TI-RSLK

Texas Instruments Robotics System Learning Kit

Texas Instruments

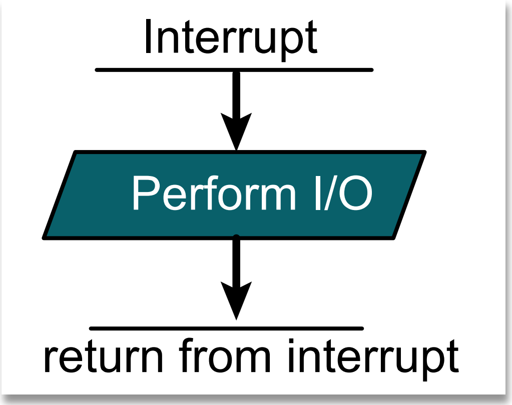
Module 14

Lecture: Real-time Systems - Theory

Real-Time Systems

You will learn in this module

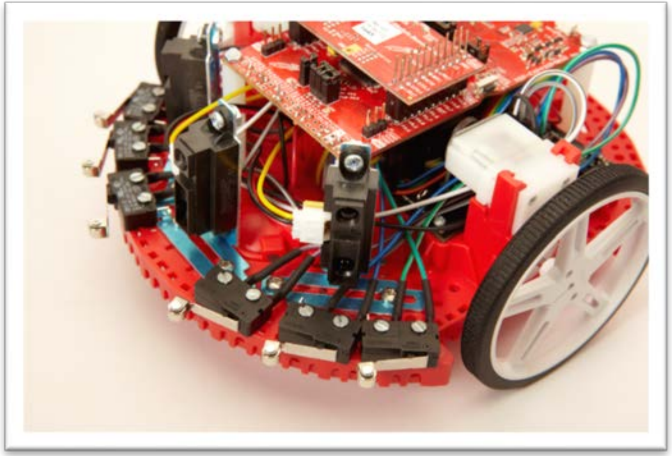
- Tasks
  - Periodic
  - Aperiodic
  - Sporadic
- Performance measures
  - Latency
  - Response time
- Real-Time Systems
  - Hard
  - Firm
  - Soft



Interrupt

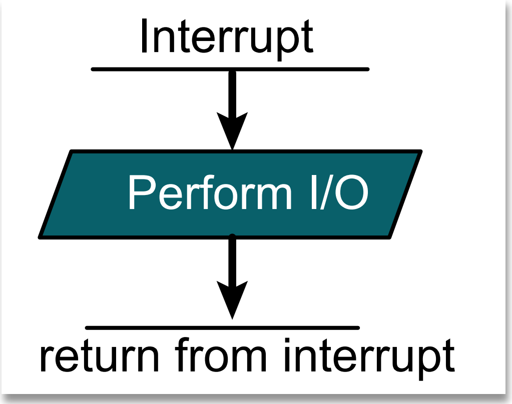
Perform I/O

return from interrupt



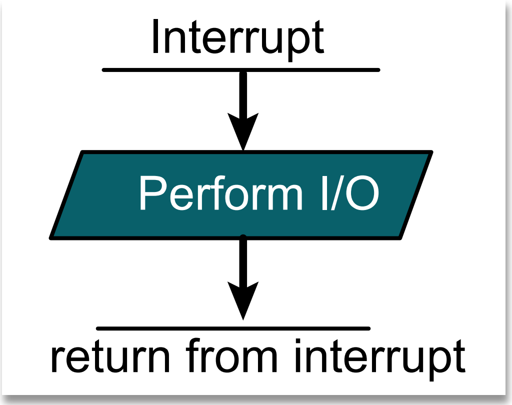
Real-Time Systems

- Tasks
  - Periodic (sampling, digital controller)
  - Aperiodic (I/O)
  - Sporadic (faults)
- Latency
- Response time
- Priority



