



MSP430

Interface com Android via Bluetooth

Érika M. S. Tagima

Início: 11h00

Agenda

🔗 **A família 430**

🔗 MSP430 vs Arduino

🔗 Datasheet MSP430G2553

🔗 Ambiente de desenvolvimento: Energia

🔗 Ambiente de desenvolvimento: Code Composer Studio

🔗 Hands-on: Enviando dados via Bluetooth

A família 430

- Baixo consumo de energia
 - 230 μA em 1 MHz, 2.2 V (MSP430G2xx3)
 - Ideal para usar com bateria
- 16 bit RISC CPU
 - 27 instruções físicas

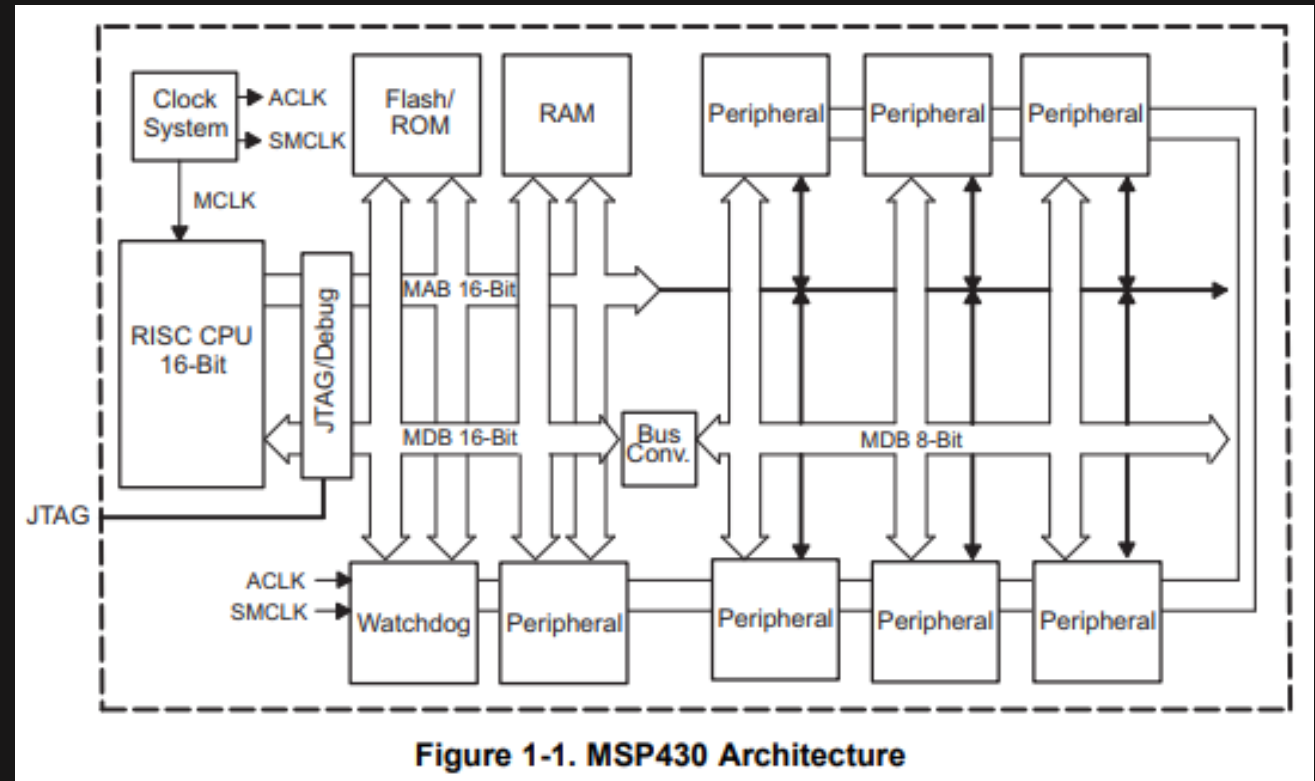
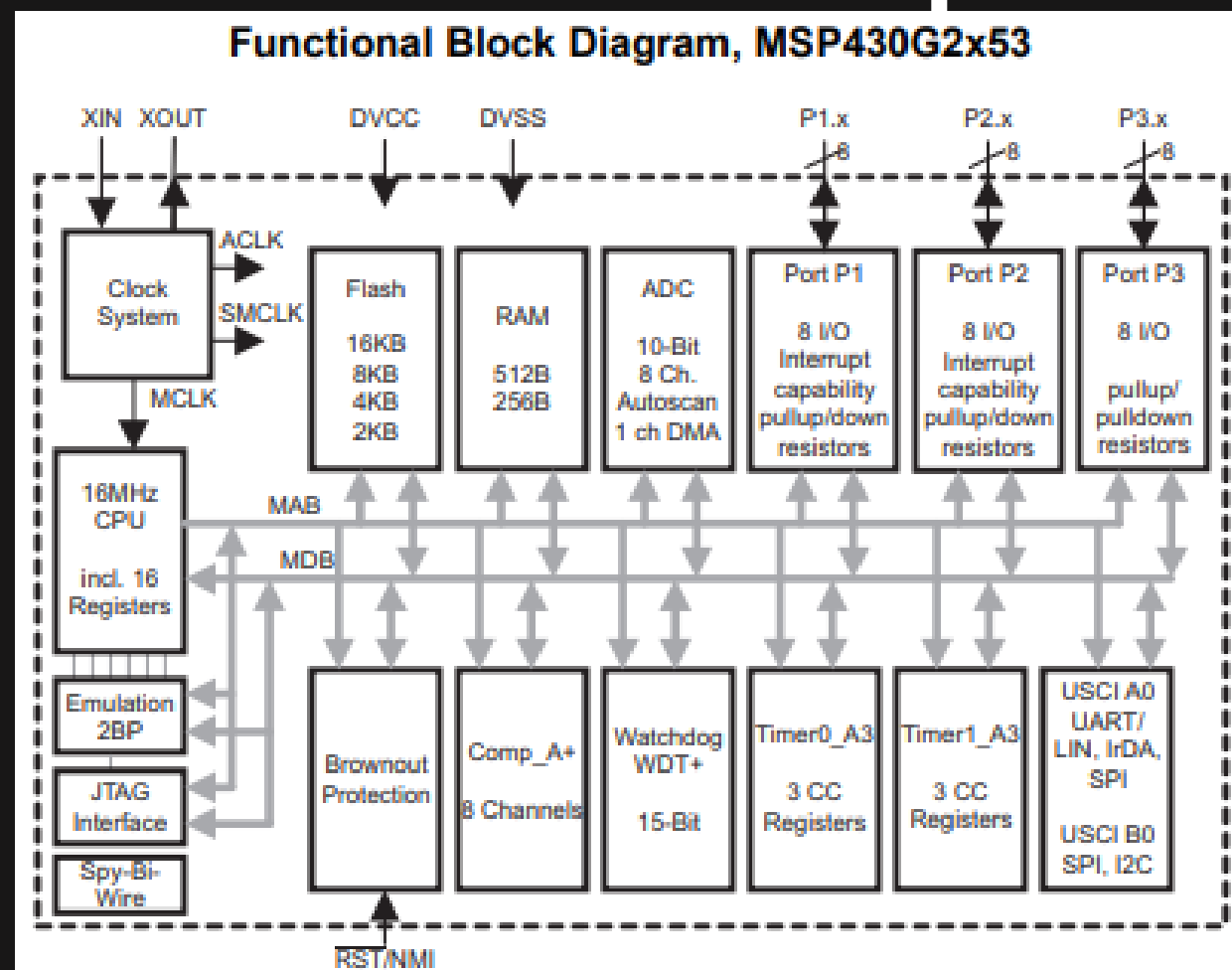


