

TI-RSLK

Texas Instruments Robotics System Learning Kit



TEXAS INSTRUMENTS



Module 14

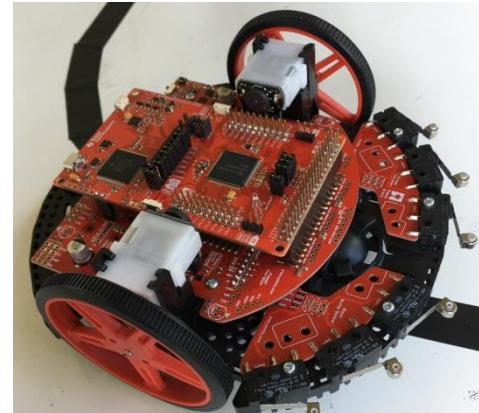
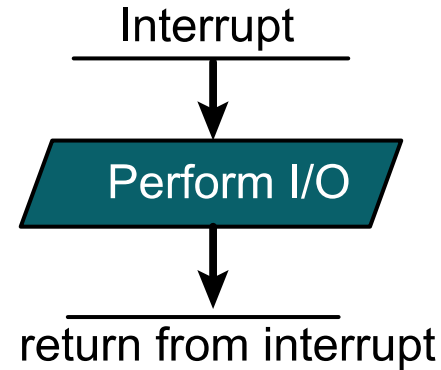
Lecture: Real-time Systems - Theory



Real-Time Systems

You will learn in this module

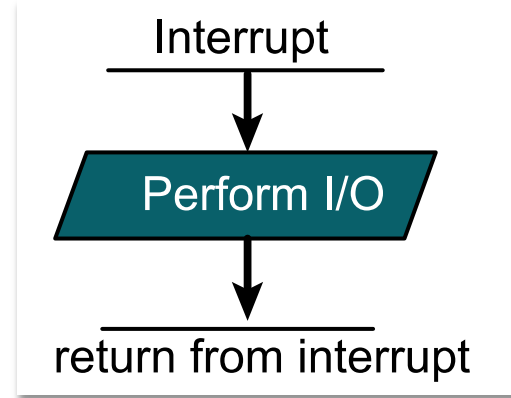
- Tasks
 - Periodic
 - Aperiodic
 - Sporadic
- Performance measures
 - Latency
 - Response time
- Real-Time Systems
 - Hard
 - Firm
 - Soft





Real-Time Systems

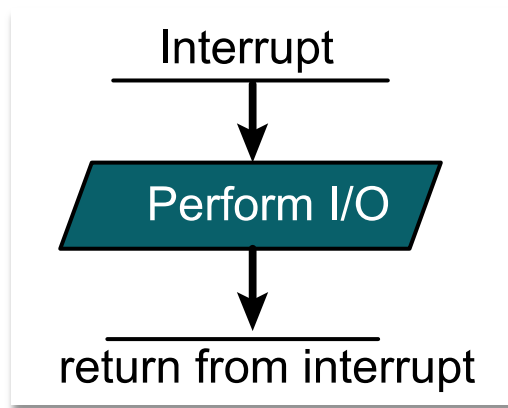
- Tasks
 - Periodic (sampling, digital controller)
 - Aperiodic (I/O)
 - Sporadic (faults)
- Latency
- Response time
- Priority





Real-Time Systems

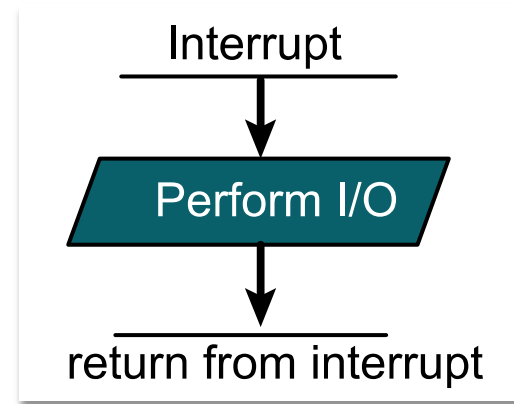
- Hard real time systems
 - Guaranteed bounded latency/response time





