# CP316 Interrupts

Terry Sturtevant

Wilfrid Laurier University

June 26, 2019

æ

∍ →

< □ > < 同 > < 回 >

Introduction Initialization Service Rules for Interrupts

# Introduction to Interrupts

Terry Sturtevant CP316 Interrupts

・ロト ・日 ・ ・ ヨト ・

문 🛌 문

Introduction Initialization Service Rules for Interrupts

# Introduction to Interrupts

Interrupts allow program control to change due to events.

• 同 • • 三 • •

B> B

Introduction Initialization Service Rules for Interrupts

# Introduction to Interrupts

Interrupts allow program control to change due to *events*. **Interrupt vectors** are program locations

▲ 同 ▶ ▲ 国 ▶ ▲

⇒ →

Introduction Initialization Service Rules for Interrupts

# Introduction to Interrupts

Interrupts allow program control to change due to *events*. Interrupt vectors are program locations where interrupt service routines are located.

▲ 同 ▶ ▲ 国 ▶ ▲

Introduction Initialization Service Rules for Interrupts

# Introduction to Interrupts

Interrupts allow program control to change due to events.

Interrupt vectors are program locations

where interrupt service routines are located.

After interrupt service, the program control returns to original. but not the other way around.

#### Introduction

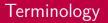
Initialization Service Rules for Interrupts

# Terminology

Terry Sturtevant CP316 Interrupts

< ロ > < 部 > < き > < き >

Introduction Initialization Service Rules for Interrupts

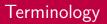


• Enabled/Disabled



< => < => < => < =>

Introduction Initialization Service Rules for Interrupts



#### • Enabled/Disabled

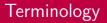
whether a specific event will generate an interrupt



<ロト < 部 ト < 注 ト <</p>

문 문 문

Introduction Initialization Service Rules for Interrupts



### • Enabled/Disabled

whether a specific event will generate an interrupt

Pending

<ロト < 部 ト < 注 ト <</p>

문 문 문

Introduction Initialization Service Rules for Interrupts

# Terminology

### • Enabled/Disabled

whether a specific event will generate an interrupt

#### Pending

whether the event has occurred

• 同 • • 三 • •

문 문 문

Introduction Initialization Service Rules for Interrupts

# Terminology

### • Enabled/Disabled

whether a specific event will generate an interrupt

#### Pending

whether the event has occurred

#### Flag

• □ > • □ > • □ > • □

문어 문

Introduction Initialization Service Rules for Interrupts

# Terminology

### • Enabled/Disabled

whether a specific event will generate an interrupt

#### Pending

whether the event has occurred

### Flag

bit in memory indicating an interrupt is pending

・ 同 ト ・ ヨ ト ・

⇒ →

Introduction Initialization Service Rules for Interrupts

# Terminology

### Enabled/Disabled

whether a specific event will generate an interrupt

### Pending

whether the event has occurred

### Flag

bit in memory indicating an interrupt is pending Note: Flags are usually set whether or not corresponding interrupts are enabled.

▲ 同 ▶ ▲ 国 ▶

⇒ →

Introduction Initialization Service Rules for Interrupts

# Terminology

### Enabled/Disabled

whether a specific event will generate an interrupt

### Pending

whether the event has occurred

### Flag

bit in memory indicating an interrupt is pending Note: Flags are usually set whether or not corresponding interrupts are enabled.

### • Set or Clear

▲ 同 ▶ ▲ 国 ▶

∍ →

Introduction Initialization Service Rules for Interrupts

# Terminology

### • Enabled/Disabled

whether a specific event will generate an interrupt

### Pending

whether the event has occurred

### Flag

bit in memory indicating an interrupt is pending

Note: Flags are usually set whether or not corresponding interrupts are enabled.