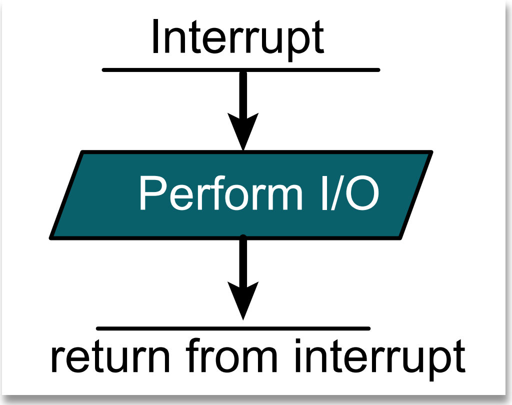
Real-Time Systems

- Hard real time systems
  - Guaranteed bounded latency/response time



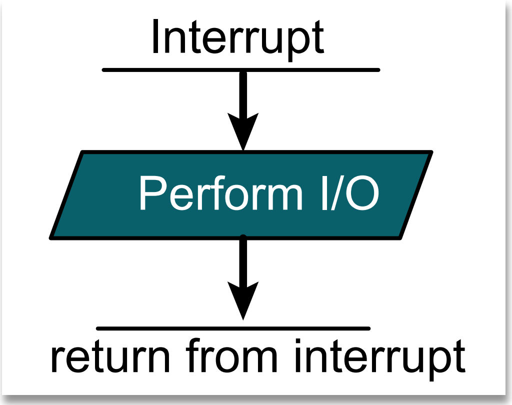
Real-Time Systems

- Firm real time systems
  - Missed deadline loss of quality



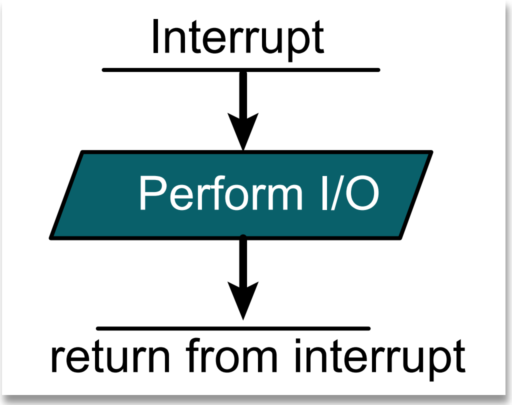
Real-Time Systems

- Soft real time systems
  - Delayed response reduces value



Real-Time Systems

- Not real time
    - Best effort, no deadlines whatsoever



Real-time behavior

Factors that affect latency

- Time to finish instruction
- Running with I=1 (disabled)
- Running higher priority interrupts

Factors that affect response time

- Time to finish instruction
- Running with I=1 (disabled)
- Running higher priority interrupts
- Performing the service

Best Practices

- Assign priority appropriately
- Try not to disable interrupts
- Make the time to execute an ISR small compared to the time between interrupt triggers
- Avoid loops inside ISR

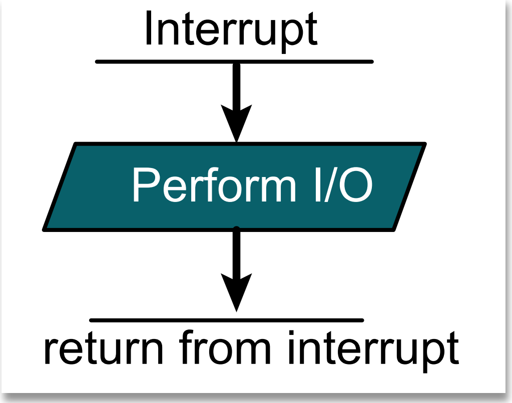
Critical Section (review)

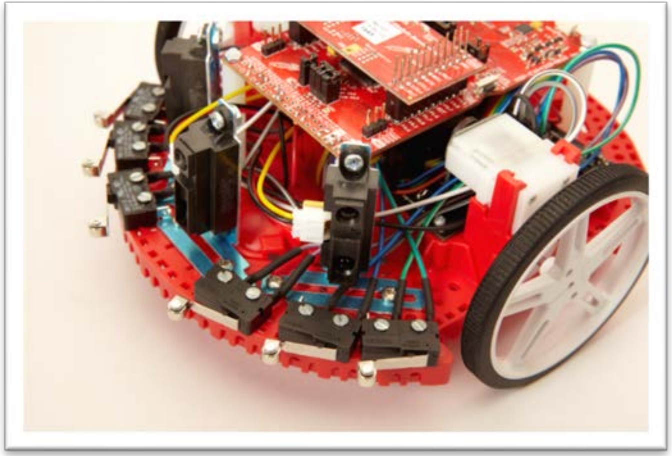
- Shared global
- Nonatomic (multistep) access
- At least one write