Figure 1-1. MSP430 Architecture

A família 430

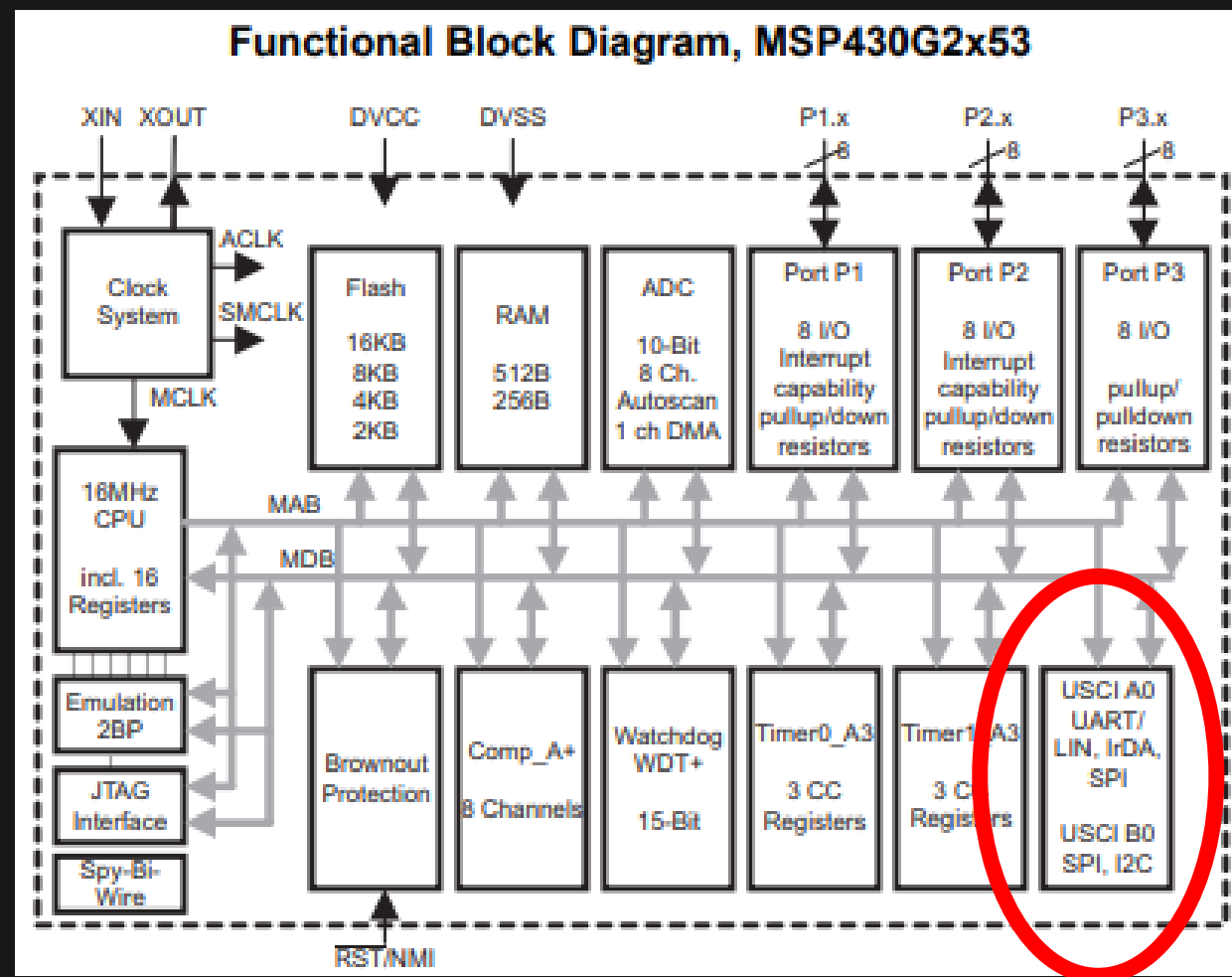
MSP430G2553



Dados retirados do documento: MSP430G2x53 e MSP430G2x13 Datasheet

<http://www.ti.com/lit/ds/symlink/msp430g2153.pdf>

A família 430



TX RX

RX

TX

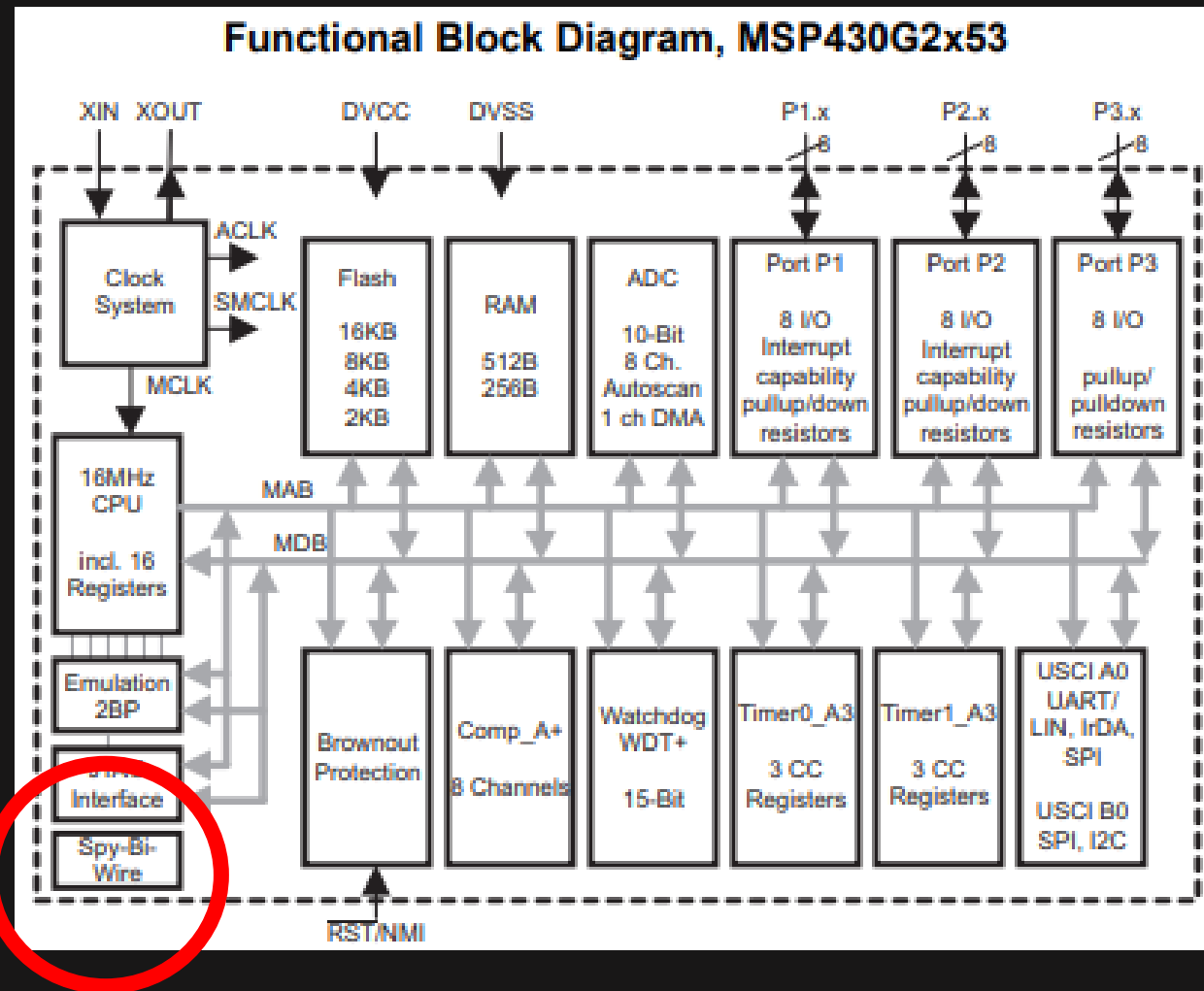
A família 430

JTAG

TMS, TCK, TDI,
TDO

Spy-Bi-Wire

TST, RST



Protocolo
desenvolvido
pela Texas
Instruments

A família 430

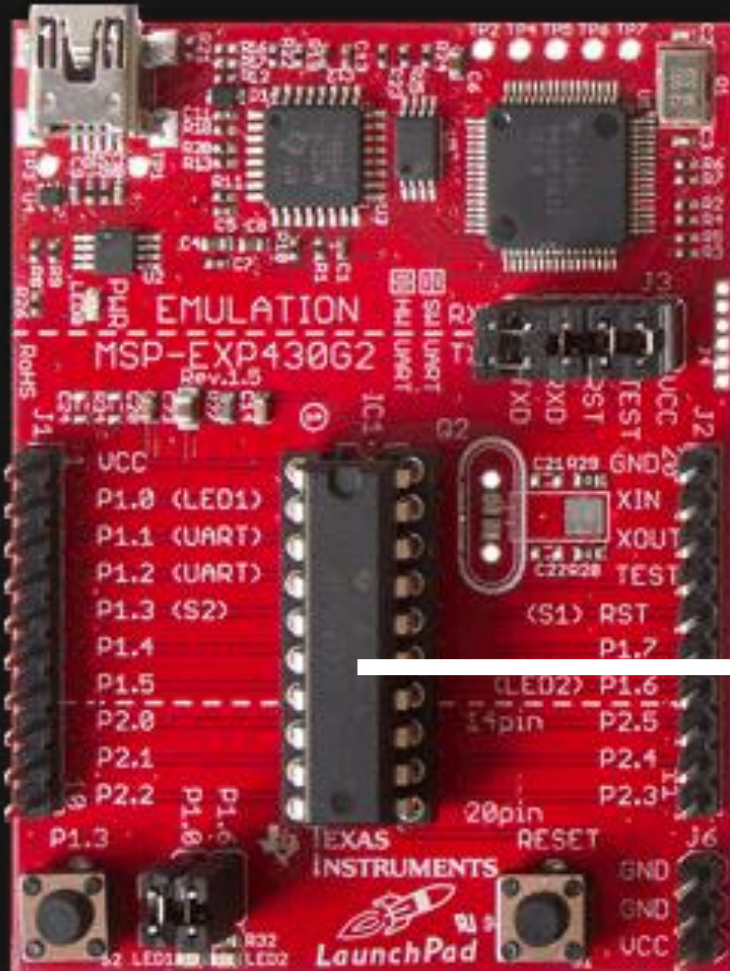
- Clock é eficiente para aplicações com uso de bateria
 - ACLK – Baixa frequência
 - DCO – Oscilador digital interno de alta frequência
 - Bom pra quando não tem USCIA0 pra Uart
 - MCLK – Master clock
 - SMCLK – Slave clock
- DCO volta para operação rapidamente quando ocorre interrupção
 - Menos de 2 μ s