#### • Set or Clear

state of flag indicating pending or not

▲ 同 ▶ → 三 ▶

#### Introduction

Initialization Service Rules for Interrupts

## Interrupts

Terry Sturtevant CP316 Interrupts

#### Introduction

Service Rules for Interrupts

## Interrupts

overview

Terry Sturtevant CP316 Interrupts

< ロ > < 回 > < 回 > < 回 > < 回 >

Introduction

Service Rules for Interrupts

## Interrupts

overview

 $\rightarrow$  Chapters 12 and 13

< => < => < => < =>



Introduction Initialization Service Rules for Interrupts

#### Here is a compiled code fragment from a typical program:

Terry Sturtevant CP316 Interrupts

æ

⇒ →

<ロト < 部 ト < 注 ト <</p>

Here is a compiled code fragment from a typical program:

```
;;; vectors
```

```
rjmp RESET ; Reset Handler
rjmp EXT_INT0 ; IRQ0 Handler
rjmp EXT_INT1 ; IRQ1 Handler
rjmp PCINT0 ; PCINT0 Handler
rjmp PCINT1 ; PCINT1 Handler
rjmp PCINT2 ; PCINT2 Handler
```

・ 同 ト ・ ヨ ト ・ ヨ ト

Introduction Initialization Service Rules for Interrupts

# Interrupt Initialization

Terry Sturtevant CP316 Interrupts

< => < => < => < =>

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

(日)

⇒ →

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

O configure individual interrupt sources (various registers)

▲ 伊 ▶ ▲ 三 ▶

Introduction Initialization Service Rules for Interrupts

### Interrupt Initialization

The sequence of initializing interrupts is important.

 configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

enable (various registers)

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

enable (various registers)

for each individual source required

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

enable (various registers)

for each individual source required

**global** enable *all* interrupts (**STATUS**)

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

enable (various registers)

for each individual source required

 global enable all interrupts (STATUS) after all individual sources have been initialized

Introduction Initialization Service Rules for Interrupts

## Interrupt Initialization

The sequence of initializing interrupts is important.

- configure individual interrupt sources (various registers) any needed configuration, initialization of variables, etc.
- elear flags (various registers)

On reset, many interrupt flags are in an unknown state, so interrupts may be incorrectly identified as pending.

• enable (various registers)

for each individual source required

global enable all interrupts (STATUS) after all individual sources have been initialized needed for any interrupt service to occur

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

Terry Sturtevant CP316 Interrupts

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0



O configure individual interrupt sources (various registers)

æ

イロト イ団ト イヨト イヨト

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- 2 clear flags (various registers)

イロト イポト イヨト イヨト

э

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- clear flags (various registers)
   OCF0A, OCF0B, TOV0 are in TIFR0

- 4 同 ト - 4 目 ト

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- configure individual interrupt sources (various registers)clear flags (various registers)
  - OCF0A, OCF0B, TOV0 are in TIFR0
  - write '1' to clear

- 4 同 1 4 日 1 4 日 1

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- clear flags (various registers)
   OCF0A, OCF0B, TOV0 are in TIFR0
   write '1' to clear
- enable (various registers)

▲ 同 ▶ ▲ 国 ▶ ▲

⇒ →

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- clear flags (various registers)
   OCF0A, OCF0B, TOV0 are in TIFR0
   write '1' to clear
- enable (various registers)
   OCIE0A, OCIE0B, TOIE0 are in TIFR0

< 🗇 🕨 < 🚍 🕨

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- clear flags (various registers)
   OCF0A, OCF0B, TOV0 are in TIFR0
   write '1' to clear
- enable (various registers)
   OCIE0A, OCIE0B, TOIE0 are in TIFR0 for each individual source required

Introduction Initialization Service Rules for Interrupts

# Example: Timer 0

- O configure individual interrupt sources (various registers)
- clear flags (various registers)
   OCF0A, OCF0B, TOV0 are in TIFR0
   CLU
  - write '1' to clear
- enable (various registers)
   OCIE0A, OCIE0B, TOIE0 are in TIFR0 for each individual source required
   I is in STATUS