Summary

Real-Time Systems

- Hard
- Firm
- Soft





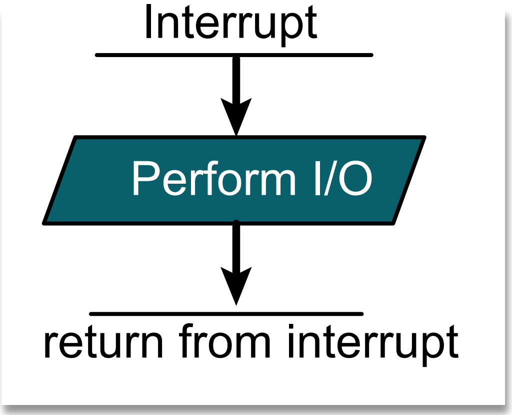
Module 14

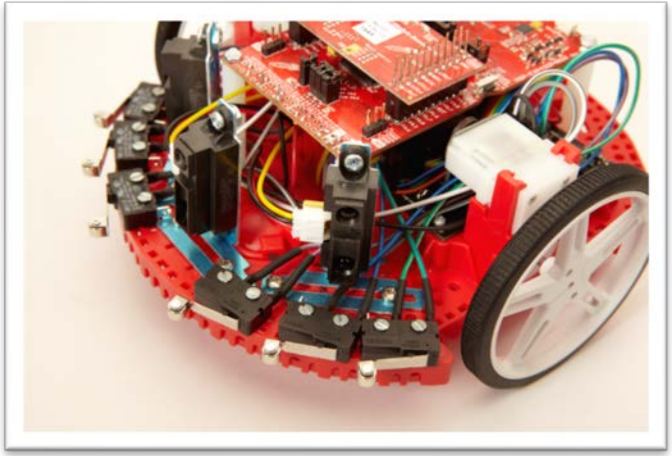
Lecture: Real-time Systems – Edge Triggered Interrupts

