Module 16

Lecture: Tachometer - Input Capture
Tachometer

You will learn in this module

- Timer A
  - Clock input, prescale
  - Input capture

- Period Measurement
  - Precision
  - Range
  - Resolution

- Motor Performance
  - Speed
  - Time constant
Timer_A for input and output

Components

- **Pins**
  - Input capture
  - Output compare

- **Precision**
  - 16-bits

- **Resolution**
  - Clock period
  - Prescale

**Components**:

- **Input capture**
- **Output compare**

**Precision**

- 16-bits

**Resolution**

- Clock period
- Prescale

**MSP432**

- **Pins**
  - P7.3 TA0.0
  - P2.4 TA0.1
  - P2.5 TA0.2
  - P2.6 TA0.3
  - P2.7 TA0.4
  - P8.0 TA1.0
  - P7.7 TA1.1
  - P7.6 TA1.2
  - P7.5 TA1.3
  - P7.4 TA1.4
  - TA2.0 P8.1
  - TA2.1 P5.6
  - TA2.2 P5.7
  - TA2.3 P6.6
  - TA2.4 P6.7
  - TA3.0 P10.4
  - TA3.1 P10.5
  - TA3.2 P8.2
  - TA3.3 P9.2
  - TA3.4 P9.3

**SMCLK**

- 48 MHz / 4 = 12 MHz, 83.33 ns
Timer_A registers

0.0000 | 15-10 | 9-8 | 7-6 | 5-4 | 3 | 2 | 1 | 0 | Name
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA0CTL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>15-14</th>
<th>13-12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7-5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TA0CCTL0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0002</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0006</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0008</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000A</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000C</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000E</td>
<td>CM</td>
<td>CCIS</td>
<td>SCS</td>
<td>SCCI</td>
<td>CAP</td>
<td>OUTMOD</td>
<td>CCIE</td>
<td>CCI</td>
<td>OUT</td>
<td>COV</td>
<td>CCIFG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15-0

0.0010 | 16-bit counter | TA0R |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0012</td>
<td>16-bit Capture/Compare 0 Register</td>
<td>TA0CCR0</td>
</tr>
<tr>
<td>0.0014</td>
<td>16-bit Capture/Compare 1 Register</td>
<td>TA0CCR1</td>
</tr>
<tr>
<td>0.0016</td>
<td>16-bit Capture/Compare 2 Register</td>
<td>TA0CCR2</td>
</tr>
<tr>
<td>0.0018</td>
<td>16-bit Capture/Compare 3 Register</td>
<td>TA0CCR3</td>
</tr>
<tr>
<td>0.001A</td>
<td>16-bit Capture/Compare 4 Register</td>
<td>TA0CCR4</td>
</tr>
<tr>
<td>0.001C</td>
<td>16-bit Capture/Compare 5 Register</td>
<td>TA0CCR5</td>
</tr>
<tr>
<td>0.001E</td>
<td>16-bit Capture/Compare 6 Register</td>
<td>TA0CCR6</td>
</tr>
</tbody>
</table>

15-3 | 2-0

0.0020 | TA0EX0 |
|--------|--------|

0.002E | TA0IV |
|--------|--------|
## Timer_A registers

<table>
<thead>
<tr>
<th>Name</th>
<th>15-10</th>
<th>9-8</th>
<th>7-6</th>
<th>5-4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASSEL</td>
<td>ID</td>
<td>MC</td>
<td>TA0CTL</td>
<td>TA0CCLR</td>
<td>TA0CAIE</td>
<td>TA0CCFG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 15-14 13-12 11 10 9 8 7-5 4 3 2 1 0

- **CM**: CM, CCIS, SCS, SCCI, CAP, OUTMOD, CCE, CCI, OUT, COV, CCIFG
- **TA0CCCR0**
- **TA0CCR1**
- **TA0CCR2**
- **TA0CCR3**
- **TA0CCR4**
- **TA0CCR5**
- **TA0CCR6**

