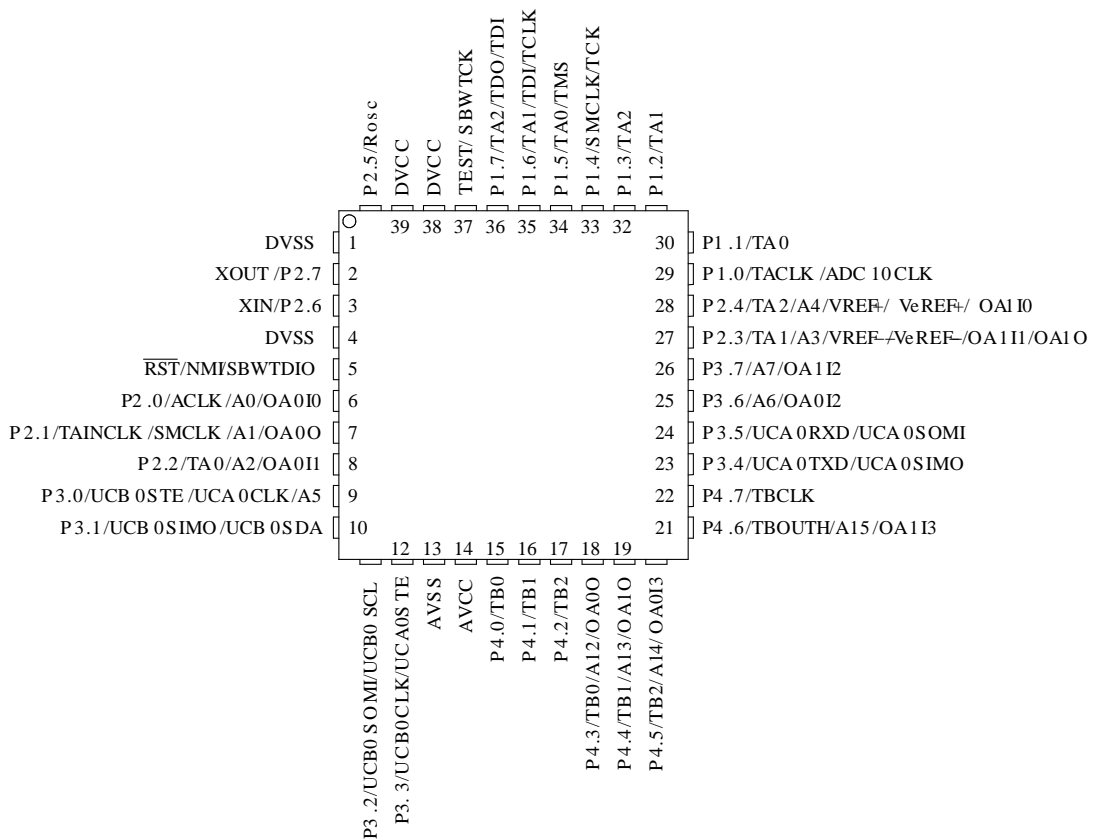
Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques



Pilotage des Entrées / Sorties

TI eZ430-RF2500

Schéma FET

Brochage et E/S

● Brochage du MSP430f2274

● **Pilotage des Entrées / Sorties**

● Mapping mémoire des

MSP430

● Exemple de port
(MSP430f149)

● Exemple de port
(MSP430f149)

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Les broches sont multiplexées
 - ◆ Un port peut être utilisé comme I/O générique
 - ◆ Il peut correspondre à une fonction (bloc de périphérique)
- Programmation logicielle
 - ◆ PxSEL : sélection IO / périphérique
 - ◆ PxIN : registre des données en lecture
 - ◆ PxOUT : registre des données en écriture
 - ◆ PxDIR : direction du port
 - ◆ Les blocs sont pilotés à des adresses fixes.

- p. 9/48

Mapping mémoire des MSP430

TI eZ430-RF2500

Schéma FET

Brochage et E/S

● Brochage du MSP430f2274

● Pilotage des Entrées / Sorties

● **Mapping mémoire des**

MSP430

● Exemple de port
(MSP430f149)

● Exemple de port
(MSP430f149)

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Accès aux registres
- Chaque registre est mappé en mémoire :

```
volatile unsigned char P5OUT asm("0x0031");  
P5OUT = 0x13;
```

```
char *p5 = (char*)0x0031;  
*p5 = 0x13;
```

- p. 10/48

Exemple de port (MSP430f149)

TI eZ430-RF2500

Schéma FET

Brochage et E/S

- Brochage du MSP430f2274
- Pilotage des Entrées / Sorties
- Mapping mémoire des MSP430
- Exemple de port (MSP430f149)
- Exemple de port (MSP430f149)

Exemple

Introduction

Gestion du temps

Interruptions

Timers

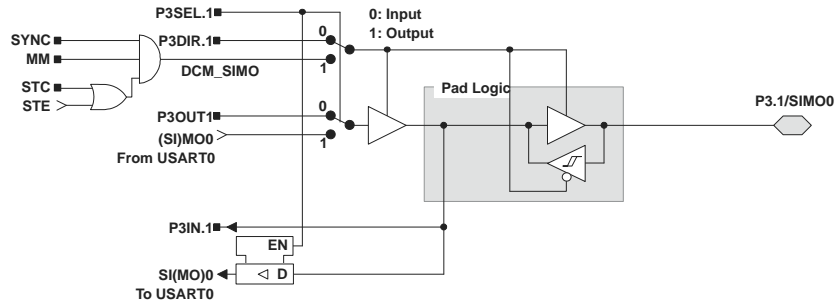
Ports série

E/S analogiques

Pilotes de périphériques

input/output schematic (continued)

port P3, P3.1, input/output with Schmitt-trigger



- p. 11/48

Exemple de port (MSP430f149)

TI eZ430-RF2500

Schéma FET

Brochage et E/S

- Brochage du MSP430f2274
- Pilotage des Entrées / Sorties
- Mapping mémoire des MSP430
- Exemple de port (MSP430f149)
- Exemple de port (MSP430f149)

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

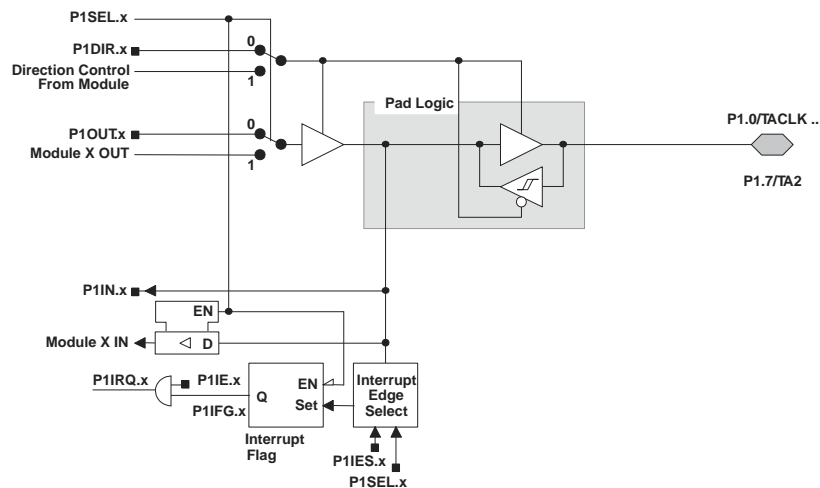
E/S analogiques

Pilotes de périphériques

- Ports 1 et 2 avec interruptions
- Registres PxIFG (flag) et PxIE (interrupt enable)