Real-Time Systems

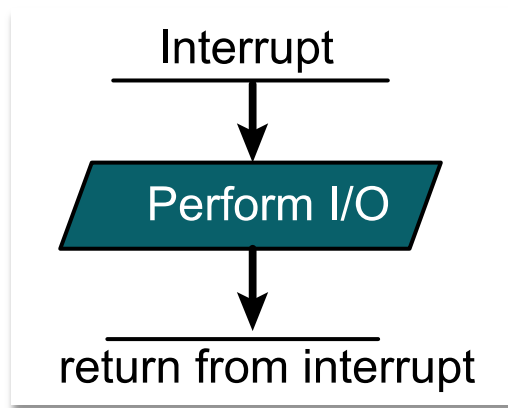
- Firm real time systems
 - Missed deadline loss of quality





Real-Time Systems

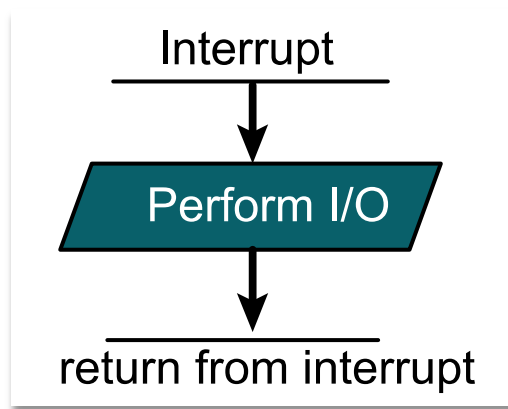
- Soft real time systems
 - Delayed response reduces value





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)

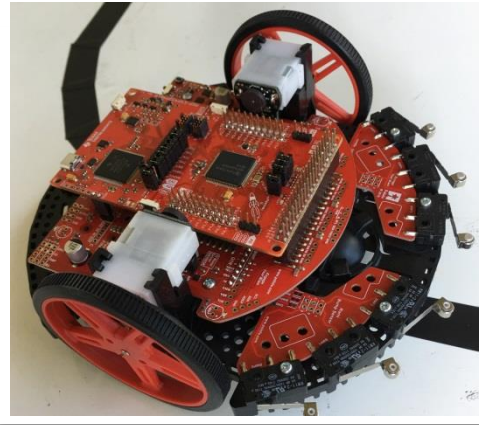
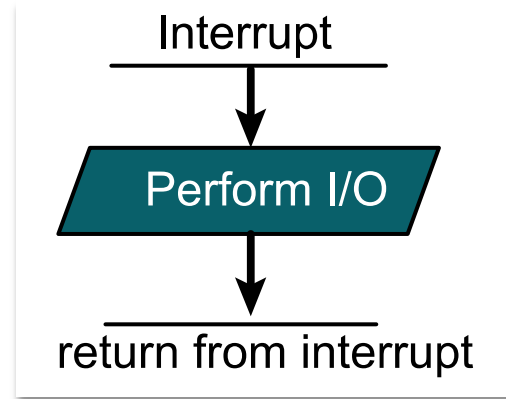
- Shared global
- Nonatomic (multistep) access
- At least one write



Summary

Real-Time Systems

- Hard
- Firm
- Soft





Module 14

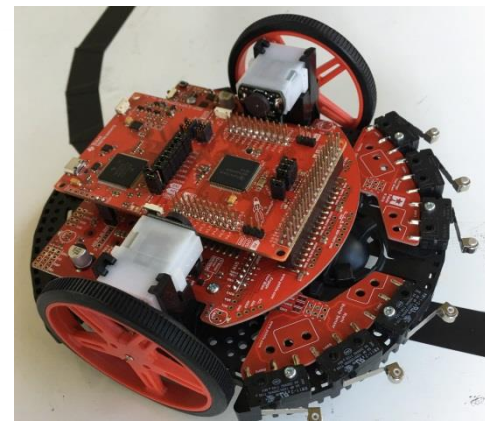
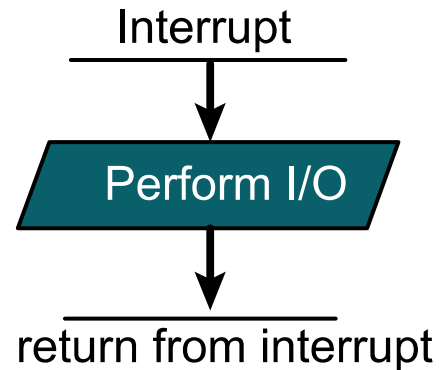
Lecture: Real-time Systems – Edge Triggered Interrupts