### 15-3 2-0

- **TA0EX0**
- **TA0IV**

```c
// 01, 00, 1, 0, 0, 1, 000, 1, 0, 0, 0, 0
TIMER_A0->CCTL[0] = 0x4910;
```

### 15-0

- **16-bit counter**
- **TA0R**
- **TA0CCR0**
- **TA0CCR1**
- **TA0CCR2**
- **TA0CCR3**
- **TA0CCR4**
- **TA0CCR5**
- **TA0CCR6**

### 15-3 2-0

- **TA0EX0**
- **TA0IV**
Clock and prescale

<table>
<thead>
<tr>
<th>TASSEL</th>
<th>Selected Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>TAxCLK</td>
</tr>
<tr>
<td>01</td>
<td>ACLK</td>
</tr>
<tr>
<td>10</td>
<td>SMCLK</td>
</tr>
<tr>
<td>11</td>
<td>INCLK</td>
</tr>
</tbody>
</table>

Resolution = $T \times 2^{ID} \times (TAIDEX+1)$
Range = Precision $\times$ Resolution
## Timer_A Modes

<table>
<thead>
<tr>
<th>CM</th>
<th>Capture mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No capture</td>
</tr>
<tr>
<td>01</td>
<td>Capture on rising edge</td>
</tr>
<tr>
<td>10</td>
<td>Capture on falling edge</td>
</tr>
<tr>
<td>11</td>
<td>Capture on both rising and falling edges</td>
</tr>
</tbody>
</table>

- **CM**: bits 15-14 = 01 — capture on rising edge
- **CCIS**: bits 13-12 = 00 — capture input on CCIxA
- **SCS**: bit 11 = 1 — synchronous capture source
- **CAP**: bit 8 = 1 — capture mode
- **CCIE**: bit 4 = 1 — arm for interrupt
- **CCIFG**: bit 0 = 0 — trigger, set by hardware, cleared by software
0) Halt the timer (MC=00),
1) Set the timer clock and prescale,
2) Set submodule 0 to capture rising, arm interrupt
3) Set the priority in the correct NVIC Priority register
4) Enable the interrupt in the NVIC Interrupt Enable register
5) Reset the timer and start it in up mode
6) Enable interrupts (in the main program after all devices initialized)
Period Measurement Example

Stripe detected

- Acknowledge
- Measure period
void TimerA0Capture_Init(void){
    P7->SEL0 |= 0x08; // configure P7.3 as TA0CCP0
    P7->SEL1 &= ~0x08; // make P7.3 in
    TIMER_A0->CTL &= ~0x0030; // halt Timer A0
    // bits9-8=10, clock source to SMCLK
    // bits7-6=00, input clock divider /1
    // bits5-4=00, stop mode
    TIMER_A0->CTL = 0x0200; // bits15-14=01, capture on rising edge
    // bits13-12=00, capture/compare input on CCI0A
    // bit11=1, synchronous capture source
    // bit8=1, capture mode
    // bit4=1, enable capture/compare interrupt
    // bit0=0, clear capture/compare interrupt pending
    TIMER_A0->CCTL[0] = 0x4910;
    TIMER_A0->EX0 &= ~0x0007; // configure for input clock divider /1
    NVIC->IP[2] = (NVIC->IP[2] & 0xFFFFFFFF) | 0x00000040; // priority 2
    NVIC->ISER[0] = 0x00000100; // enable interrupt 8 in NVIC
    TIMER_A0->CTL |= 0x0024; // reset and start Timer A0 in continuous up mode
}

Interrupts enabled in the main program after all devices initialized
```c
uint16_t First, Period;
void TA0_0_IRQHandler(void){
    TIMER_A0->CCTL[0] &= ~0x0001;
    // acknowledge capture/compare interrupt 0
    Period = TIMER_A0->CCR[0] - First;
    First = TIMER_A0->CCR[0];
}
```

**Period Measurement**

- **Interrupt on rising edge**
- **Stripe detected**
  - `Period = TA0CCR0-FIRST`
  - `First = TA0CCR0`