Agenda

- 🔗 A família 430
- 🔗 **MSP430 vs Arduino**
- 🔗 Datasheet MSP430G2553
- 🔗 Ambiente de desenvolvimento: Energia
- 🔗 Ambiente de desenvolvimento: Code Composer Studio
- 🔗 Hands-on: Enviando dados via Bluetooth

MSP430 vs Arduino

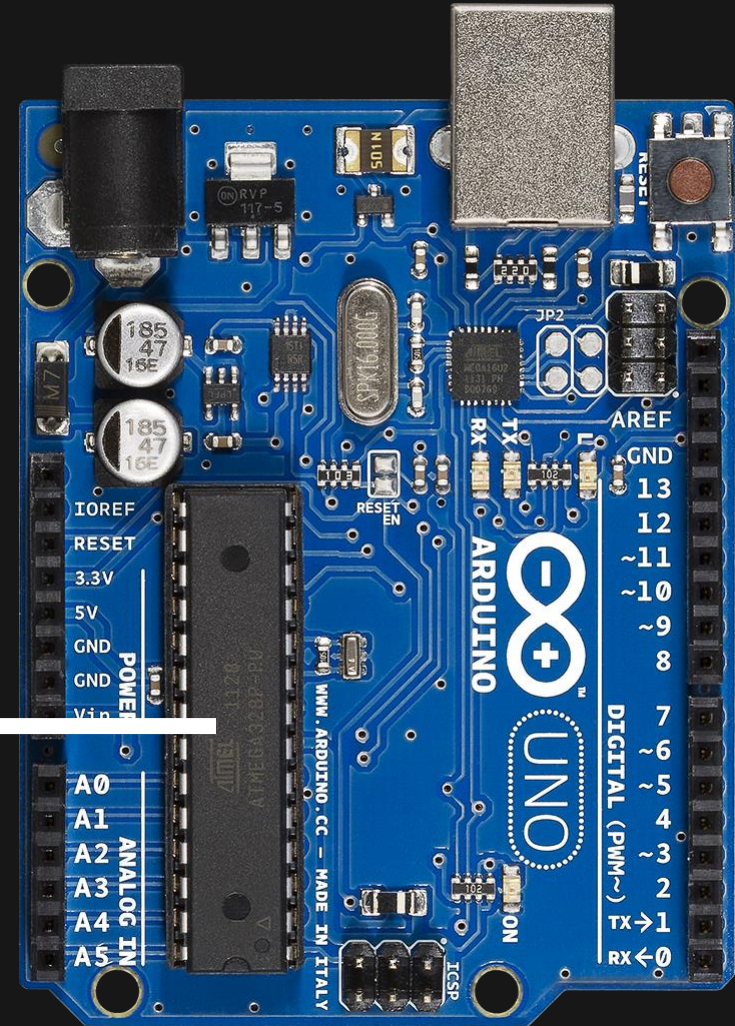
MSP430
16bits



LaunchPad MSP-EXP430G2

ATmega328

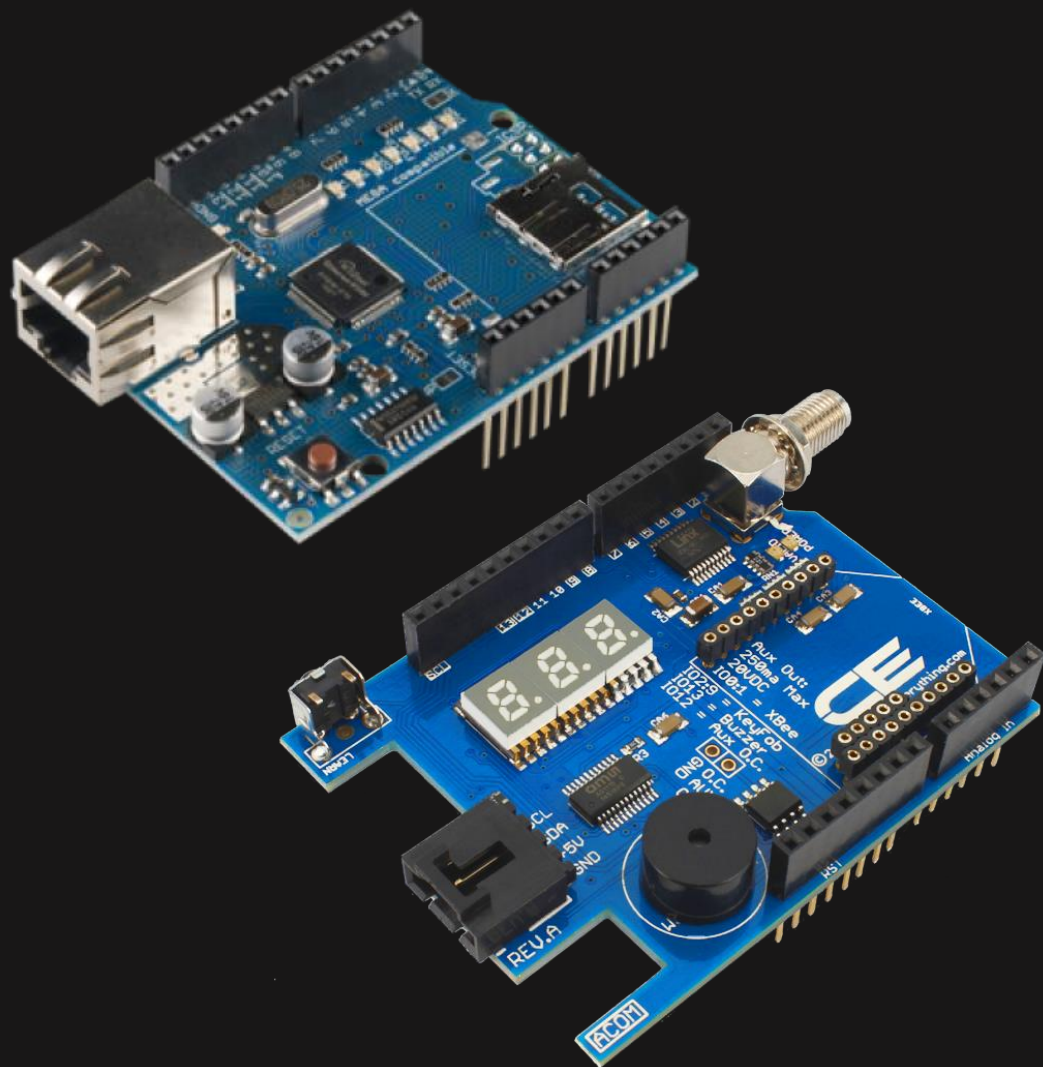
MSP430G2553



Arduino UNO

Arduino
8 bits

MSP430 vs Arduino



fórum arduino

Forums - Arduino - RoboCore

<https://www.robocore.net/modules.php?name=Forums&file=viewforum&f=35> ▼

RoboCore - **Arduino**, Sensores, Motores, Baterias...

Fórum - FilipeFlop

<https://www.filipeflop.com> > Fórum ▼

Aqui você tira suas dúvidas sobre a plataforma **Arduino**, como placas, ... Se você é apaixonado por este mini-pc chamado Raspberry Pi, este é o seu **fórum**.