I/O Triggered Interrupts

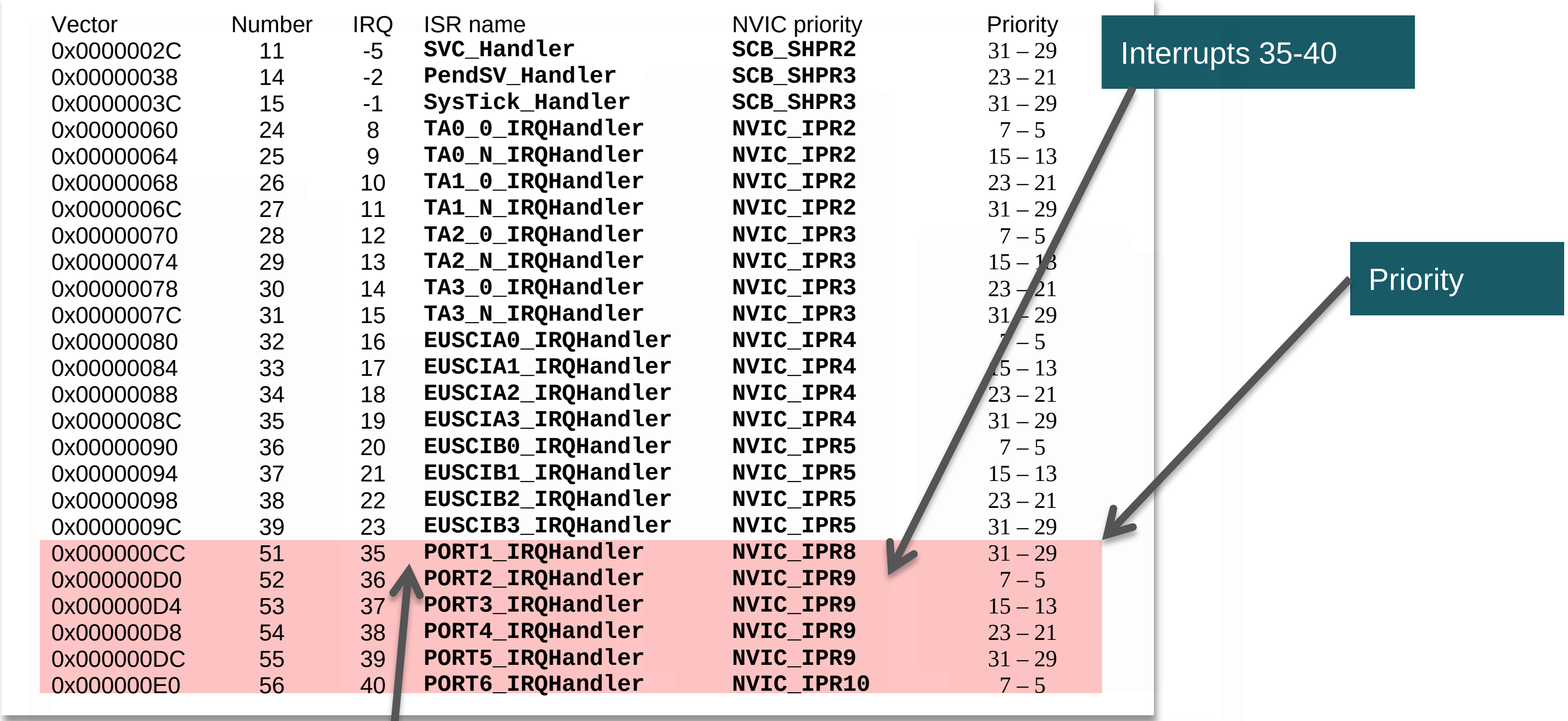
You will learn in this module

- Real-Time Systems
- Interrupts and the NVIC
  - Enable/disable
  - Priority
- Execute profiling
  - Scope or logic analyzer
- Edge-triggered interrupts
  - Select an edge
  - Polling versus vector
  - Acknowledgement





Interrupt Vectors, numbers, names, and priority



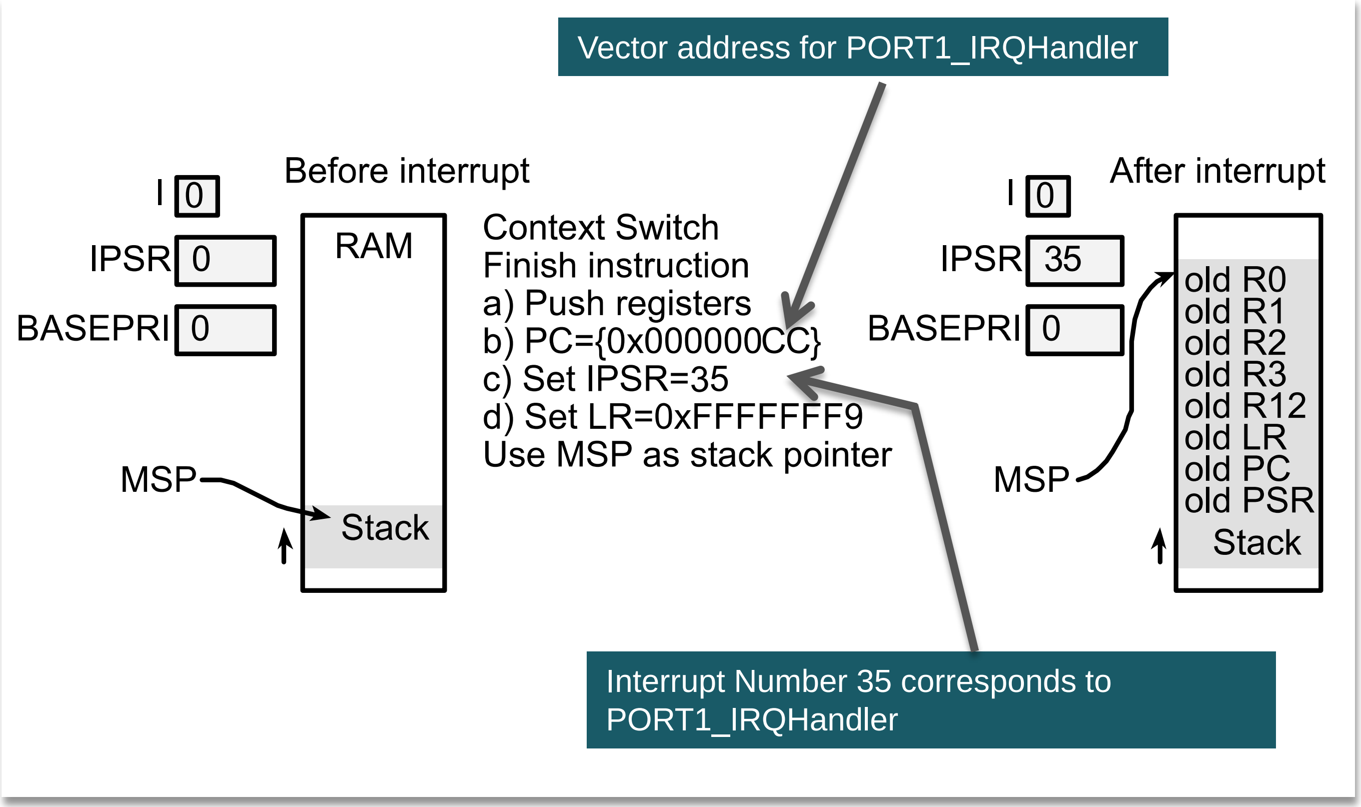
Vector	Number	IRQ	ISR name	NVIC priority	Priority
0x0000002C	11	-5	SVC_Handler	SCB_SHPR2	31 – 29
0x00000038	14	-2	PendSV_Handler	SCB_SHPR3	23 – 21
0x0000003C	15	-1	SysTick_Handler	SCB_SHPR3	31 – 29
0x00000060	24	8	TA0_0_IRQHandler	NVIC_IPR2	7 – 5
0x00000064	25	9	TA0_N_IRQHandler	NVIC_IPR2	15 – 13
0x00000068	26	10	TA1_0_IRQHandler	NVIC_IPR2	23 – 21
0x0000006C	27	11	TA1_N_IRQHandler	NVIC_IPR2	31 – 29
0x00000070	28	12	TA2_0_IRQHandler	NVIC_IPR3	7 – 5
0x00000074	29	13	TA2_N_IRQHandler	NVIC_IPR3	15 – 13
0x00000078	30	14	TA3_0_IRQHandler	NVIC_IPR3	23 – 21
0x0000007C	31	15	TA3_N_IRQHandler	NVIC_IPR3	31 – 29
0x00000080	32	16	EUSCIA0_IRQHandler	NVIC_IPR4	7 – 5
0x00000084	33	17	EUSCIA1_IRQHandler	NVIC_IPR4	15 – 13
0x00000088	34	18	EUSCIA2_IRQHandler	NVIC_IPR4	23 – 21
0x0000008C	35	19	EUSCIA3_IRQHandler	NVIC_IPR4	31 – 29
0x00000090	36	20	EUSCIB0_IRQHandler	NVIC_IPR5	7 – 5
0x00000094	37	21	EUSCIB1_IRQHandler	NVIC_IPR5	15 – 13
0x00000098	38	22	EUSCIB2_IRQHandler	NVIC_IPR5	23 – 21
0x0000009C	39	23	EUSCIB3_IRQHandler	NVIC_IPR5	31 – 29
0x000000CC	51	35	PORT1_IRQHandler	NVIC_IPR8	31 – 29
0x000000D0	52	36	PORT2_IRQHandler	NVIC_IPR9	7 – 5
0x000000D4	53	37	PORT3_IRQHandler	NVIC_IPR9	15 – 13
0x000000D8	54	38	PORT4_IRQHandler	NVIC_IPR9	23 – 21
0x000000DC	55	39	PORT5_IRQHandler	NVIC_IPR9	31 – 29
0x000000E0	56	40	PORT6_IRQHandler	NVIC_IPR10	7 – 5