**Diagram**

- TA0R
  - FF00
  - FF01
  - FF02
  - FFFE
  - FFFF
  - 0000
  - 0001
  - 0077
  - 0078
  - 0079

- **Period**
  - Precision: 16 bits
  - Resolution: 83.33ns
  - Range: 10us to 5.4ms
Summary

Measuring speed with input capture

- Timer_A
  - Clock input,
  - Prescale
  - Input capture

- Period Measurement
  - Precision
  - Range
  - Resolution

- Motor Performance
  - Speed
  - Time constant
Module 16

Lecture: Tachometer - Interface
Tachometer

You will learn in this module

- Encoder
  - Motor speed
  - Motor direction
- Motor Performance
  - Speed
  - Time constant
Encoders

Gearmotor for the Romi/TI-RSLK has an encoder already installed
360 pulses/rotation

Speed (rpm) =
(1 rotation/360 pulses) *
(1,000,000,000 ns/sec) *
(60 sec/min)
/(Period * 83.33 ns/pulse)

\[ \text{Speed} = \frac{2,000,000}{\text{Period}} \]
Lab 5:
- Plug gear motor assembly with encoder as you build the TI-RSLK Chassis board and the robot
MSP432 Planning

P10.4 Capture

P10.5 Capture

P5.0 GPIO

P5.2 GPIO
TI-RSLK Chassis board – Interface circuit MSP432-Encoder

Direction on P5.0, P5.2
- 1 = forward
- 0 = backward

Input capture on P10.4, P10.5

Tachometer Left
Vcc
OUT A
OUT B
Gnd

Tachometer Right
Vcc
OUT A
OUT B
Gnd

+3.3V=R3V3

MSP432-Port5

MSP432-Port10

P5.0
P5.1
P5.2
P5.4
P5.5
P5.6
P5.7

P10.0
P10.1
P10.2
P10.3
P10.4
P10.5
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>0x00000002C</td>
<td>11</td>
<td>-5</td>
<td>SVC_Handler</td>
<td>SCB_SHPR2</td>
<td>31 – 29</td>
</tr>
<tr>
<td>0x000000038</td>
<td>14</td>
<td>-2</td>
<td>PendSV_Handler</td>
<td>SCB_SHPR3</td>
<td>23 – 21</td>
</tr>
<tr>
<td>0x00000003C</td>
<td>15</td>
<td>-1</td>
<td>SysTick_Handler</td>
<td>SCB_SHPR3</td>
<td>31 – 29</td>
</tr>
<tr>
<td>0x000000060</td>
<td>24</td>
<td>8</td>
<td>TA0_0_IRQHandler</td>
<td>NVIC_IPR2</td>
<td>7 – 5</td>
</tr>
<tr>
<td>0x000000064</td>
<td>25</td>
<td>9</td>
<td>TA0_N_IRQHandler</td>
<td>NVIC_IPR2</td>
<td>15 – 13</td>
</tr>
<tr>
<td>0x000000068</td>
<td>26</td>
<td>10</td>
<td>TA1_0_IRQHandler</td>
<td>NVIC_IPR2</td>
<td>23 – 21</td>
</tr>
<tr>
<td>0x00000006C</td>
<td>27</td>
<td>11</td>
<td>TA1_N_IRQHandler</td>
<td>NVIC_IPR2</td>
<td>31 – 29</td>
</tr>
<tr>
<td>0x000000070</td>
<td>28</td>
<td>12</td>
<td>TA2_0_IRQHandler</td>
<td>NVIC_IPR3</td>
<td>7 – 5</td>
</tr>
<tr>
<td>0x000000074</td>
<td>29</td>
<td>13</td>
<td>TA2_N_IRQHandler</td>
<td>NVIC_IPR3</td>
<td>15 – 13</td>
</tr>
<tr>
<td>0x000000078</td>
<td>30</td>
<td>14</td>
<td>TA3_0_IRQHandler</td>
<td>NVIC_IPR3</td>
<td>23 – 21</td>
</tr>
<tr>
<td>0x00000007C</td>
<td>31</td>
<td>15</td>
<td>TA3_N_IRQHandler</td>
<td>NVIC_IPR3</td>
<td>31 – 29</td>
</tr>
<tr>
<td>0x000000080</td>
<td>32</td>
<td>16</td>
<td>EUSCIA0_IRQHandler</td>
<td>NVIC_IPR4</td>
<td>7 – 5</td>
</tr>
<tr>
<td>0x000000084</td>
<td>33</td>
<td>17</td>
<td>EUSCIA1_IRQHandler</td>
<td>NVIC_IPR4</td>
<td>15 – 13</td>
</tr>
<tr>
<td>0x000000088</td>
<td>34</td>
<td>18</td>
<td>EUSCIA2_IRQHandler</td>
<td>NVIC_IPR4</td>
<td>23 – 21</td>
</tr>
<tr>
<td>0x00000008C</td>
<td>35</td>
<td>19</td>
<td>EUSCIA3_IRQHandler</td>
<td>NVIC_IPR4</td>
<td>31 – 29</td>
</tr>
<tr>
<td>0x000000090</td>
<td>36</td>
<td>20</td>
<td>EUSCIB0_IRQHandler</td>
<td>NVIC_IPR5</td>
<td>7 – 5</td>
</tr>
</tbody>
</table>

```c
void TA3_0_IRQHandler(void)
{
    TIMER_A3->CCTL[0] &= ~0x0001; // ack
    // body
}

void TA3_N_IRQHandler(void)
{
    TIMER_A3->CCTL[1] &= ~0x0001; // ack
    // body
}
```
Motor Model