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

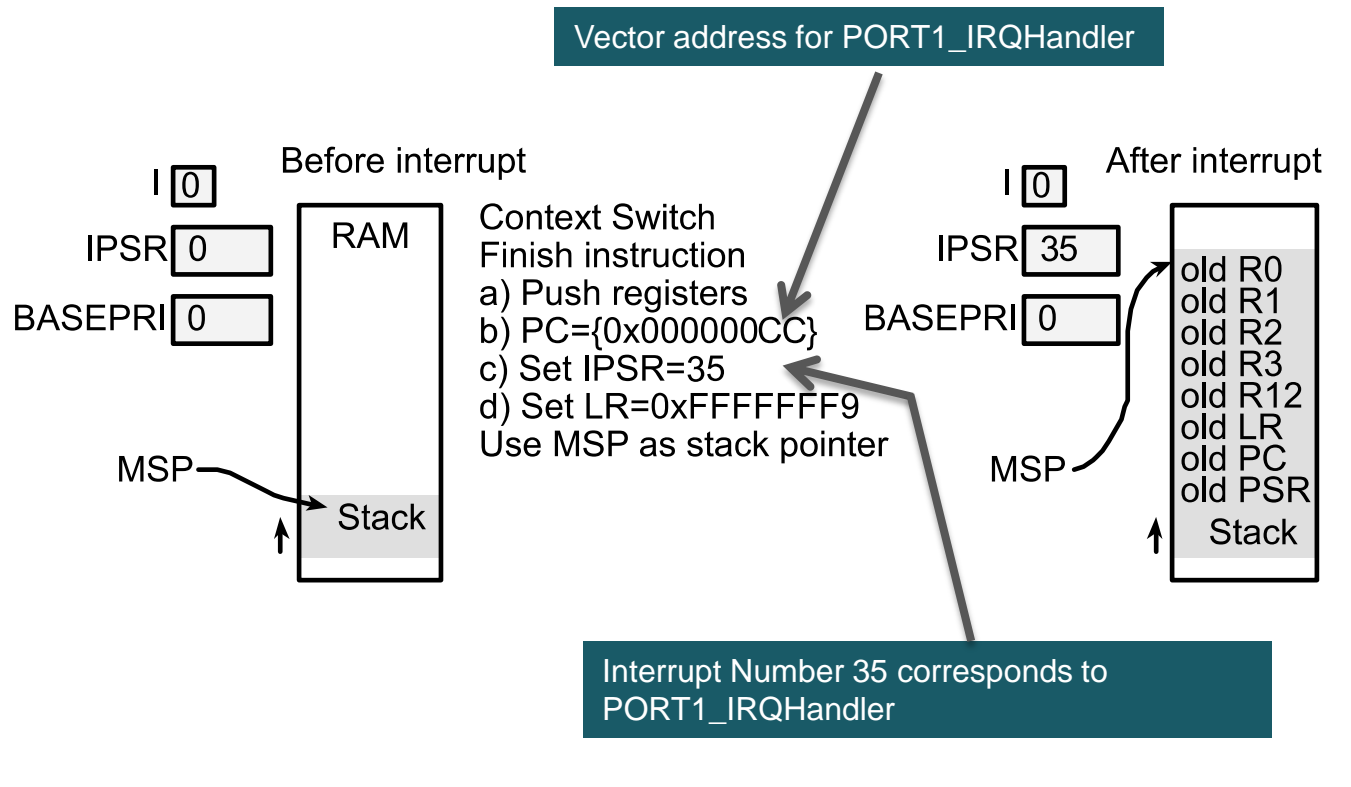
Interrupts 35-40

Priority

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



Interrupt processing (review)