Interrupts 35-40

Priority

```
void PORT1_IRQHandler(void){
  P1->IFG &= ~0x10; // clear flag4
}
```

Interrupt processing (review)



Context Switch
Finish instruction
a) Push registers
b) PC={0x000000CC}
c) Set IPSR=35
d) Set LR=0xFFFFFFF9
Use MSP as stack pointer

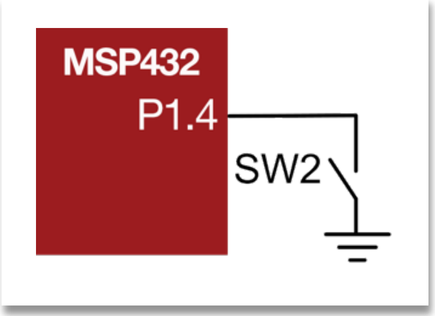
Single Switch Interface

Falling edge on touch

```
void EdgeTrigger_Init(void){
  FallingEdges4 = 0;
  P1->SEL0 &= ~0x10;
  P1->SEL1 &= ~0x10;  // configure P1.4 as GPIO
  P1->DIR &= ~0x10;   // make P1.4 input Button 2
  P1->REN |= 0x10;    // enable pull resistors
  P1->OUT |= 0x10;    // P1.4  pull-up
  P1->IES |= 0x10;    // P1.4  falling edge event
  P1->IFG &= ~0x10;   // clear flag4
  P1->IE |= 0x10;     // arm interrupt on P1.4
  NVIC->IP[8]=(NVIC->IP[8]&0x00FFFFFF)|0x40000000;
  NVIC->ISER[1] = 0x00000008; // enable
  EnableInterrupts();
}
```

Priority 2

Interrupt 35



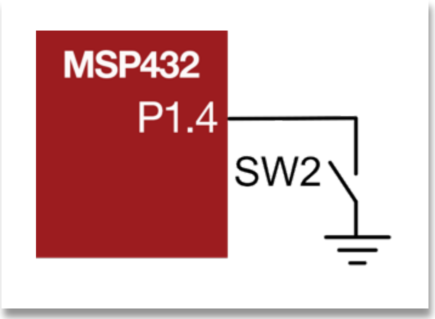
Single Switch Interface

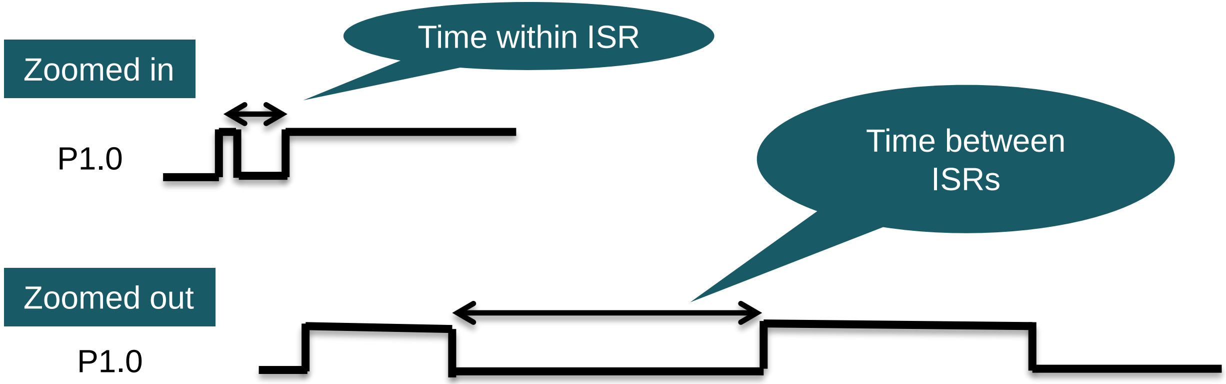
Falling edge on touch

```
void PORT1_IRQHandler(void){
  P1->OUT ^= 0x01;
  P1->OUT ^= 0x01;
  P1->IFG &= ~0x10; // clear flag4
  FallingEdges4 = FallingEdges4+1;
  P1->OUT ^= 0x01;
}
```

Acknowledge

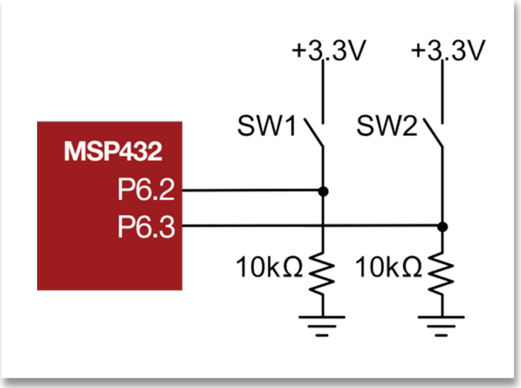
Profile

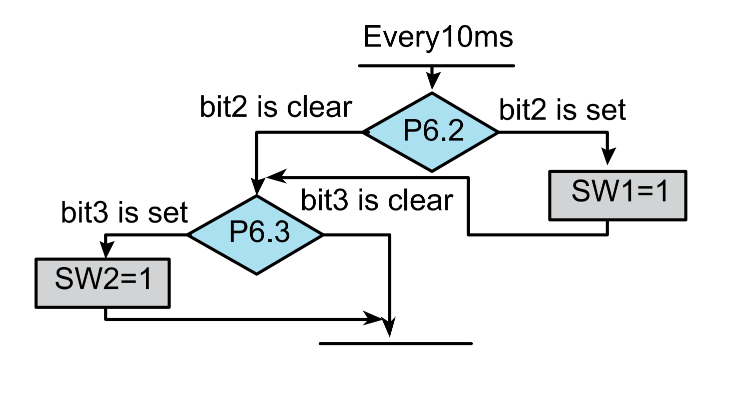




Two-Switch Periodic Polling Interface

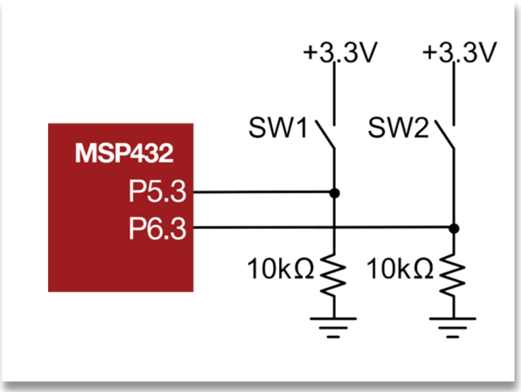
```
void Poll(void){// 10 ms
  if(P6->OUT&0x04){
    SW1 = 1;
  }
  if(P6->OUT&0x08){
    SW2 = 1;
  }
}
int main(void){
  Clock_Init48MHz();
  P6->DIR &= ~0x0C;
  TimerA2_Init(&Poll, 5000);
  EnableInterrupts();
  while(1){}
}
```





Two-Switch Vectored Interface

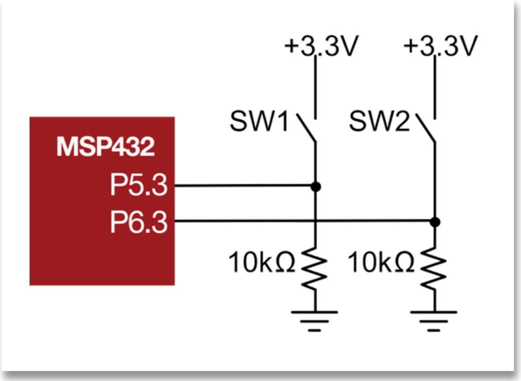
```
void VectorButtons_Init(void){
  P5->SEL0 &= ~0x08;
  P5->SEL1 &= ~0x08; // GPIO
  P5->DIR &= ~0x08;  // make in
  P5->IES &= ~0x08;  // rising edge event
  P5->IFG &= ~0x08;  // clear flag3
  P5->IE |= 0x08;    // arm interrupt
  NVIC->IP[9]=(NVIC->IP[9]&0x00FFFFFF)|0x40000000;
  NVIC->ISER[1] = 0x00000080;  // interrupt 39
  P6->SEL0 &= ~0x08;
  P6->SEL1 &= ~0x08; // GPIO
  P6->DIR &= ~0x08;  // make in
  P6->IES &= ~0x08;  // rising edge event
  P6->IFG &= ~0x08;  // clear flag3
  P6->IE |= 0x08;    // arm interrupt on P6.3
  NVIC->IP[10]=(NVIC->IP[10]&0xFFFFFF00)|0x00000040;
  NVIC->ISER[1] = 0x00000100;}  // interrupt 40
```



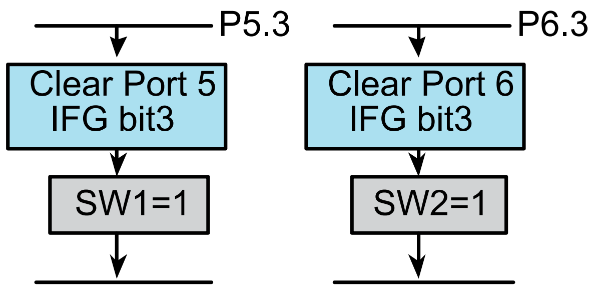
Two-Switch Vectored Interface

Vectored Interrupt

```
void PORT5_IRQHandler(void){
  P5->IFG &= ~0x08; // ack
  SW1 = 1; // signal
}
void PORT6_IRQHandler(void){
  P6->IFG &= ~0x08; // ack
  SW2 = 1; // signal
}
```

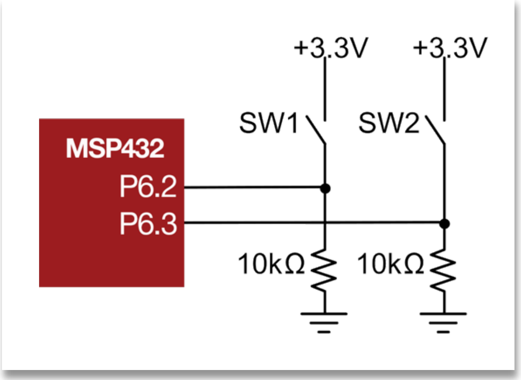


Vectored solution



Two-Switch Polled Interface

```
void PolledButtons_Init(void){
  P6->SEL0 &= ~0x0C;
  P6->SEL1 &= ~0x0C; // GPIO
  P6->DIR &= ~0x0C;  // make in
  P6->IES &= ~0x0C;  // rising edge event
  P6->IFG &= ~0x0C;  // clear flag3 and flag2
  P6->IE |= 0x0C;    // arm
  NVIC->IP[10]=(NVIC->IP[10]&0xFFFFFF00)
          |0x00000040;
  NVIC->ISER[1] = 0x00000100; // interrupt 40
}
```

