TI-RSLK MAX
Texas Instruments Robotics System Learning Kit
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

```
Interrupt

Perform I/O

return from interrupt
```
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

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

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

Interrupt Number 35 corresponds to PORT1_IRQHandler

Context Switch
Finish instruction
a) Push registers
b) PC=0x000000CC
c) Set IPSR=35
d) Set LR=0xFFFFFFFF9
Use MSP as stack pointer
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();
}
void PORT1_IRQHandler(void){
  P1->OUT ^= 0x01;
  P1->OUT ^= 0x01;
  P1->IFG &= ~0x10; // clear flag4
  FallingEdges4 = FallingEdges4+1;
  P1->OUT ^= 0x01;
}
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) {}
void VectorButtons_Init(void) {
    P5->SEL0 &= ~0x08;                // GPIO
    P5->SEL1 &= ~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;                // GPIO
    P6->SEL1 &= ~0x08; // rising edge event
    P6->IFG &= ~0x08; // clear flag3
    P6->IE |= 0x08; // arm interrupt on P6.3
    NVIC->IP[10]=(NVIC->IP[10]&0xFFFFF000)|0x00000040;
    NVIC->ISER[1] = 0x000000100;}  // interrupt 40
void PORT5_IRQHandler(void){
    P5->IFG &= ~0x08; // ack
    SW1 = 1; // signal
}
void PORT6_IRQHandler(void){
    P6->IFG &= ~0x08; // ack
    SW2 = 1; // signal
}
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] & 0xFFFFFFFF) | 0x00000040;
  NVIC->ISER[1] = 0x00000100; // interrupt 40
}
void PORT6_IRQHandler(void) {
    uint8_t status;
    status = P6->IV;
    if (status == 0x06) {
        SW1 = 1; // signal
    }
    status = P6->IV;
    if (status == 0x08) {
        SW2 = 1;
    }
}

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

### Poll

- **MSP432**
- **SW1**
- **SW2**
- **+3.3V**
- **10kΩ**

### Diagram

- **Poll**
- **P6IV**
- **0x06 (bit2 is set)**
- **0x08 (bit3 is set)**
- **SW1=1**
- **SW2=1**
- **bit2 is clear**
- **bit3 is clear**
- **Automatically clears IFG bit2**
- **Automatically clears IFG bit3**

**P6.2 or P6.3**
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 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