input/output schematic

port P1, P1.0 to P1.7, input/output with Schmitt-trigger



- p. 12/48

Exemple simple

```
int main(void)
{
    unsigned char b;

    // Direction = 0:input 1:output
    // Selector = 0:GPIO 1:peripheral
    P1DIR |= (BIT_GREEN | BIT_RED);
    P1SEL &= ~(BIT_GREEN | BIT_RED);
    P1OUT &= ~(BIT_GREEN | BIT_RED);

    b = 0x01;
    while (0)
    {
        wait(50000); wait(50000);
        wait(50000); wait(50000);

        P1OUT = b;
        b <<= 1;
        if (b == 0x4) b = 0x01;
    }
    return 0;
}
```

```
#include <io.h>

#define BIT_GREEN (1 << 1)
#define BIT_RED (1 << 0)

void wait(unsigned int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        asm(" nop; \n");
        asm(" nop; \n");
    }
}
```

- p. 13/48

Présentation des périphériques

Le module eZ430-RF2500 est équipé d'un msp430f2274. Ces modèles de msp ont les propriétés suivantes:

- Fréquence interne jusqu'à 16MHz
- TimerA avec 3 registres 16 bits
- TimerB avec 3 registres 16 bits
- USART avec UART + SPI + I2C
- Convertisseurs 10 bits

- p. 14/48

Brochage du MSP430f2274

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
 - Présentation des périphériques
 - Brochage du MSP430f2274
 - Schéma bloc du MSP430f2274
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques

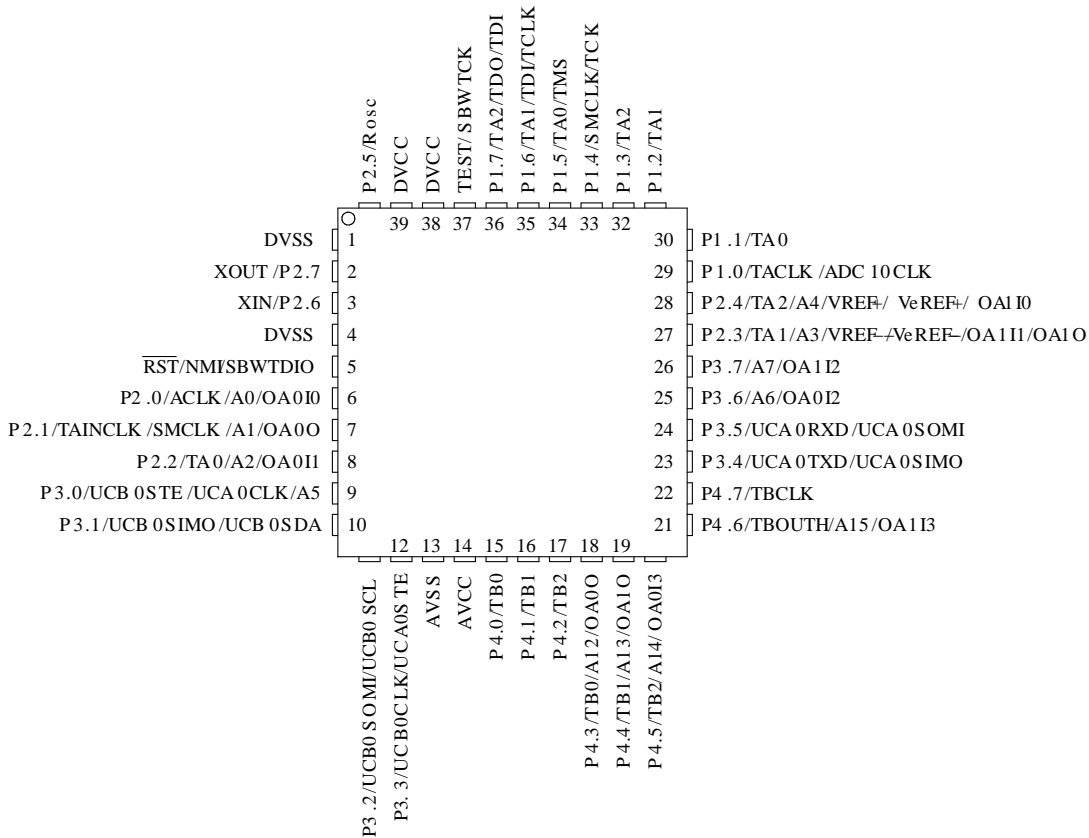
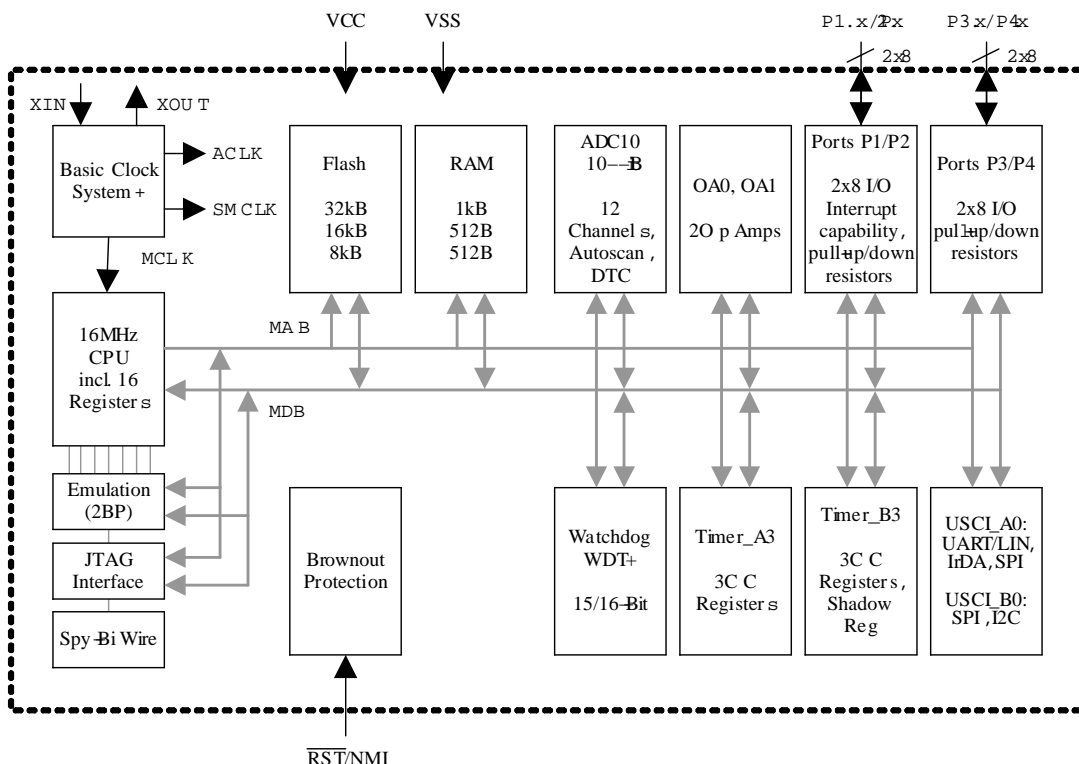