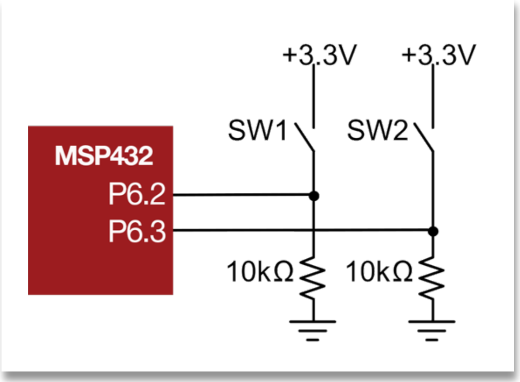


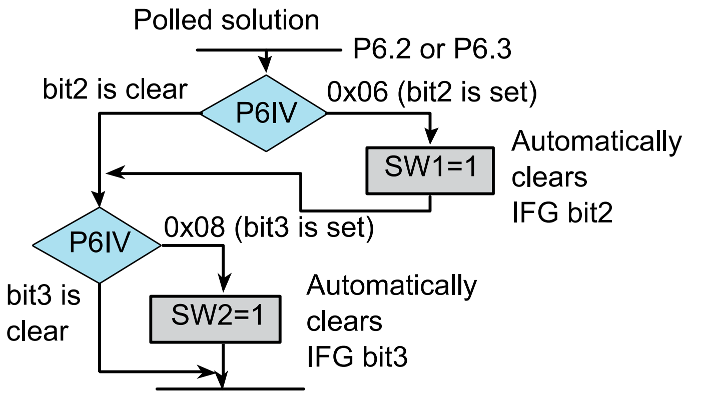
Two-Switch Polled Interface

```
void PORT6_IRQHandler(void){
uint8_t status;
  status = P6->IV;
  if(status==0x06){
    SW1 = 1; // signal
    status = P6->IV;
  }
  if(status==0x08){
    SW2 = 1;
  }
}
```

Poll

PxIV it will get the number (2*(*n*+1)) where *n* is the pin number of the lowest bit with a pending interrupt





Summary

Interrupts and the NVIC

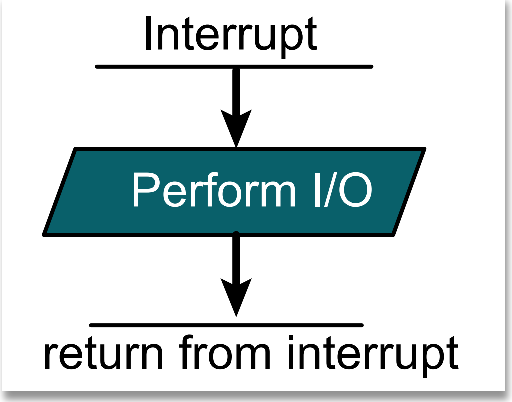
- Enable/disable
- Priority

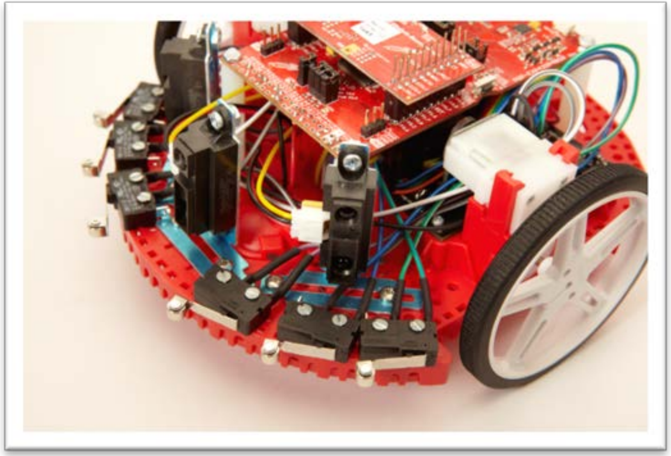
Execute profiling

- Scope or logic analyzer

Edge-triggered interrupts

- Select an edge
- Polling versus vector
- Acknowledgement





IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265