Forum - Arduino Portugal

<https://www.arduinoportugal.pt/forums/> ▼

Todas as questões sobre Componentes de Eletrônica e dúvidas na criação de Placas de Circuitos Integrados são neste **forum**. 3; 4; Há 9 meses, 3 semanas.

Projeto Arduino

projetoarduino.forumeiros.com/ ▼

Fórum voltado a estudantes, hobistas e profissionais que se interessam por projetos de eletrônica com ênfase no **Arduino**.

Fórum - Embarcados - Sua fonte de informações sobre Sistemas ...

<https://www.embarcados.com.br> > Fórum ▼

Fórum sobre sistemas embarcados, firmware, hardware e linux embarcado. ... me indicar um projeto para testes de portas serial RS232 e RS485 com **arduino**, ...

Arduino - Programação de microcontroladores - Clube do Hardware

<https://www.clubedohardware.com.br> > ... > Programação de microcontroladores ▼

MSP430 vs Arduino



Search through millions of questions and answers



[Login / Register](#)

TI E2E™ Community >

[Forums](#)

[Blogs](#)

[TI Training](#)

[Getting Started](#)

[简体中文](#)

MSP low-power microcontrollers

▼ More

[+ Ask a new question](#)

Welcome to the MSP low-power microcontrollers Section of the TI E2E Support Community. Ask questions, share knowledge, explore ideas, and help solve problems with fellow engineers. To post a question, click on the forum tab then "+ New".

By name



Ascending



[Go to Microcontrollers](#)

[Forums](#)

[MSP low-power microcontroller forum](#)

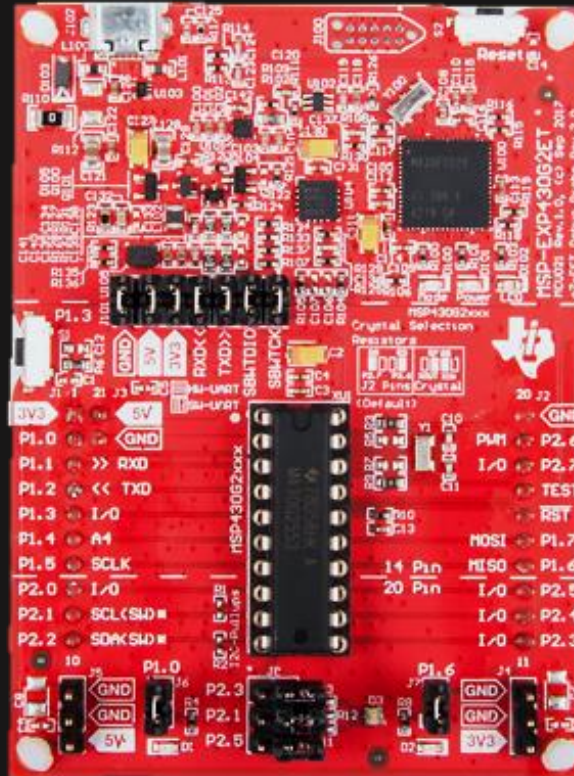


TI E2E Top Contributors

MSP430 vs Arduino



LaunchPad MSP-EXP430G2
(DEFASADO)



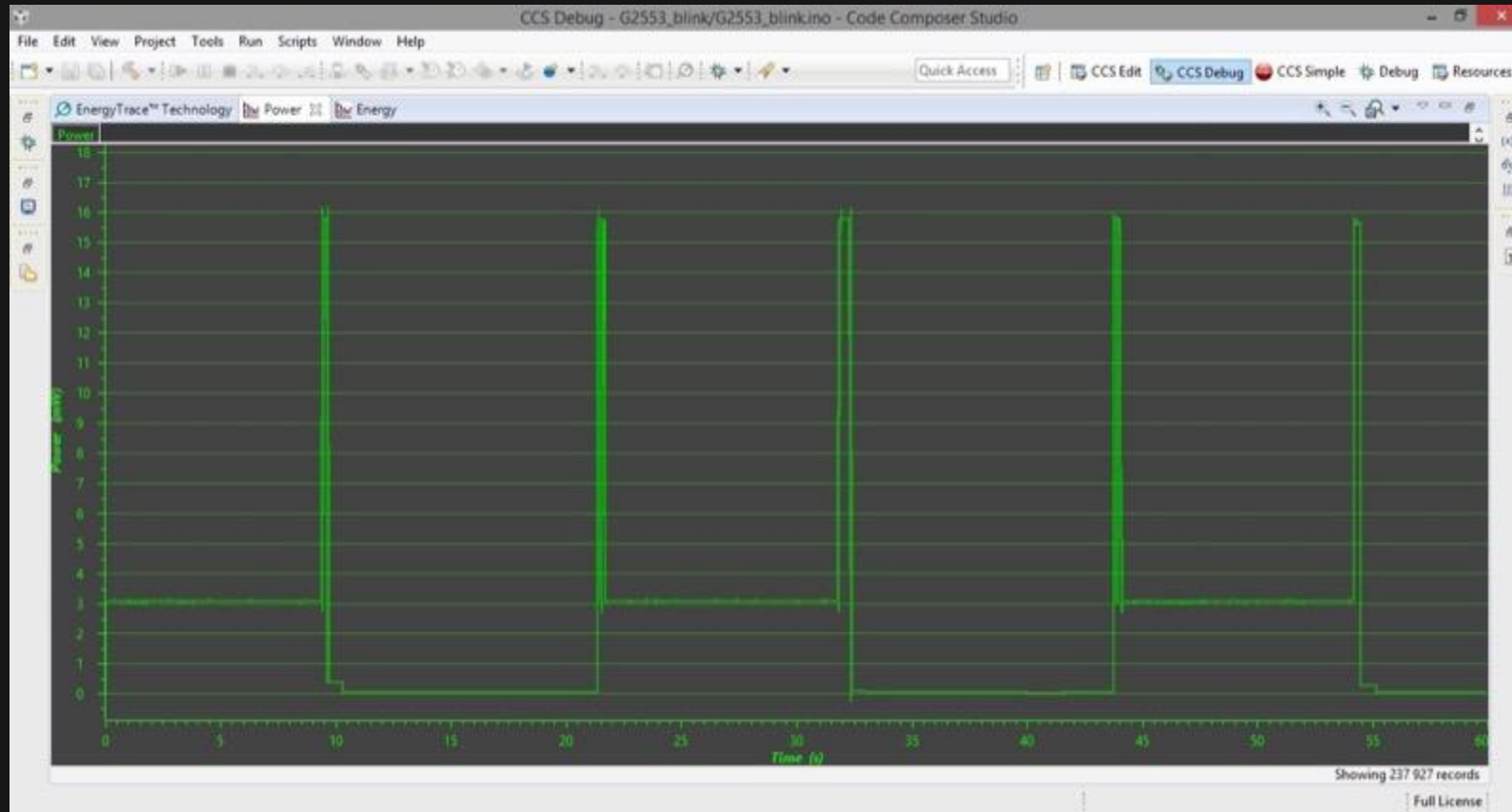
LaunchPad MSP-EXP430G2ET
(ATIVO)

No site da  TEXAS
INSTRUMENTS
\$9,99 (USD)

Melhorias:

- EnergyTrace Measurement
- 3.3V e 5V
- Micro USB
- ...

MSP430 vs Arduino



EnergyTrace Measurement

MSP430 vs Arduino

- Para prototipagem: **Arduino**
- Para projetos que requerem mais robustez e gasto de energia controlado: **MSP430**

Agenda

- 🔗 A família 430
- 🔗 MSP430 vs Arduino
- 🔗 **Datasheet MSP430G2553**
- 🔗 Ambiente de desenvolvimento: Energia
- 🔗 Ambiente de desenvolvimento: Code Composer Studio
- 🔗 Hands-on: Enviando dados via Bluetooth

Datasheet MSP430G2553

Table 1. Available Options⁽¹⁾⁽²⁾

Device	BSL	EEM	Flash (KB)	RAM (B)	Timer_A	COMP_A+ Channel	ADC10 Channel	USCI_A0, USCI_B0	Clock	I/O	Package Type
MSP430G2553IRHB32	1	1	16	512	2x TA3	8	8	1	LF, DCO, VLO	24	32-QFN
MSP430G2553IPW28										24	28-TSSOP
MSP430G2553IPW20										16	20-TSSOP
MSP430G2553IN20										16	20-PDIP

Tensão: 1.8V ~ 3.6V

Ideal é 3.3V

Datasheet MSP430G2553

Table 1. Available Options⁽¹⁾⁽²⁾

Device	BSL	EEM	Flash (KB)	RAM (B)	Timer_A	COMP_A+ Channel	ADC10 Channel	USCI_A0, USCI_B0	Clock	I/O	Package Type
MSP430G2553IRHB32	1	1	16	512	2x TA3	8	8	1	LF, DCO, VLO	24	32-QFN
MSP430G2553IPW28										24	28-TSSOP
MSP430G2553IPW20										16	20-TSSOP
MSP430G2553IN20										16	20-PDIP

MSP430G2453 – 8kB Flash
MSP430G2353 – 4kB Flash
MSP430G2253 – 2kB Flash
MSP430G2153 – 1kB Flash

MSP430G2513 – 8kB Flash
MSP430G2413 – 4kB Flash
MSP430G2313 – 4kB Flash
MSP430G2213 – 2kB Flash
MSP430G2113 – 1kB Flash

S/ ADC

Datasheet MSP430G2553

- Modos de operação
 - 1 modo ativo (Active Mode)
 - 5 modos de baixa energia (Low Power Mode)
 - +GIE: General Interruption Enabled

	CPU	ACLK	DCO	SMCLK	MCLK	Cristal
AM	Ativo	Ativo	Ativo	Ativo	Ativo	Ativo
LPM0	Desativado	Ativo	Ativo	Ativo	Desativado	Ativo
LPM1	Desativado	Ativo	Depende	Ativo	Desativado	Ativo
LPM2	Desativado	Ativo	Ativo	Desativado	Desativado	Ativo
LPM3	Desativado	Ativo	Desativado	Desativado	Desativado	Ativo
LPM4	Desativado	Desativado	Desativado	Desativado	Desativado	Desativado

Datasheet MSP430G2553

Table 5. Interrupt Sources, Flags, and Vectors

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Timer+ Flash key violation PC out-of-range ⁽¹⁾	PORIFG RSTIFG WDTIFG KEYV ⁽²⁾	Reset	0FFFEh	31, highest
NMI Oscillator fault Flash memory access violation	NMIIFG OFIFG ACCVIFG ⁽²⁾⁽³⁾	(non)-maskable (non)-maskable (non)-maskable	0FFFCCh	30
Timer1_A3	TA1CCR0 CCIFG ⁽⁴⁾	maskable	0FFFAh	29
Timer1_A3	TA1CCR2 TA1CCR1 CCIFG, TAIFG ⁽²⁾⁽⁴⁾	maskable	0FFF8h	28
Comparator_A+	CAIFG ⁽⁴⁾	maskable	0FFF6h	27
Watchdog Timer+	WDTIFG	maskable	0FFF4h	26
Timer0_A3	TA0CCR0 CCIFG ⁽⁴⁾	maskable	0FFF2h	25
Timer0_A3	TA0CCR2 TA0CCR1 CCIFG, TAIFG ⁽⁵⁾⁽⁴⁾	maskable	0FFF0h	24
USCI_A0/USCI_B0 receive USCI_B0 I2C status	UCA0RXIFG, UCB0RXIFG ⁽²⁾⁽⁵⁾	maskable	0FFEEh	23
USCI_A0/USCI_B0 transmit USCI_B0 I2C receive/transmit	UCA0TXIFG, UCB0TXIFG ⁽²⁾⁽⁶⁾	maskable	0FFECCh	22
ADC10 (MSP430G2x53 only)	ADC10IFG ⁽⁴⁾	maskable	0FFEAh	21
			0FFE8h	20
I/O Port P2 (up to eight flags)	P2IFG.0 to P2IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE6h	19
I/O Port P1 (up to eight flags)	P1IFG.0 to P1IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE4h	18
			0FFE2h	17
			0FFE0h	16
See ⁽⁷⁾			0FFDEh	15
See ⁽⁸⁾			0FFDEh to 0FFC0h	14 to 0, lowest

Datasheet MSP430G2553

Table 5. Interrupt Sources, Flags, and Vectors

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Timer+ Flash key violation PC out-of-range ⁽¹⁾	PORIFG RSTIFG WDTIFG KEYV ⁽²⁾	Reset	0FFFEh	31, highest
NMI Oscillator fault Flash memory access violation	NMIIFG OFIFG ACCVIFG ⁽²⁾⁽³⁾	(non)-maskable (non)-maskable (non)-maskable	0FFFCCh	30
Timer1_A3	TA1CCR0 CCIFG ⁽⁴⁾	maskable	0FFFAh	29
Timer1_A3	TA1CCR2 TA1CCR1 CCIFG, TAIFG ⁽²⁾⁽⁴⁾	maskable	0FFF8h	28
Comparator_A+	CAIFG ⁽⁴⁾	maskable	0FFF6h	27
Watchdog Timer+	WDTIFG	maskable	0FFF4h	26
Timer0_A3	TA0CCR0 CCIFG ⁽⁴⁾	maskable	0FFF2h	25
Timer0_A3	TA0CCR2 TA0CCR1 CCIFG, TAIFG ⁽⁵⁾⁽⁴⁾	maskable	0FFF0h	24
USCI_A0/USCI_B0 receive USCI_B0 I2C status	UCA0RXIFG, UCB0RXIFG ⁽²⁾⁽⁵⁾	maskable	0FFEEh	23
USCI_A0/USCI_B0 transmit USCI_B0 I2C receive/transmit	UCA0TXIFG, UCB0TXIFG ⁽²⁾⁽⁶⁾	maskable	0FFECCh	22
ADC10 (MSP430G2x53 only)	ADC10IFG ⁽⁷⁾	maskable	0FFEAh	21
			0FFE8h	20
I/O Port P2 (up to eight flags)	P2IFG.0 to P2IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE6h	19
I/O Port P1 (up to eight flags)	P1IFG.0 to P1IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE4h	18
			0FFE2h	17
			0FFE0h	16
See ⁽⁷⁾			0FFDEh	15
See ⁽⁸⁾			0FFDEh to 0FFC0h	14 to 0, lowest

UART:
USCIA0

Vetor de interrupção RX:
UCA0RXIFG

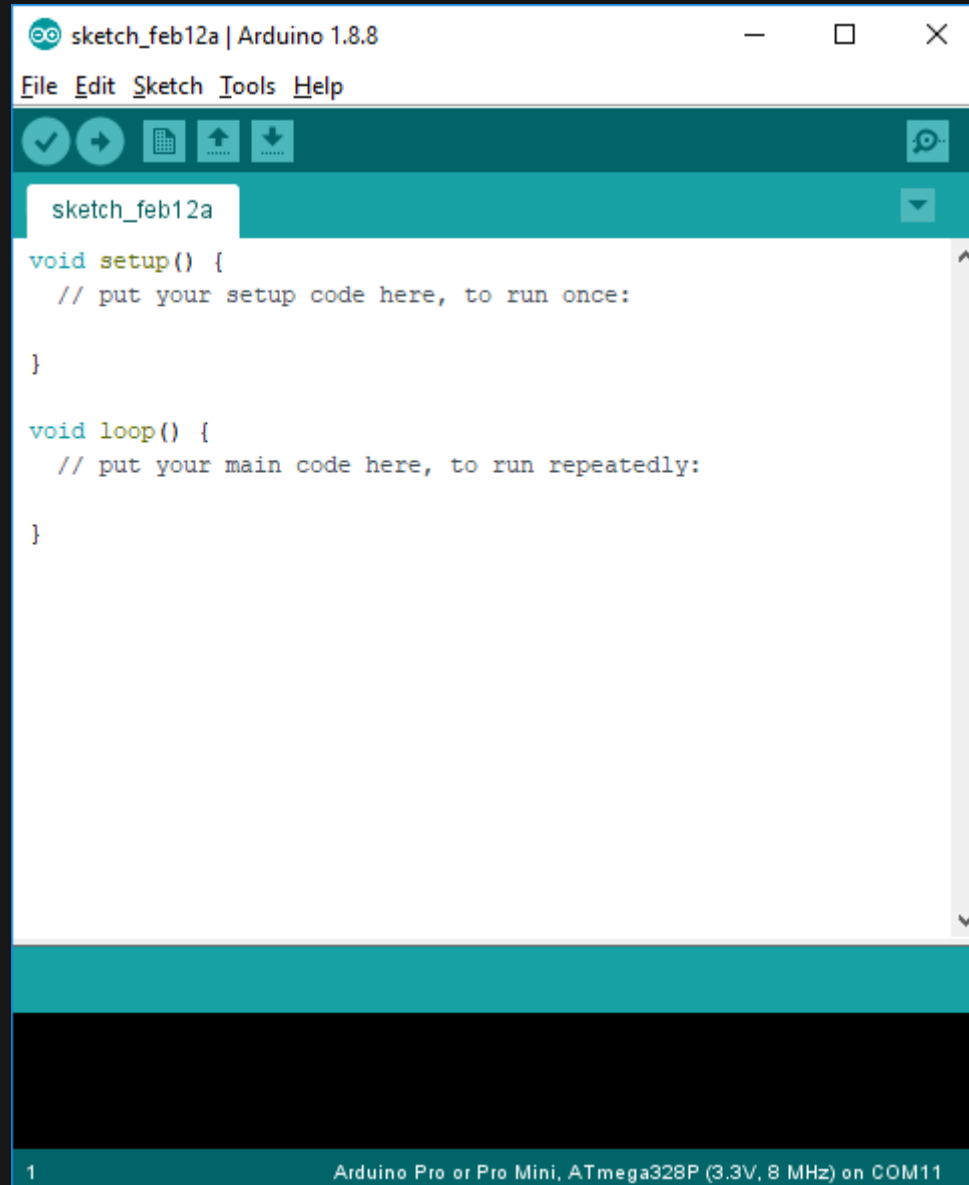
Vetor de interrupção TX:
UCA0TXIFG



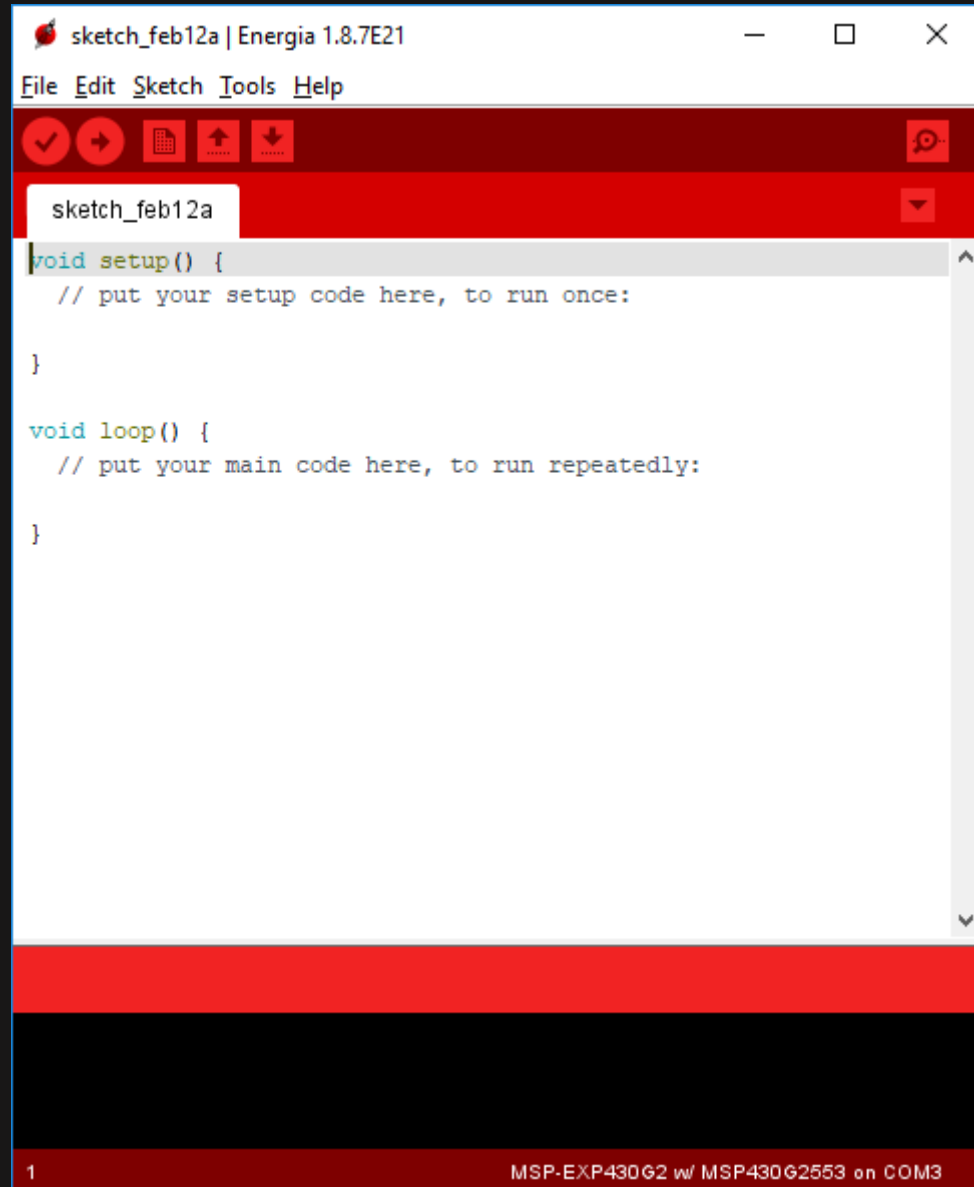
Agenda

- 🔗 A família 430
- 🔗 MSP430 vs Arduino
- 🔗 Datasheet MSP430G2553
- 🔗 **Ambiente de desenvolvimento: Energia**
- 🔗 Ambiente de desenvolvimento: Code Composer Studio
- 🔗 Hands-on: Enviando dados via Bluetooth

Arduino



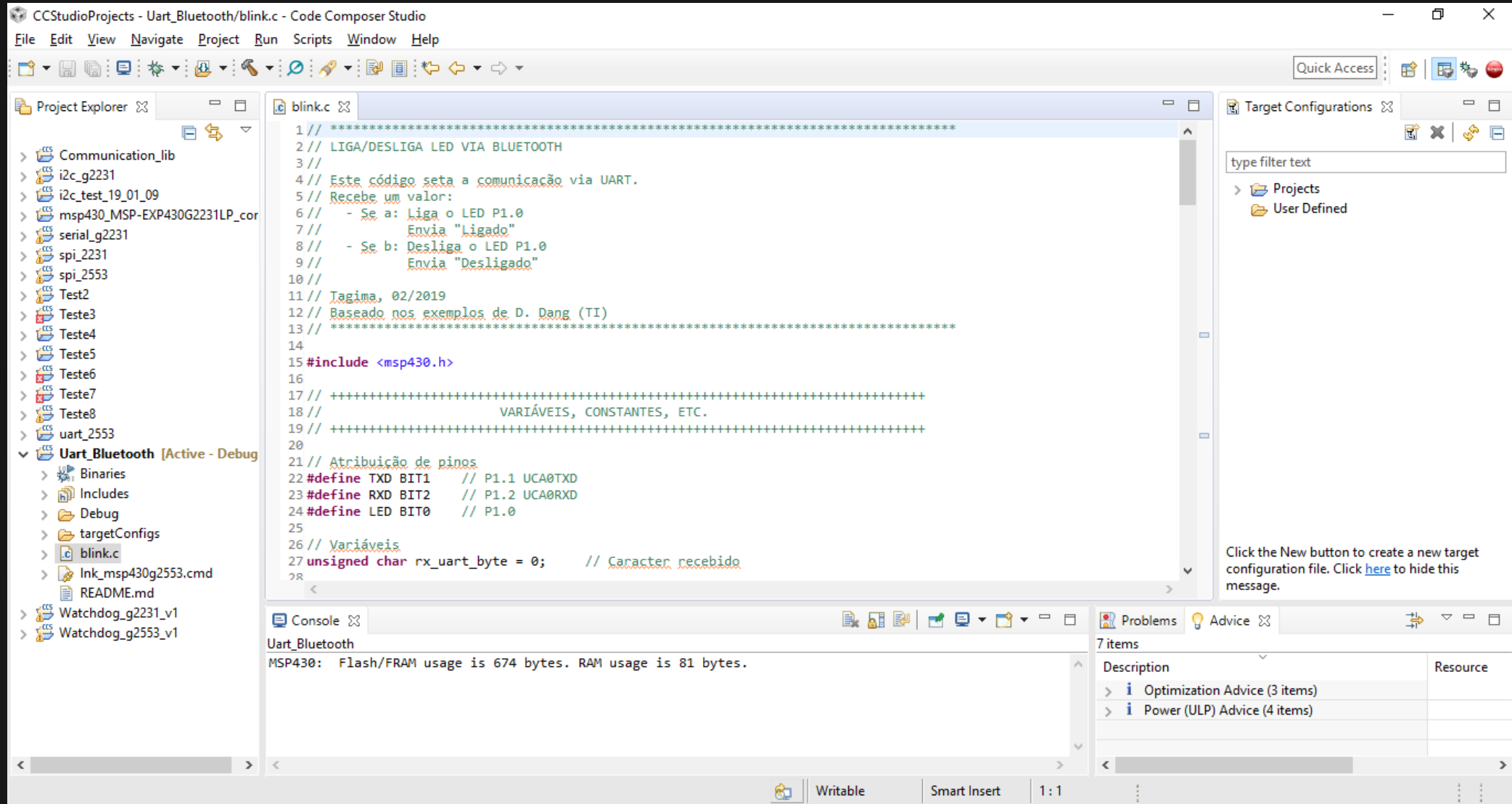
Energia



Agenda

- 🔗 A família 430
- 🔗 MSP430 vs Arduino
- 🔗 Datasheet MSP430G2553
- 🔗 Ambiente de desenvolvimento: Energia
- 🔗 **Ambiente de desenvolvimento: Code Composer Studio**
- 🔗 Hands-on: Enviando dados via Bluetooth