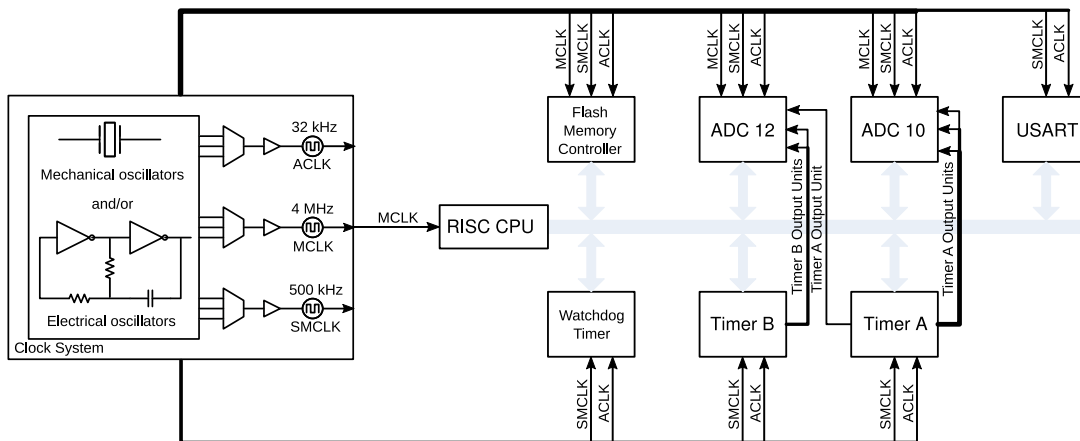
Schéma bloc du MSP430f2274

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
 - Présentation des périphériques
 - Brochage du MSP430f2274
 - Schéma bloc du MSP430f2274
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques



Gestion des horloges

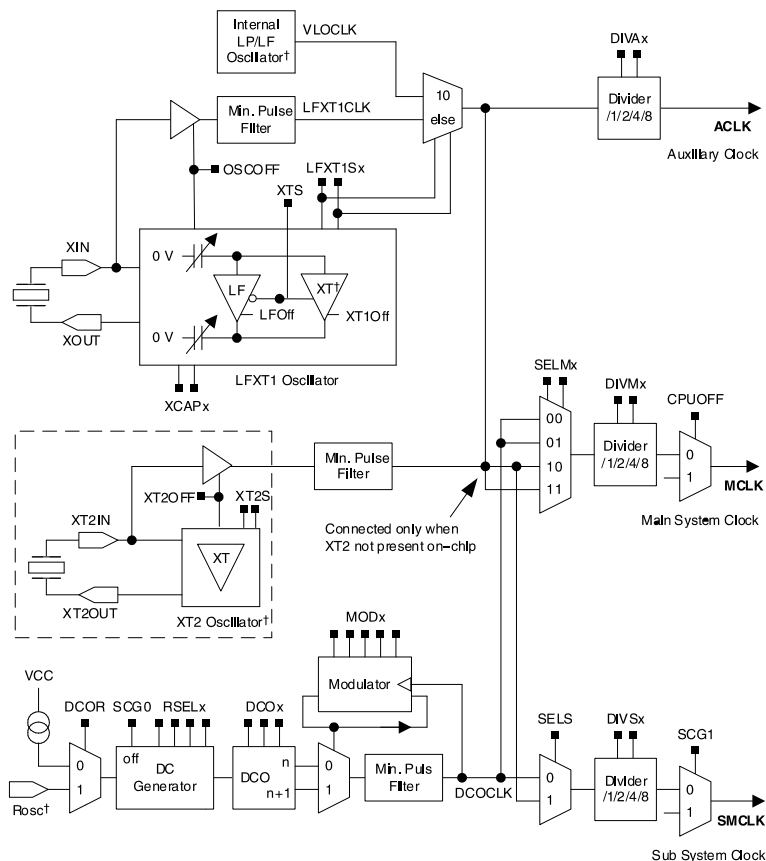
- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
 - Gestion des horloges
 - Basic Clock +
 - Réglage du DCO
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques



- Un seul bloc d'horloge est utilisé comme référence
- Les blocs doivent être paramétrés en fonction de ces références

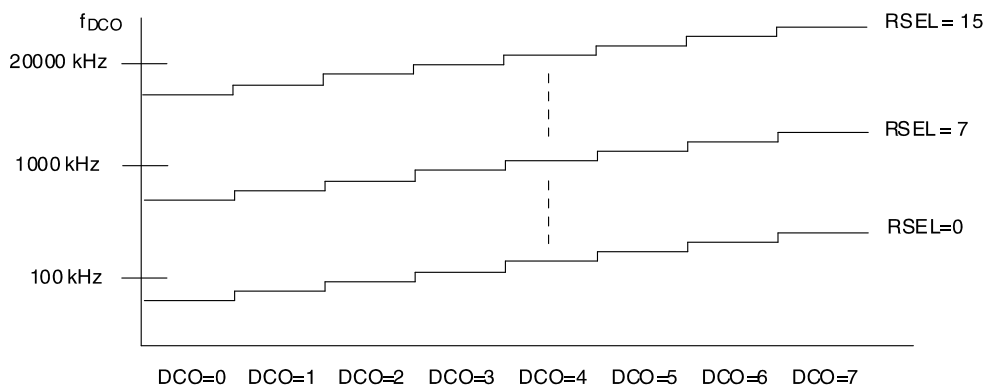
Basic Clock +

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
 - Gestion des horloges
 - Basic Clock +
 - Réglage du DCO
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques



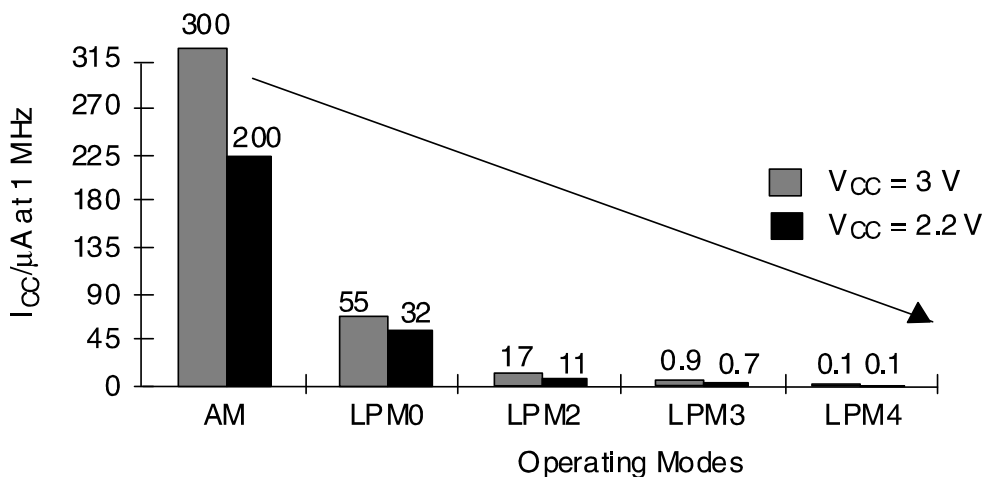
Réglage du DCO

Le bloc DCO est programmable à l'aide de son jeu de registre. Les fréquences atteignables couvrent toute la plage de 1.1MHz à 16MHz. La dérive de la fréquence est sensible.