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

MSP432

P1.4

SW2





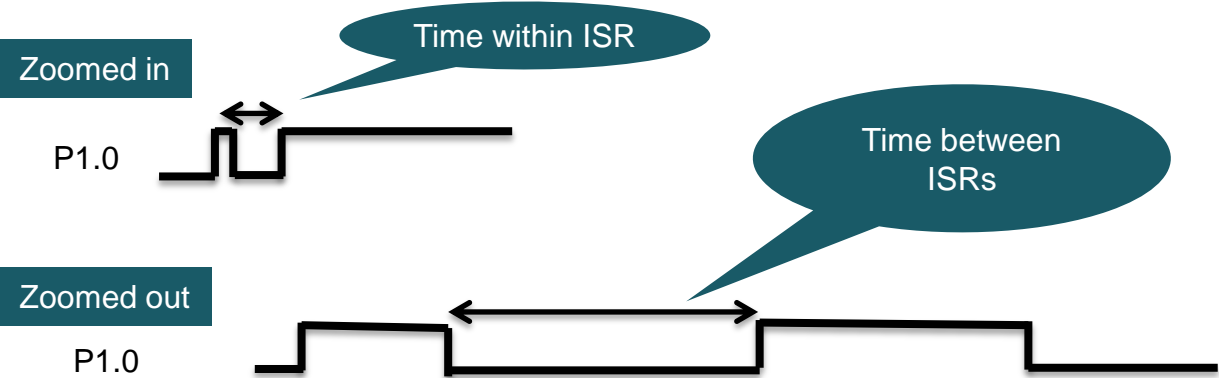
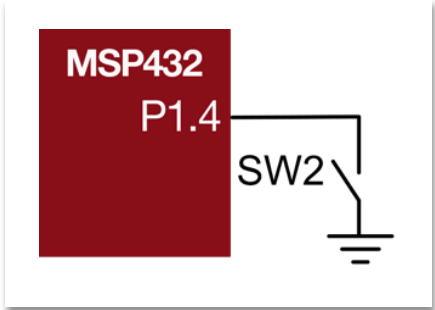
Single Switch Interface

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;  
}
```

Profile

Acknowledge

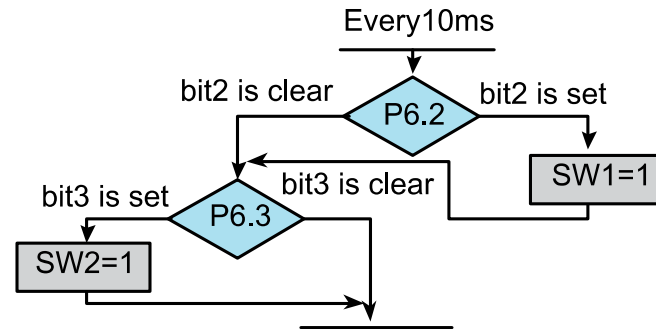
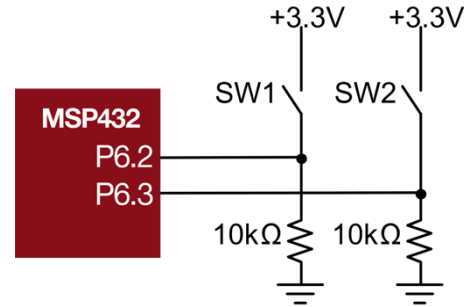




Two-Switch Periodic Polling Interface

```
void Poll(void) { // 10 ms
    if (P6->OUT & 0x04) {
        SW1 = 1;
    }
    if (P6->OUT & 0x08) {
        SW2 = 1;
    }
}

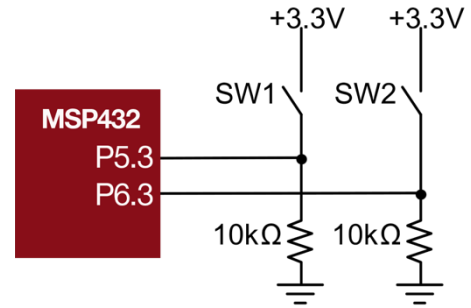
int main(void) {
    Clock_Init48MHz();
    P6->DIR &= ~0x0C;
    TimerA2_Init(&Poll, 5000);
    EnableInterrupts();
    while(1) {}
}
```





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]&0xFFFFFFF0)|0x00000040;
    NVIC->ISER[1] = 0x00000100;} // interrupt 40
```