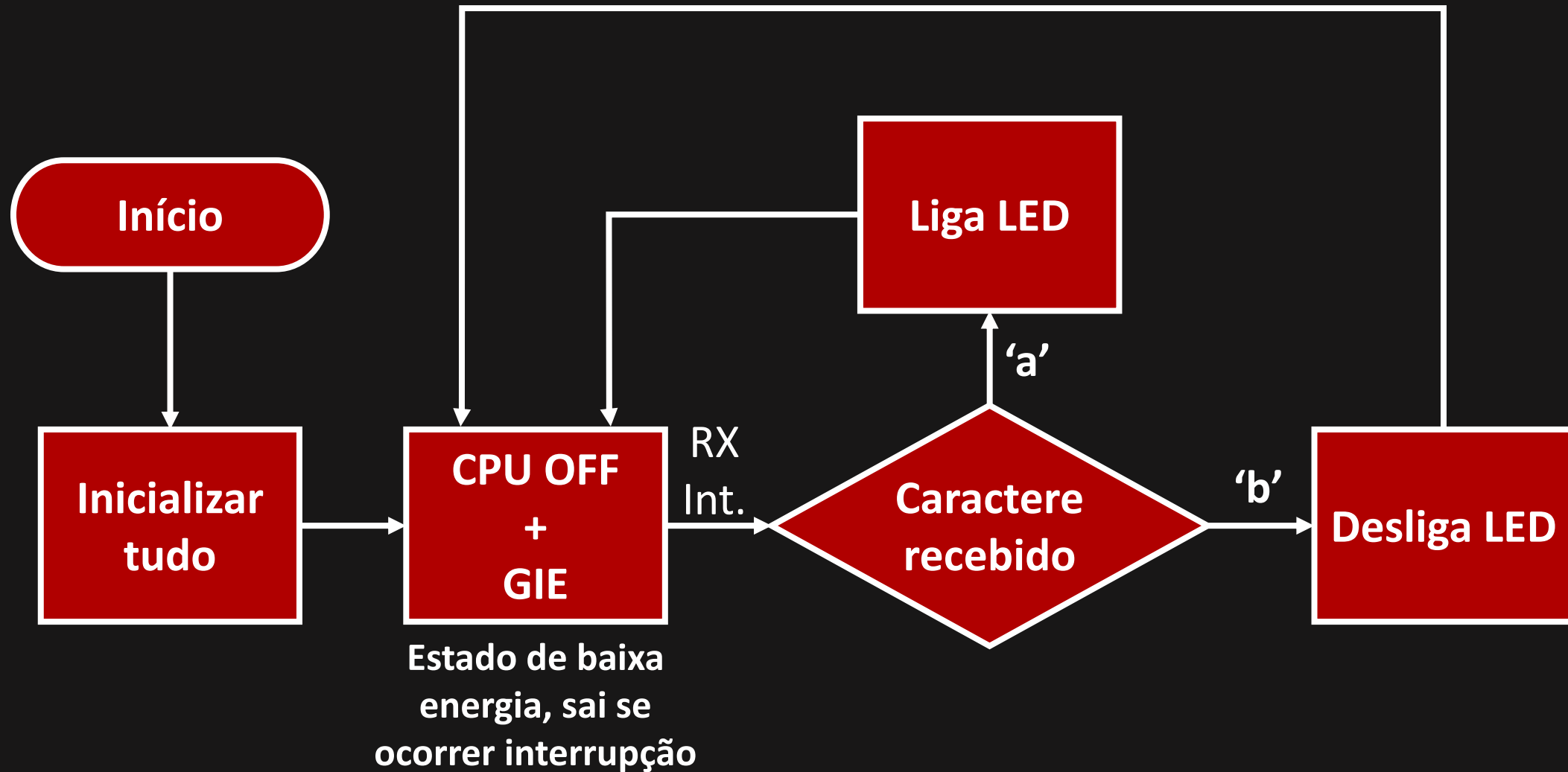
Code Composer Studio



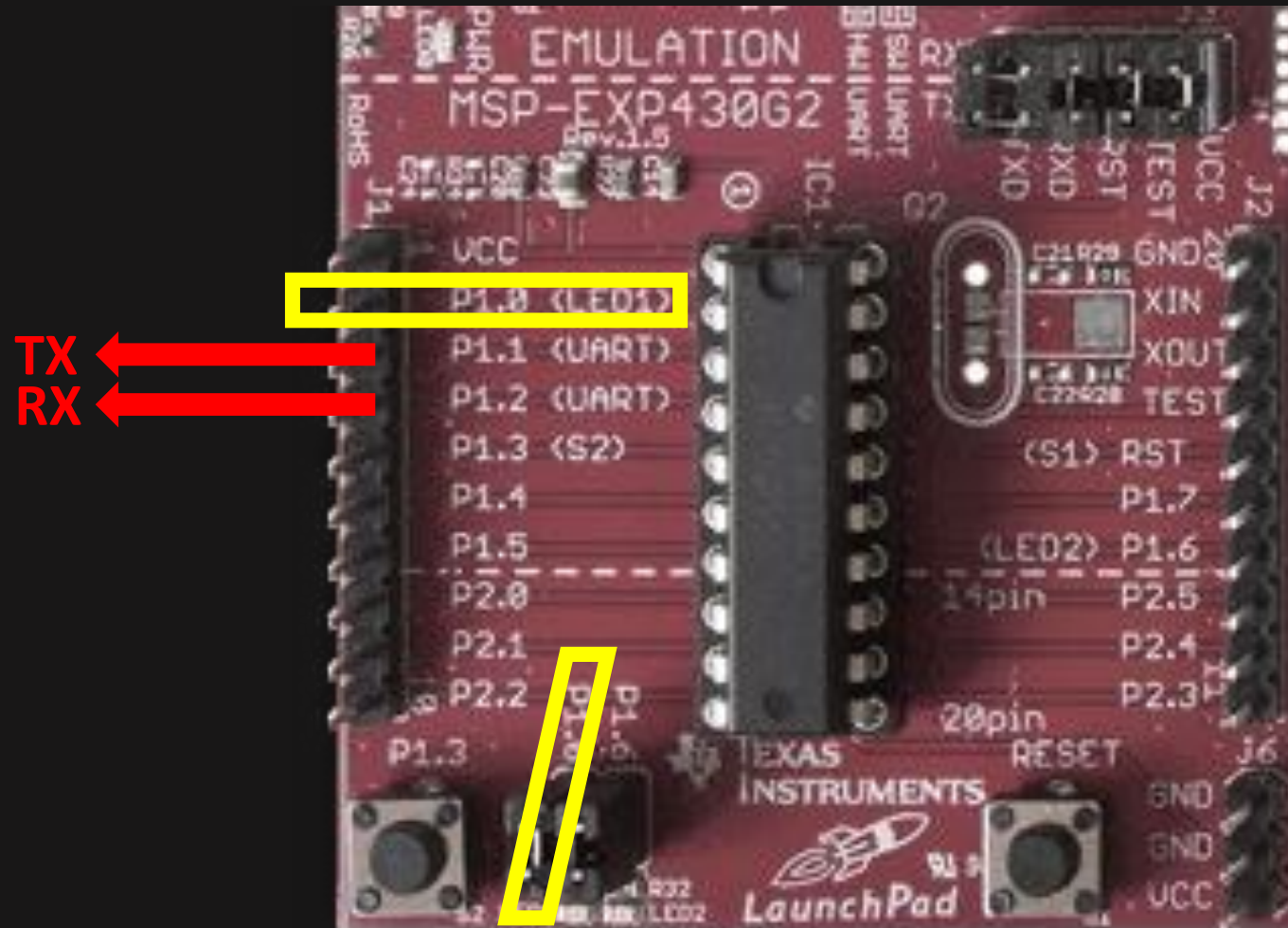
Agenda

- 📄 A família 430
- 📄 MSP430 vs Arduino
- 📄 Datasheet MSP430G2553
- 📄 Ambiente de desenvolvimento: Energia
- 📄 Ambiente de desenvolvimento: Code Composer Studio
- 📄 **Hands-on: Enviando dados via Bluetooth**

O que fazer?