→ < Ξ →</p>

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines

Terry Sturtevant CP316 Interrupts

æ

イロト イ団ト イヨト イヨト

Introduction Initialization Service Rules for Interrupts

### Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

< A ► <

Introduction Initialization Service Rules for Interrupts

### Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

save vital registers

Image: A image: A

Introduction Initialization Service Rules for Interrupts

## Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

save vital registers STATUS

- 4 同 🕨 - 4 回 🕨 - 4

∍ →

Introduction Initialization Service Rules for Interrupts

## Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

- save vital registers STATUS
- Ontire source

▲ 同 ▶ ▲ 国 ▶ ▲

Introduction Initialization Service Rules for Interrupts

## Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

- save vital registers STATUS
- Ontire source

Check specific flag bit to confirm that the expected source caused the interrupt.

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

## Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

- save vital registers STATUS
- Ontire source

Check specific flag bit to confirm that the expected source caused the interrupt.

Ø process

▲ 伊 ▶ ▲ 三 ▶

Introduction Initialization Service Rules for Interrupts

## Interrupt Service Routines

The sequence of actions in *servicing* interrupts is also important.

- save vital registers STATUS
- Ontire source

Check specific flag bit to confirm that the expected source caused the interrupt.

Ø process

more on this later

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

æ

∍ →

イロト イヨト イヨト イ

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)



Terry Sturtevant CP316 Interrupts

æ

イロト イ団ト イヨト イヨト

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

- Ø process
- elear flags (if required)

(日)

⇒ →

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

#### Ø process

clear flags (if required)

Some flags need to be cleared explicitly; others will happen as part of normal service.

< /i>
< /i>
< /i>
< /i>
< /i>
< /i>

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

#### Ø process

clear flags (if required)

Some flags need to be cleared explicitly; others will happen as part of normal service.

Always check whether a flag will be cleared automatically or not.

▲ 同 ▶ ▲ 国 ▶

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

#### Ø process

clear flags (if required)

Some flags need to be cleared explicitly; others will happen as part of normal service.

Always check whether a flag will be cleared automatically or not.

In the second second

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

#### Ø process

clear flags (if required)

Some flags need to be cleared explicitly; others will happen as part of normal service.

Always check whether a flag will be cleared automatically or not.

restore vital registers STATUS

▲ 伊 ▶ ▲ 三 ▶

Introduction Initialization Service Rules for Interrupts

# Interrupt Service Routines (continued)

#### Ø process

Oclear flags (if required)

Some flags need to be cleared explicitly; others will happen as part of normal service.

Always check whether a flag will be cleared automatically or not.

- restore vital registers STATUS
- 🧿 reti

like a subroutine return, but re-enables interrupts

Introduction Initialization Service Rules for Interrupts

# Rules for Interrupts

Terry Sturtevant CP316 Interrupts

< => < => < => < =>

æ

Introduction Initialization Service Rules for Interrupts

## Rules for Interrupts

• Never wait in interrupts.

æ

イロト イ団ト イヨト イヨト

Introduction Initialization Service Rules for Interrupts

## Rules for Interrupts

• Never wait in interrupts.

set flags, have waiting in main program

- 4 回 ト 4 回 ト 4

∍ →

Introduction Initialization Service Rules for Interrupts

# Rules for Interrupts

- Never wait in interrupts.
   set flags, have waiting in main program
- Use interrupts for events of very short duration

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

# Rules for Interrupts

- Never wait in interrupts.
   set flags, have waiting in main program
- Use interrupts for events of very short duration
  - i.e. if polling may miss them entirely

▲ 同 ▶ → 三 ▶

Introduction Initialization Service Rules for Interrupts

## Interrupt Sources

Terry Sturtevant CP316 Interrupts

æ

Introduction Initialization Service Rules for Interrupts

## Interrupt Sources

internal, external

< => < => < => < =>

æ

Introduction Initialization Service Rules for Interrupts

## Interrupt Sources

internal, external pin change, timers, serial, ADC, etc.

<ロト < 部 ト < 注 ト <</p>

문▶ 문