Duty cycle, $X(t)$ → Motor → Speed $Y(t)$

Linear model
- Gain, $m$
- Time constant, $\tau$

Duty cycle, $X(s)$ → Motor → Speed $Y(s)$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{m}{1+s\tau}$$

Motors are not linear
Friction affects everything

$$y(t) = S_0 + \Delta S \cdot e^{-t/\tau}$$
Motor Time Constant

Duty cycle, X(t) → Motor → Speed Y(t)

![Graph showing speed vs. time for a motor with time constant τ, plotted as y(t) = S₀ + ΔS e⁻ᵗ/τ.](image)

25% to 50%

Time (ms)

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Speed (rpm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>150</td>
<td>120</td>
</tr>
<tr>
<td>200</td>
<td>140</td>
</tr>
<tr>
<td>250</td>
<td>160</td>
</tr>
</tbody>
</table>

τ

y(t) = S₀ + ΔS e⁻ᵗ/τ
Summary

Measuring speed with input capture

- Timer_A
  - Clock input,
  - Prescale
  - Input capture

- Period Measurement
  - Precision
  - Range
  - Resolution

- Motor Performance
  - Speed
  - Time constant
Module 16
Lecture: Tachometer - Odometry
You will learn in this module

- How the wheels affect position and angle
  - Robot wheelbase
  - Wheel circumference
  - Wheel rotations

- Geometry used in odometry
  - Wheel counts as input
  - Old position and angle as input
  - New position and angle as output

- Limitations of odometry
  - Calibration
  - Accumulation of error
Fixed point

What
Value = integer*constant

Why
Integer math is faster than floating point
32-bit floating point only has 23 bits of precision

When
Need non-integer values
Range of values is known and small

How
Choose a constant, e.g., 0.0001cm
Choose an integer type, e.g., int32_t
e.g., range will be ±200000 cm
e.g., resolution will be 0.0001cm
Physical dimensions of the robot

Number of slots/rotation, $n = 360$
Wheel diameter, $d = 70000$ in units of $0.0001\text{cm}$
Wheelbase (distance between wheels), $w = 140000$ in units of $0.0001\text{cm}$
Wheel circumference, $c = \pi d = 219911$ in units of $0.0001\text{cm}$
Robot position $(x, y)$ in units of $0.0001\text{cm}$
Robot heading $\theta$ in units of $2\pi/16384$ radians
We define the center of gravity (cog) as a point equidistant from the pivot points
When to run odometry?