Modes de veille

- La consommation du MSP430 est gérée grâce à des modes de veille
- Le premier mode permet de couper les instructions
- Les autres modes coupent les autres horloges



Modes de veille

TI eZ430-RF2500

Schéma FET

Brochage et E/S

Exemple

Introduction

Gestion du temps

Interruptions

● Modes de veille

● Modes de veille

● Interruptions

● Ecriture d'un gestionnaire d'interruption

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Macro LPM0,
 - ◆ Modification du bit CPUOFF du registre de contrôle
- La sortie d'un mode de sommeil est faite à la réception d'une interruption (timer)
- LPM0_EXIT modifie la valeur du bit CPUOFF dans la sauvegarde de SR sur la pile
- Désactivation partielles des périphériques du MSP
 - ◆ LPM0 : horloge MCLK principale désactivée
 - ◆ LPM1 : MCLK et DCO désactivés
 - ◆ LPM2 : MCLK, DCO, SMCLK
 - ◆ ...
- Utiliser les interruptions pour contrôler le programme
- Reste la mise en veille des périphériques

- p. 21/48

Interruptions

TI eZ430-RF2500

Schéma FET

Brochage et E/S

Exemple

Introduction

Gestion du temps

Interruptions

● Modes de veille

● Modes de veille

● Interruptions

● Ecriture d'un gestionnaire d'interruption

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Interruptions à 2 niveaux:
 - ◆ GIE dans le registre SR
 - ◆ Gestion des sources dans les blocs
- Interruptions à source uniques: l'acquittement est implicite
- Interruptions à sources multiples: acquittement nécessaire
 - ◆ Bits PxIE + PxIES + PxIFG des ports 1 et 2.
 - ◆ Comparateurs des registres des timers
 - ◆ ...
- Les interruptions sont, par défaut, non ré-entrantes. Le bit GIE est mis à 0 par le matériel à l'entrée d'un gestionnaire d'interruption.

- p. 22/48

Écriture d'un gestionnaire d'interruption

```
//interrupt (PORT1_VECTOR) port1_irq_handler(void)
#pragma vector=PORT1_VECTOR
__interrupt void
port1_irq_handler(void)
{
    if (P1IFG & (P1IE & (1 << 2)))
    {
        SWITCH_RED_LED();
    }
    P1IFG = 0;
}
```

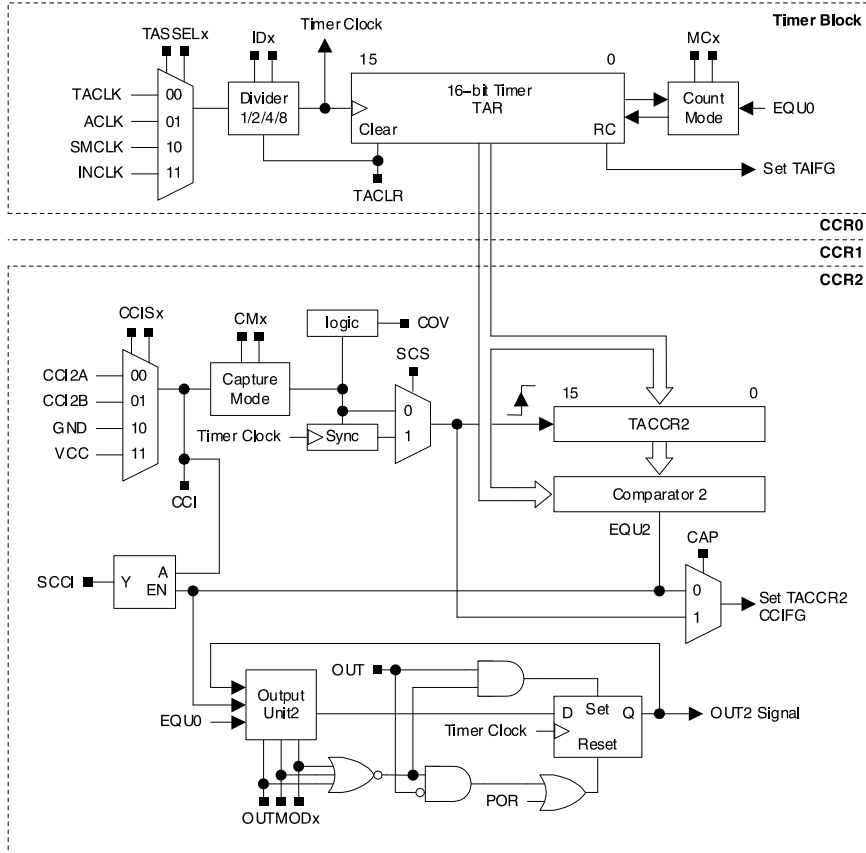
- p. 23/48

Timers

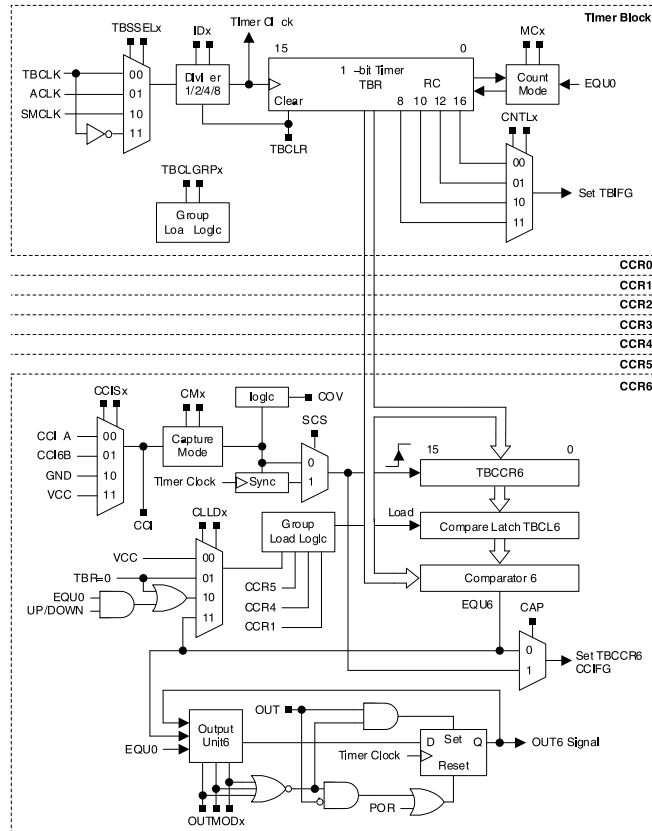
- Utilisation d'une horloge de référence par bloc
- Comparateurs multiples et gestion des interruptions
- Interruptions
 - ◆ acquittement automatique (TxCCR0)
 - ◆ acquittement logiciel (TxIV)
 - ◆ Les interruptions sont priorisées
- Les timers contiennent des registres sur 16 bits
- Les différents modes permettent d'avoir une gestion efficace du temps

- p. 24/48

Timer A



Timer B

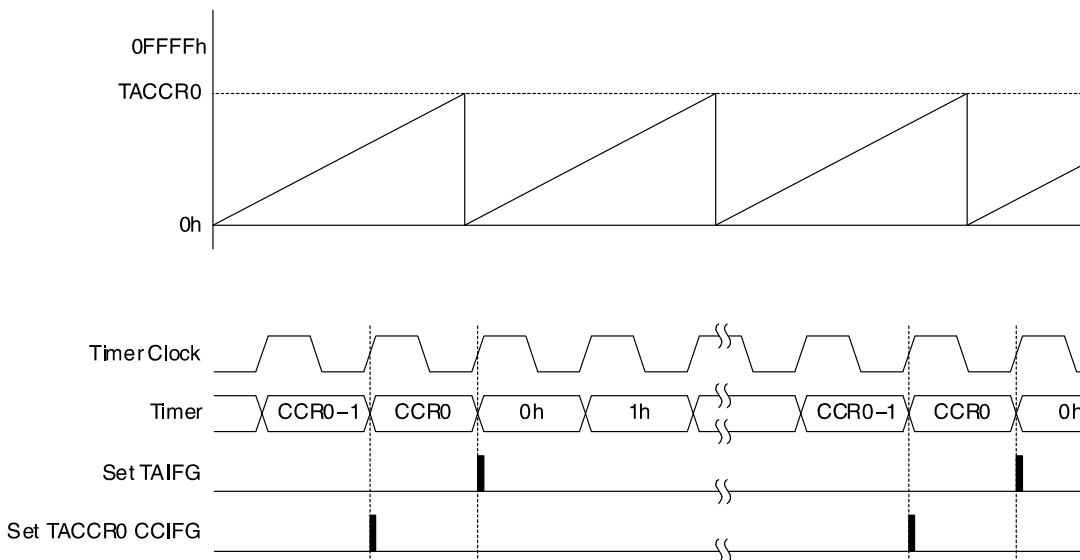


- Tl eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
 - Timers
 - **Timer A**
 - Timer B
 - Timers et modes
 - Timers et modes (2)
 - Timers et modes (3)
- Ports série
- E/S analogiques
- Pilotes de périphériques

- Tl eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
 - Timers
 - Timer A
 - **Timer B**
 - Timers et modes
 - Timers et modes (2)
 - Timers et modes (3)
- Ports série
- E/S analogiques
- Pilotes de périphériques

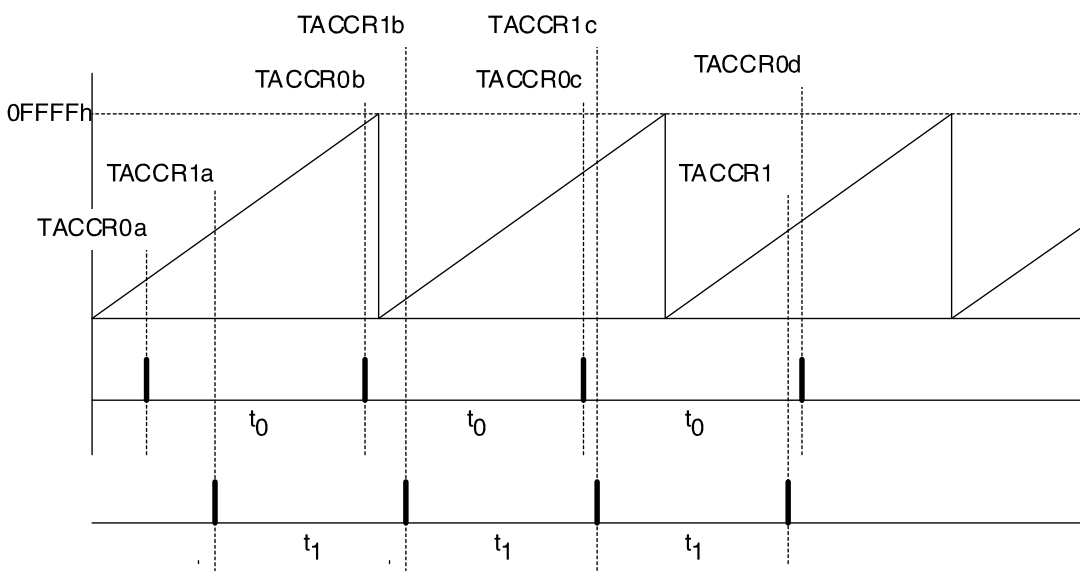
Timers et modes

■ Le mode UP est sans doute le plus utilisé.



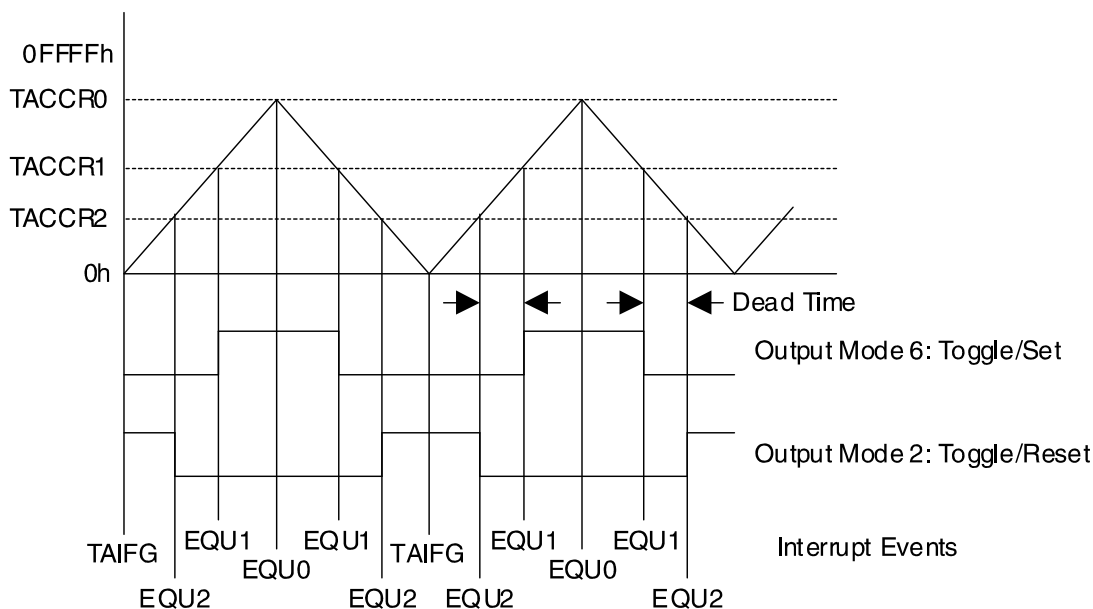
Timers et modes (2)

■ “Continuous Mode”



Timers et modes (3)

■ “Up/Down Mode”



- p. 29/48

Ports série

■ Plusieurs modèles disponibles

- ◆ Certains sont en mode exclusif
 - Modules USI (SPI, I2C)
 - Modules USART (UART, SPI, I2C)
- ◆ Ceux du 2274 sont doubles (Modules USCI)
 - USCI_Ax (UART + IrDA + SPI)
 - USCI_Bx (SPI + I2C)

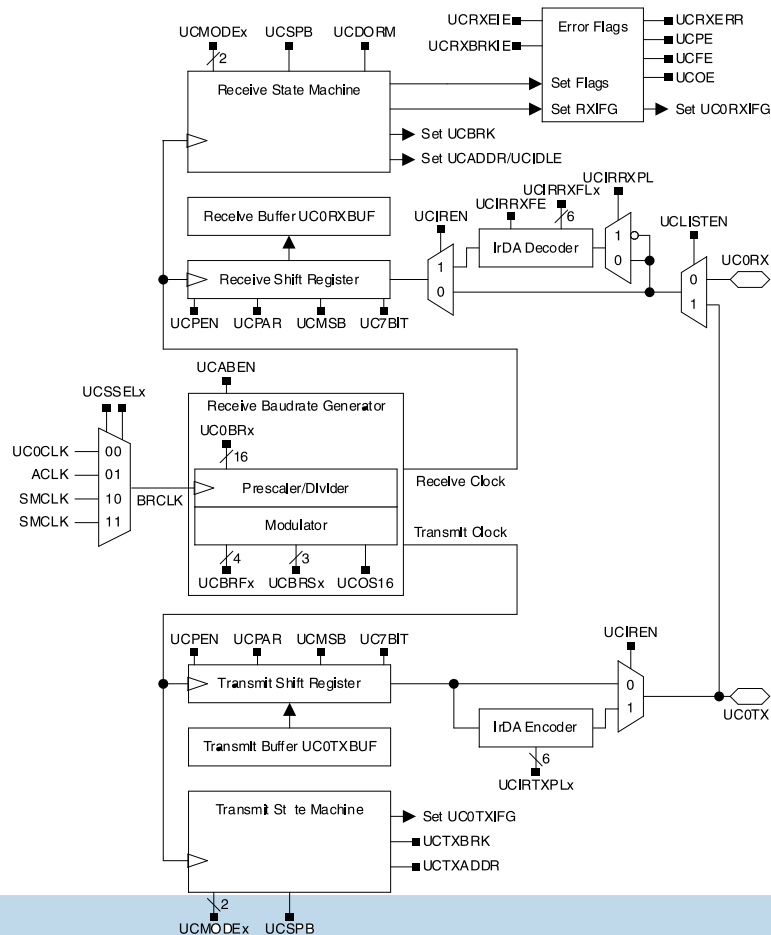
■ Les ports séries peuvent se piloter par interruption

■ Le débit est calculé à partir d'une horloge de référence

■ Le débit est dépendant des horloges externes

- p. 30/48

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
 - Ports série
 - **USCI A**
 - Ports série
- E/S analogiques
- Pilotes de périphériques

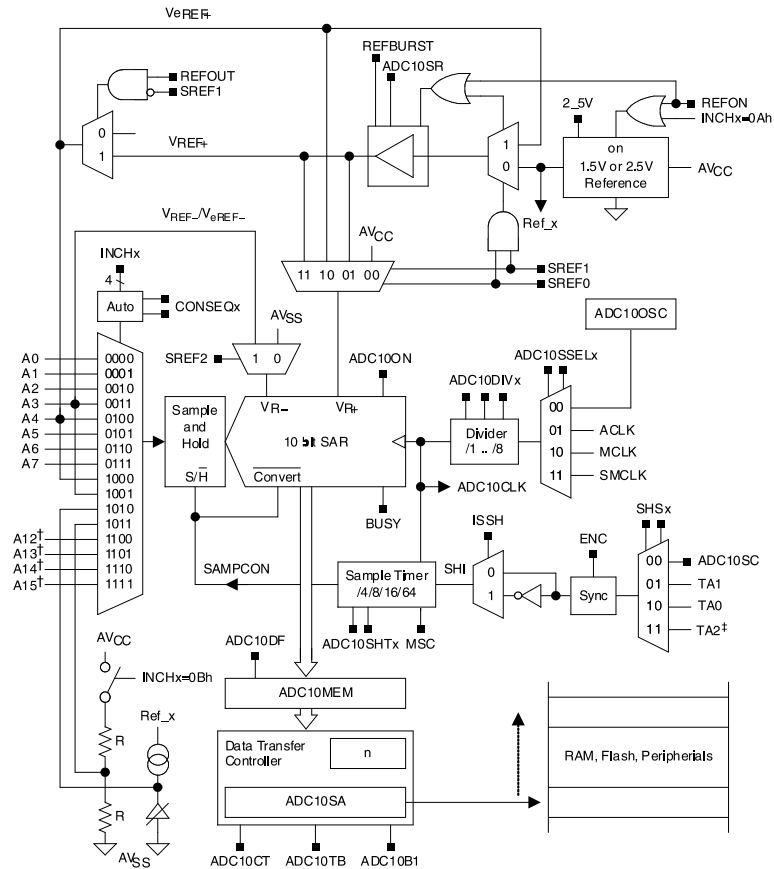


Ports série

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
 - Ports série
 - **USCI A**
 - Ports série
- E/S analogiques
- Pilotes de périphériques

- Les paramètres à utiliser pour la modulation sont à choisir dans les tables fournies par le manuel.
- La fréquence de référence ne doit pas changer.
- Le choix de la fréquence de référence est donc important.
- Si beaucoup d'erreurs apparaissent il est conseillé de passer à un code correcteur d'erreur logiciel.

ADC 10



- p. 33/48

ADC 10

- La programmation de l'ADC reste dans le modèle que pour les autres composants:
 - ◆ Choisir une fréquence d'échantillonnage par rapport à une fréquence de référence
 - ◆ Choisir un canal à numériser
 - ◆ Attendre la stabilisation
 - ◆ Demander la lecture
 - ◆ Attendre la validation
 - ◆ Possibilité de recevoir une interruption au besoin

- p. 34/48

Programmation de pilotes

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

- Un pilote de périphérique doit utiliser les ressources de la plateforme.
- Une attention spéciale doit être prise pour ne pas trop simplifier le matériel.
- Une configuration *globale* de la plateforme est obligatoire pour tirer le meilleur de l'application.
- Les données des applications peuvent être utilisés pour simplifier les sources et réduire la complexité du code.

- p. 35/48

Périphériques et schématique

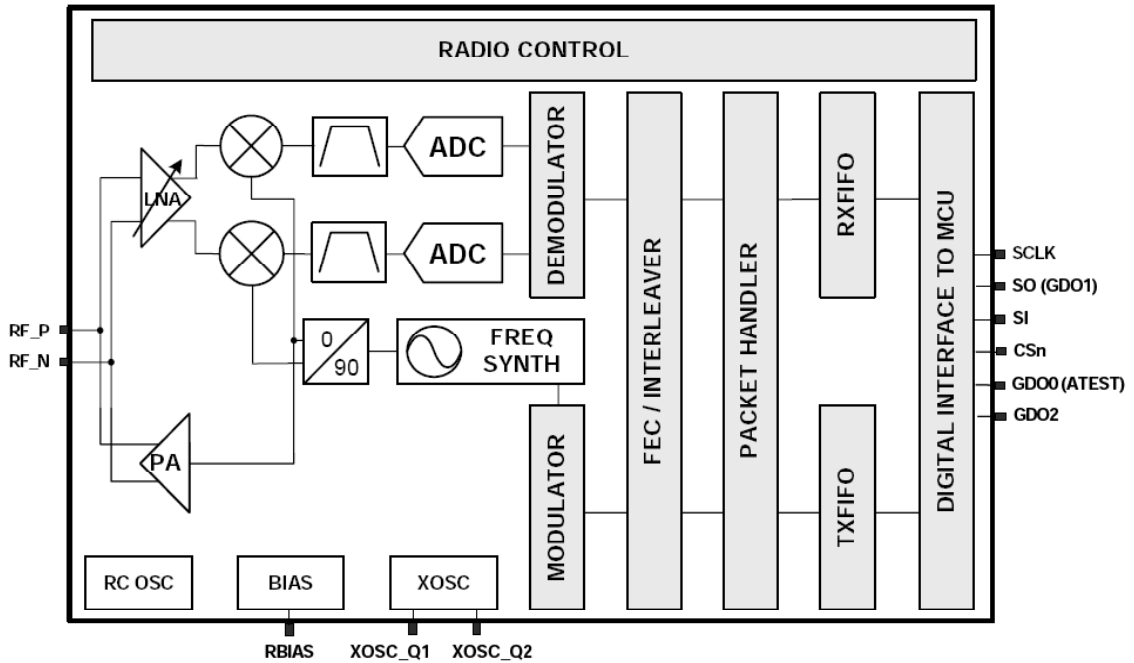
- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

1. La schématique n'est que le point de départ.
 - connexions entre les composants
 - low level (bit) communications
2. Il faut manipuler également le comportement des composants.

- p. 36/48

Schéma du CC2500

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)



SmartRF

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

États du CC2500

- États
- Transitions
- Temps / latences
- Consommation

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

Machine d'état du CC2500

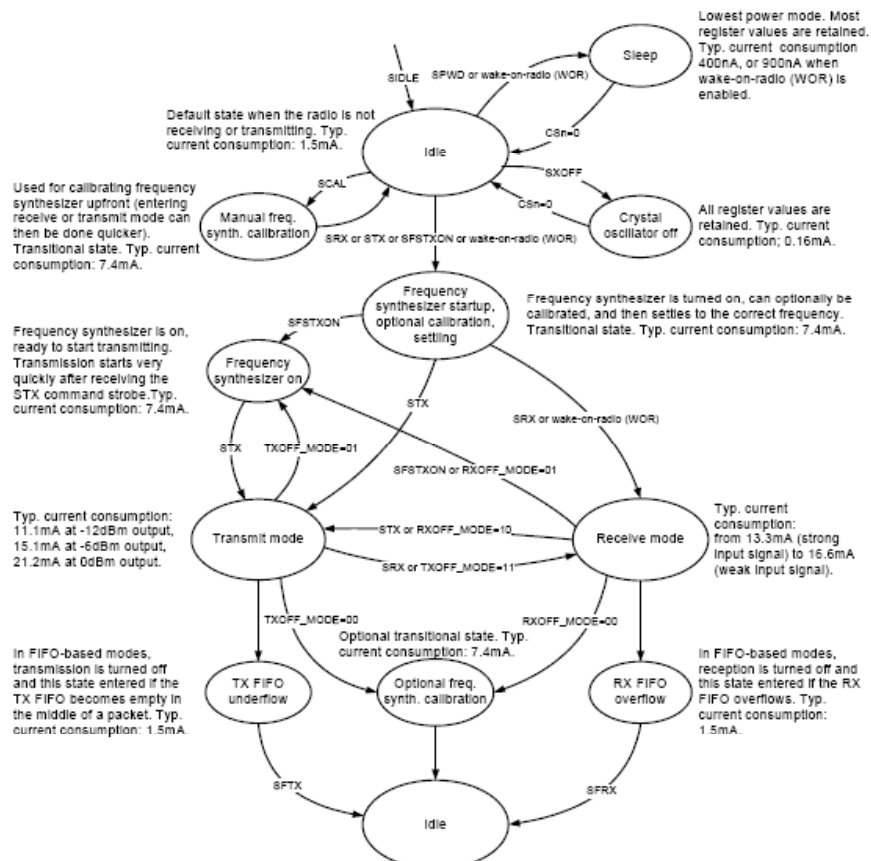
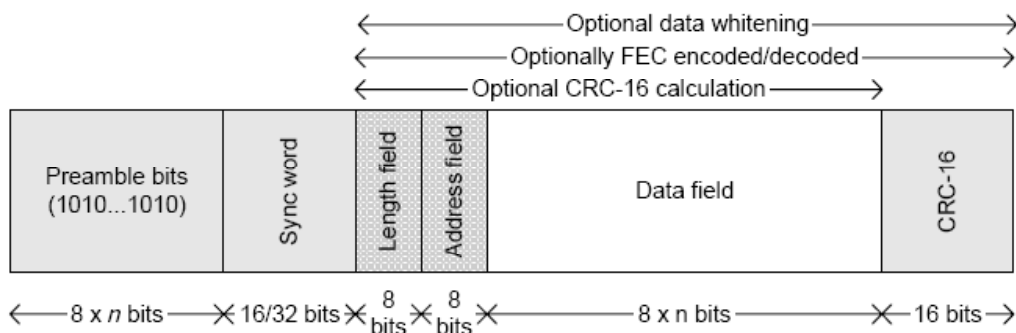


Figure 5. Simplified state diagram, with typical usage and current consumptions at 250 kHz

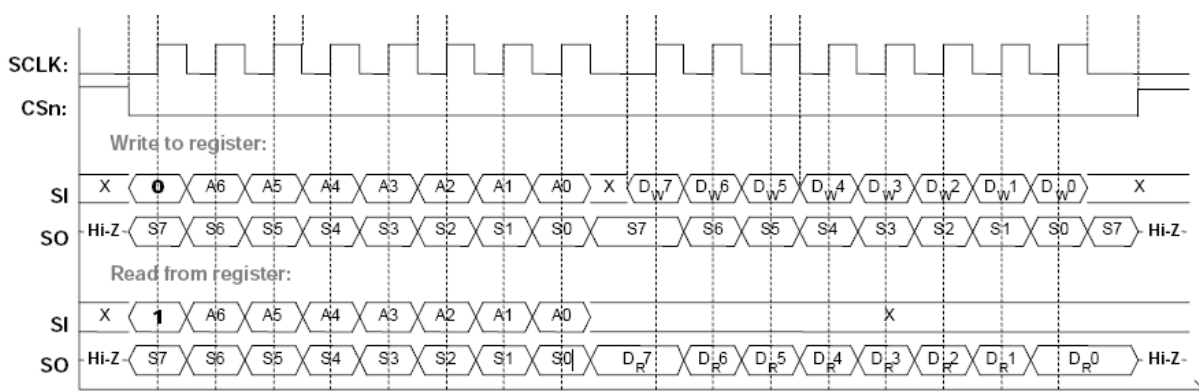
format du paquet du CC2500



Legend:

- Inserted automatically processed and not
- Optional user-processed but not
- Unprocessed user and/or whitening

Programmation du CC2500 par port SPI



Command Strobe

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - **command Strobe**
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

Address	Strobe Name	Description
0x30	SRES	Reset chip.
0x31	SFSTXON	Enable and calibrate frequency synthesizer (if <code>MCSM0</code> , <code>FS_AUTOCAL=1</code>). If in RX (with CCA): Go to a wait state where only the synthesizer is running (for quick RX / TX turnaround).
0x32	SXOFF	Turn off crystal oscillator.
0x33	SCAL	Calibrate frequency synthesizer and turn it off (enables quick start). SCAL can be strobed in IDLE state without setting manual calibration mode (<code>MCSM0</code> , <code>FS_AUTOCAL=0</code>)
0x34	SRX	Enable RX. Perform calibration first if coming from IDLE and <code>MCSM0</code> , <code>FS_AUTOCAL=1</code> .
0x35	STX	In IDLE state: Enable TX. Perform calibration first if <code>MCSM0</code> , <code>FS_AUTOCAL=1</code> . If in RX state and CCA is enabled: Only go to TX if channel is clear.
0x36	SIDLE	Exit RX / TX, turn off frequency synthesizer and exit Wake-On-Radio mode if applicable.
0x38	SWOR	Start automatic RX polling sequence (Wake-on-Radio) as described in Section 19.5.
0x39	SPWD	Enter power down mode when <code>CS1</code> goes high.
0x3A	SFRX	Flush the RX FIFO buffer. Only issue in IDLE, <code>TXFIFO_UNDERFLOW</code> or <code>RXFIFO_OVERFLOW</code> states.
0x3B	SFTX	Flush the TX FIFO buffer. Only issue in IDLE, <code>TXFIFO_UNDERFLOW</code> or <code>RXFIFO_OVERFLOW</code> states.
0x3C	SWORRST	Reset real time clock.
0x3D	SNOP	No operation. May be used to pad strobe commands to two bytes for simpler software.

Écriture de pilotes

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - **command Strobe**
 - **Écriture de pilotes**
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

L'écriture d'un pilote doit suivre les principes suivants:

- Contrôle de la machine à état depuis le logiciel.
 - Utilisation de toutes les fonctionnalités matérielles.
 - Fournir une abstraction aux autres parties de l'application.
- Envoyer un paquet sur la radio ne doit pas être plus compliqué qu'un appel de fonction.

Pilote d'émission en attente

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

```
/* pkt < 64 bytes, wait EOP */
void cc1100_utx(char *buffer, uint8_t length)
{
    cc1100_tx_error = 0;
    cc1100_check_fifo_xflow_flush();
    /* Fill tx fifo */
    CC1100_SPI_TX_FIFO_BYTE (length);
    CC1100_SPI_TX_FIFO_BURST (buffer, length);
    /* Send packet and wait for complete */
    cc1100_gdo2_set_signal(CC1100_GDOx_SYNC_WORD);
    CC1100_HW_GDO2_DINT();
    CC1100_SPI_STROBE(CC1100_STROBE_STX);
    while (! ( CC1100_HW_GDO2_READ() )); /* GDO2 goes high = SYNC TX */
    while ( ( CC1100_HW_GDO2_READ() )); /* GDO2 goes low = EOP */
    CC1100_HW_GDO2_EINT();
}
```

Pilote d'émission avec intr. (1)

- TI eZ430-RF2500
- Schéma FET
- Brochage et E/S
- Exemple
- Introduction
- Gestion du temps
- Interruptions
- Timers
- Ports série
- E/S analogiques
- Pilotes de périphériques
 - Programmation de pilotes
 - Périphériques et schématique
 - schéma bloc
 - smartRF
 - États du CC2500
 - state machine
 - paquet
 - SPI
 - command Strobe
 - Écriture de pilotes
 - Pilote d'émission en attente
 - Pilote d'émission avec intr. (1)

```
void cc1100_tx(char *buffer, uint8_t length)
{
    uint8_t txbytes; /* bytes free in the fifo */
    uint8_t tosend;

    cc1100_check_fifo_xflow_flush();
    cc1100_tx_packet = buffer;
    cc1100_tx_length = length;
    cc1100_tx_offset = 0;
    cc1100_tx_ongoing = 1;
    cc1100_tx_error = 0;
    cc1100_tx_sent = 0;
    /* Fill tx fifo */
    CC1100_SPI_TX_FIFO_BYTE (length);
    TX_WRITE_BLOCK();

    /* Send packet but don't wait complete */
    CC1100_HW_GDO2_DINT();
    CC1100_HW_GDO0_DINT();
    cc1100_gdo0_set_signal(CC1100_GDOx_TX_FIFO); /* gdo0 tx fifo */
    cc1100_gdo2_set_signal(CC1100_GDOx_SYNC_WORD); /* gdo2 sync & eop */
    CC1100_HW_GDO2_IRQ_ON_DEASSERT(); /* want an irq for EOP */
    CC1100_HW_GDO0_IRQ_ON_DEASSERT(); /* want an irq on Tx < thr */
    CC1100_SPI_STROBE(CC1100_STROBE_STX); /* start */
    while (! ( CC1100_HW_GDO2_READ() )); /* GDO2 high = sync TX */
    CC1100_HW_GDO0_EINT(); /* allow irq on gdo0 */
    CC1100_HW_GDO2_EINT(); /* allow irq on gdo2 */
}
```


Pilote d'émission avec intr. (2)

Tl eZ430-RF2500

Schéma FET

Brochage et E/S

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Programmation de pilotes
- Périphériques et schématique
- schéma bloc
- smartRF
- États du CC2500
- state machine
- paquet
- SPI
- command Strobe
- Écriture de pilotes
- Pilote d'émission en attente
- Pilote d'émission avec intr. (1)

```
void cc1100_tx_pkt_data()
{
    uint8_t txbytes; /* bytes free in the fifo */
    uint8_t tosend;

    CC1100_SPI_ROREG(CC1100_REG_TXBYTES, txbytes);
    txbytes = 63 - txbytes; /* room free in Tx FIFO */
    tosend = MIN(txbytes, cc1100_tx_length);
    CC1100_SPI_TX_FIFO_BURST(cc1100_tx_packet, tosend);
    cc1100_tx_packet += tosend;
    cc1100_tx_length -= tosend;
}

/* called when done on asynchronous Tx */
void cc1100_tx_done_intr(void)
{
    if (cc1100_check_fifo_xflow_flush())
    {
        cc1100_tx_error = 1;
    }
    cc1100_tx_ongoing = 0;
    cc1100_tx_sent     = 1;

    CC1100_HW_IRQ_PACKET_ASSERT();
}
```

- p. 47/48

Pilote d'émission avec intr. (3)

Tl eZ430-RF2500

Schéma FET

Brochage et E/S

Exemple

Introduction

Gestion du temps

Interruptions

Timers

Ports série

E/S analogiques

Pilotes de périphériques

- Programmation de pilotes
- Périphériques et schématique
- schéma bloc
- smartRF
- États du CC2500
- state machine
- paquet
- SPI
- command Strobe
- Écriture de pilotes
- Pilote d'émission en attente
- Pilote d'émission avec intr. (1)

```
void cc1100_interrupt_handler(uint8_t pin)
{
    int interrupt_policy;
    if (pin == CC1100_GDO0)
        interrupt_policy = cc1100_gdo0_cfg;
    else if (pin == CC1100_GDO2)
        interrupt_policy = cc1100_gdo2_cfg;
    else
        return;

    switch (interrupt_policy)
    {
        case CC1100_GDOx_TX_FIFO: /* Tx FIFO threshold */
            cc1100_tx_pkt_data();
            break;
        case CC1100_GDOx_RX_FIFO: /* Rx FIFO threshold */
            cc1100_rx_pkt_data();
            break;
        case CC1100_GDOx_SYNC_WORD:
            if (cc1100_tx_ongoing == 1) /* Tx EOP */
                cc1100_tx_done_intr();
            else if (cc1100_rx_ongoing) /* Rx EOP */
                cc1100_rx_pkt_data();
            else /* Rx SYNC */
                cc1100_rx_pkt_start();
            break;
        case CC1100_GDOx_CHIP_RDY:
            break;
        default:
            break;
    }
}
```

- p. 48/48