Pinagem



Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Setando:

```
IE2 |= UCA0RXIE;
```

Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Setando:

IE2 |= UCA0RXIE;



Bitwise

Operação bit a bit

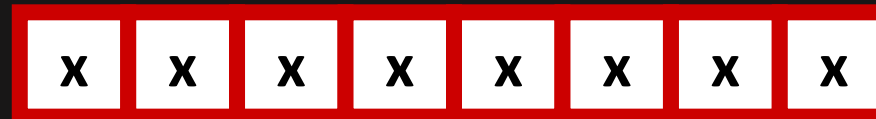
Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Setando:

$IE2 \mid= UCA0RXIE;$

IE2



(Interrupt Enable Register 2)

UCA0RXIE



(Flag correspondente a posição de UCA0RXIE no registrador IE2)

Novo IE2



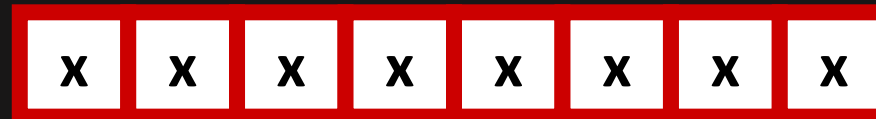
Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Setando:

$IE2 \mid= UCA0RXIE;$

IE2



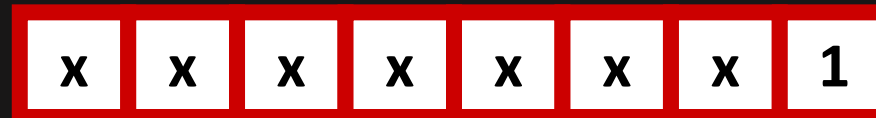
(Interrupt Enable Register 2)

UCA0RXIE



(Flag correspondente a posição de UCA0RXIE no registrador IE2)

Novo IE2



Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Zerando:

```
IE2 &= ~UCA0RXIE;
```

Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Zerando:

```
IE2 &= ~UCA0RXIE;
```



Bitwise

Operação bit a bit

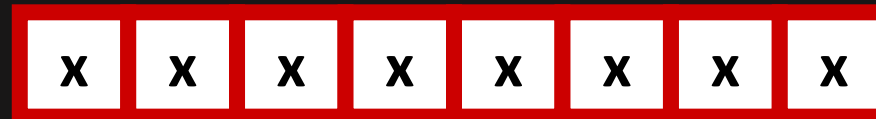
Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Zerando:

$IE2 \&= \sim UCA0RXIE;$

IE2



(Interrupt Enable Register 2)

UCA0RXIE



(Flag correspondente a posição de UCA0RXIE no registrador IE2)

Novo IE2



Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Zerando:

$IE2 \&= \sim UCA0RXIE;$

IE2



(Interrupt Enable Register 2)

$\sim UCA0RXIE$



(Flag correspondente a posição de UCA0RXIE no registrador IE2)

Novo IE2



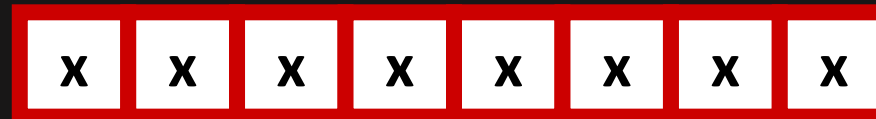
Zerando e setando flags

Queremos manipular o vetor de interrupção, habilitando (setando) ou desabilitando (zerando) a interrupção do TX

- Zerando:

$IE2 \&= \sim UCA0RXIE;$

IE2



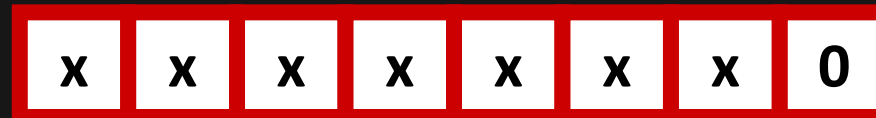
(Interrupt Enable Register 2)

$\sim UCA0RXIE$



(Flag correspondente a posição de UCA0RXIE no registrador IE2)

Novo IE2



Uart_9600_init_1MHz

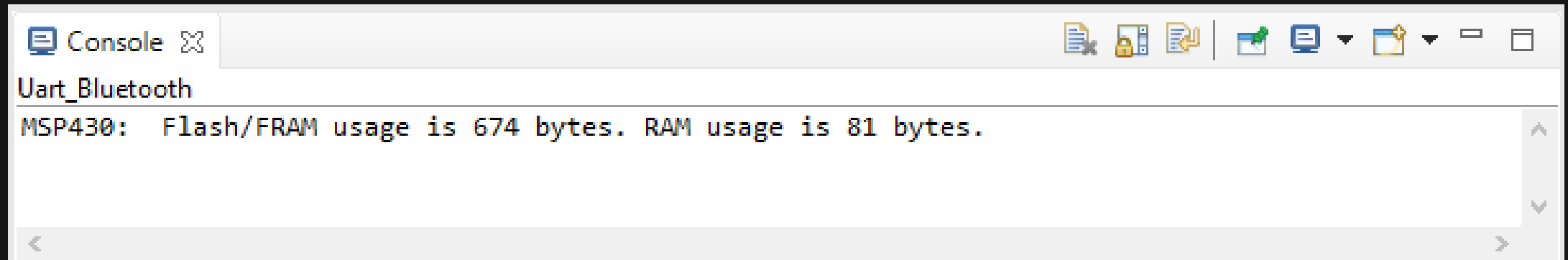
- Estamos usando o DCO em 1MHz

$$Baudrate = \frac{1Mhz}{104} \cong 9600$$

E esse código no Energia?

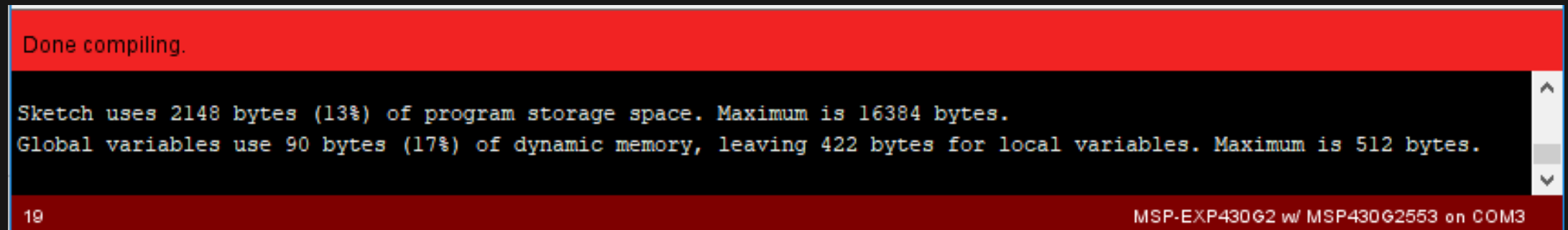
(Hands-on)

Então pra que usar o CCS?



The screenshot shows the 'Console' window in the CCS IDE. The window title is 'Console' with a close button. Below the title bar, the text 'Uart_Bluetooth' is visible. The main content area displays the message: 'MSP430: Flash/FRAM usage is 674 bytes. RAM usage is 81 bytes.' The window has a scrollbar on the right and a horizontal scrollbar at the bottom.

```
Uart_Bluetooth
MSP430: Flash/FRAM usage is 674 bytes. RAM usage is 81 bytes.
```



The screenshot shows the 'Console' window in the CCS IDE. The window title is 'Console' with a close button. Below the title bar, the text 'Done compiling.' is visible. The main content area displays the message: 'Sketch uses 2148 bytes (13%) of program storage space. Maximum is 16384 bytes. Global variables use 90 bytes (17%) of dynamic memory, leaving 422 bytes for local variables. Maximum is 512 bytes.' The window has a scrollbar on the right and a horizontal scrollbar at the bottom.

```
Done compiling.
Sketch uses 2148 bytes (13%) of program storage space. Maximum is 16384 bytes.
Global variables use 90 bytes (17%) of dynamic memory, leaving 422 bytes for local variables. Maximum is 512 bytes.
```


Então pra que usar o CCS?

- Energy Trace Measurement
- Liberdade de opções

Código disponível em

https://github.com/Tagima/CPBR12_MSP430