Tachometers interrupt 360 times/rotation
ISRs maintain \( L\text{count} \), \( R\text{count} \)

Odometry is math that
• Takes \( L\text{count} \), \( R\text{count} \) as inputs
• Updates \((x, y, \theta)\)

You should not run odometry on each count.
Option 1: Periodic
• Run the odometry calculations at fixed rate.
• Choose a rate so the counts are about 10 to 40.
• If either count goes above 40, you could increase rate.
• If both counts drop below 10, you could decrease rate.

Option 2: As needed
• Run the odometry when either count gets to a fixed limit, e.g., 20.

Errors if counts are too big
Errors if counts are too small
Odometry

- Tachometer used to count rotations
- Predict position from wheel rotations
- Assume the wheels do not slip

Examples
If both wheels move the same number of counts, it goes in a straight line.

Arc distance each wheel moves
- \( Lr = L\text{count} \times \frac{c}{n} \)
- \( Rr = R\text{count} \times \frac{c}{n} \)
Odometry

- Tachometer used to count rotations
- Predict position from wheel rotations
- Assume the wheels do not slip

Examples
If one wheel goes forward the same count as the other goes back, it pivots about center of the robot.

Arc distance each wheel moves
- \( L_r = L_{count} \times \frac{c}{n} \)
- \( R_r = R_{count} \times \frac{c}{n} \)
Odometry

- Tachometer used to count rotations
- Predict position from wheel rotations
- Assume the wheels do not slip

Examples
If one wheel moves but the other does not, it pivots about single contact point of the stationary wheel

Arc distance each wheel moves
- $L_r = L_{count} \times c/n$
- $R_r = R_{count} \times c/n$
Odometry

- Tachometer used to count rotations
- Predict position from wheel rotations
- Assume the wheels do not slip

Examples
If one wheel moves more than the other, there will be both a motion and a rotation.

Arc distance each wheel moves
- $L_r = L_{\text{count}} \times c/n$
- $R_r = R_{\text{count}} \times c/n$
Odometry: find the pivot point

Inputs: \( L\text{count}, R\text{count} \)

Arc distance each wheel moves
- \( Lr = L\text{count}*c/n \)
- \( Rr = R\text{count}*c/n \)

Assume
- \( Rr \) and \( Lr \) are both positive
- \( Rr > Lr \)

Using similar triangles
- \( \frac{L}{Lr} = \frac{L+w}{Rr} \)
- \( \frac{L}{Lr} - \frac{L}{Rr} = \frac{w}{Rr} \)
- \( L \cdot Rr - L \cdot Lr = w \cdot Lr \)
- \( L = w \cdot Lr/(Rr - Lr) \)
Odometry: find the change in distance

The exact calculation for position change is

\[ dz = (L+w/2)\tan(d\theta/2) \]

If \( d\theta \) is small, \( \tan(d\theta/2) \approx d\theta/2 \)

\[ dz = (L+w/2) \times (d\theta/2) \]
Odometry: break move into two parts

First part of the move
- \[ x = x + dz \cdot \cos(\theta) \] (0.0001cm)
- \[ y = y + dz \cdot \sin(\theta) \] (0.0001cm)
- \[ \theta = \theta + d\theta \] (2\pi/16384 radian)

Second part of the move
- \[ x = x + dz \cdot \cos(\theta) \] (0.0001cm)
- \[ y = y + dz \cdot \sin(\theta) \] (0.0001cm)
Odometry: errors accumulate

Error in tachometer
- Extra counts
- Missing counts

Error in constants
- $w$ wheelbase
- $c$ wheel circumference

Errors in calculations
- $\tan(d\theta/2) \approx d\theta/2$
- $\sin(\theta)$
- $\cos(\theta)$
Summary

Odometry
- Tachometer input
  - Robot wheelbase
  - Wheel circumference
- Where am I?
  - Position
  - Angle
- Limitations
  - Calibration
  - Accumulation of errors
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