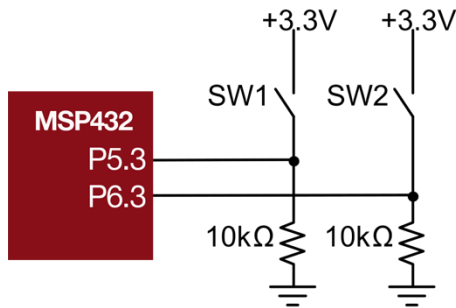




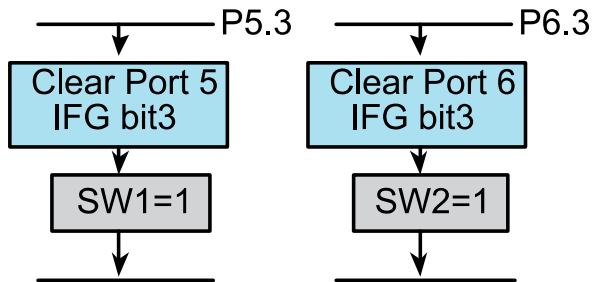
Two-Switch Vectored Interface

Vectored Interrupt

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



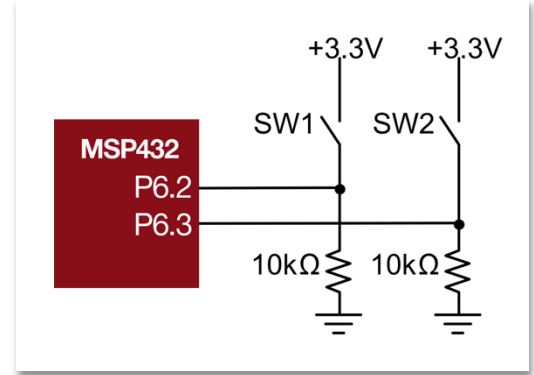
Vectored solution





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] & 0xFFFFF00)  
        | 0x00000040;  
    NVIC->ISER[1] = 0x00000100; // interrupt 40  
}
```



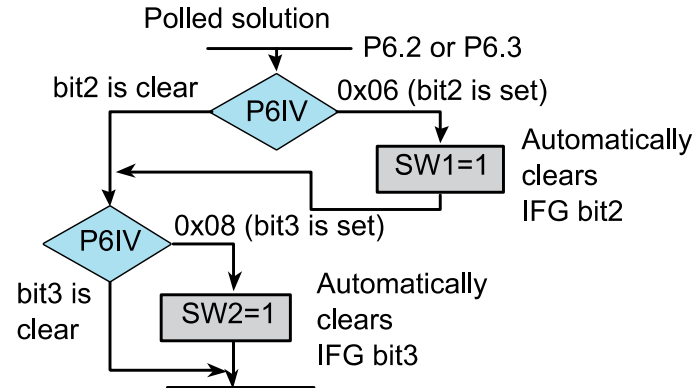
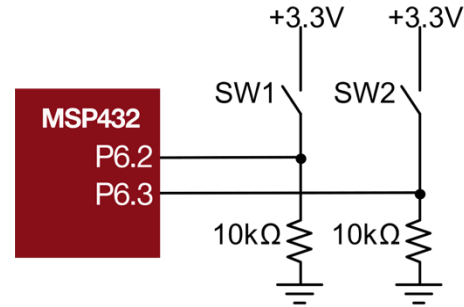


Two-Switch Polled Interface

```
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

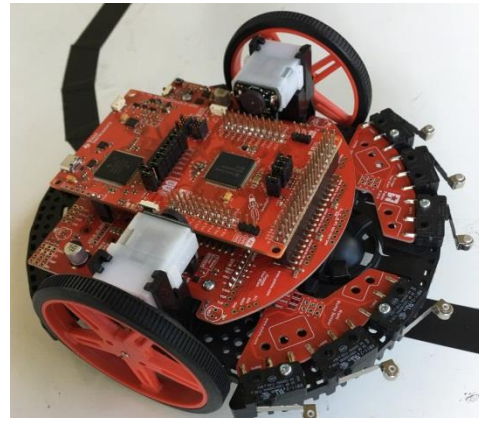
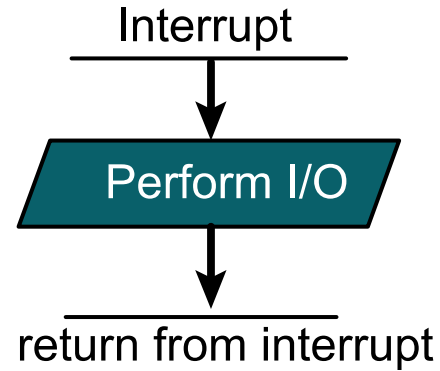
- Enable/disable
- Priority

Execute profiling

- Scope or logic analyzer

Edge-triggered interrupts

- Select an edge
- Polling versus vector
- Acknowledgement



ti.com/rslk



IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated