Using the HT66F45x0 PTM Bidirectional Capture Input Application Note

Introduction

The HT66F45x0 family of MCUs provides a range of Timer Modules (TMs) such as Periodic TM (PTM) and Standard TM (STM). These TMs integrate both Timer and PWM functions, providing operations such as timing/counter, PWM, single pulse output and capture input as well as other functions such as compare matching output.

In this application note, the HT66F4530 is taken as an example to illustrate the use of the PTM bidirectional capture mode in a practical example.

Functional Description

PTM function description: A 10-bit up-counter and a comparator are located at the heart of the PTM. The specific structure is shown below.

Note: As the HT66F4530 has only one PTM, therefore n=0. The HT66F45x0 has no PTPnI, so the input capture mode can only select PTCKn.

The PTM clock source can be selected by PTnCK[2:0]. The clock source can be selected from the system frequency (fSYS), the high frequency clock source (fH), the low frequency clock source (fSUB) or the external pin PTCKn.

The PTM contains two internal comparators, CCRA and CCRP. The CCRA and CCRP bits are the same as the counter and can be compared with all of the counter bits.
The PTM has five operating modes such as compare match output, capture input, PWM output or single pulse output and timer/counter. All operating mode settings are achieved by setting the associated register.

Operating Principles

PTM Register Description

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOC0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0PAU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOCK2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOCK1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOCK0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOC1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0M1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0M0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0IO1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0IO0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0CC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0POL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0CAPTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0CCCLR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D0</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
</tr>
<tr>
<td>PTM0D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0D7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTOCMD</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
</tr>
<tr>
<td>PTM0AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0AH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCR</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
</tr>
<tr>
<td>CCRA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTM0RPL</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
</tr>
<tr>
<td>PTM0RPH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PTM0C0 Register

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>PT0PAU</td>
<td>PT0CK2</td>
<td>PT0CK1</td>
<td>PT0CK0</td>
<td>PT0ON</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **PT0PAU: PTM0 Counter Pause Control Bit**
  - 0: Run
  - 1: Pause

- **PT0CK2~PT0CK0: Select PTM0 Counter Clock**
  - 000: \( f_{SYS}/4 \)
  - 001: \( f_{SYS} \)
  - 010: \( f_{i}/16 \)
  - 011: \( f_{i}/64 \)
  - 100: \( f_{SUB} \)
  - 101: \( f_{SUB} \)
  - 110: PTCK0 rising edge
  - 111: PTCK0 falling edge
  Note: \( f_{SYS} \) is the system clock, \( f_{i} \) and \( f_{SUB} \) are the other internal clock sources. Refer to the datasheet for more information

- **PT0ON: PTM0 Counter On/Off Control Bit**
  - 0: Off
  - 1: On
  Note: Setting this bit high enables the counter to run. Clearing this bit disables PTM0 to reduce power consumption.

PTM0C1 Register

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>PT0M1</td>
<td>PT0M0</td>
<td>PT0IO1</td>
<td>PT0IO0</td>
<td>PT0CC</td>
<td>PT0POL</td>
<td>PT0CAPTS</td>
</tr>
<tr>
<td></td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>POR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **PT0M1~PT0M0: Select PTM0 Operating Mode Bit**
  - 00: Compare Match Output Mode
  - 01: Capture Input Mode
  - 10: PWM Output Mode or Single Pulse Output Mode
  - 11: Timer/Counter Mode
• PT0IO1~PT0IO0: Select PTM0 External Pin PTP0 or PTCK0 function
  Compare Match Output Mode
  00: No change
  01: Output low
  10: Output high
  11: Toggle output

PWM Output Mode/Single Pulse Output Mode
  00: Forced inactive state
  01: Forced active state
  10: PWM output
  11: Single pulse output

Capture Input Mode
  00: PTCK0 rising edge capture
  01: PTCK0 falling edge capture
  10: PTCK0 dual edge capture
  11: Disable input capture

Timer/Counter Mode
  Not used

• PT0OC: PTM0 PTP0 Output Control Bit
  Compare Match Output Mode
  0: Initial low
  1: Initial high

PWM Output Mode/Single Pulse Output Mode
  0: Active low
  1: Active high

PT0POL: PTM0 PTP0 Output Polarity Control Bit
  0: Non-invert
  1: Invert

PT0CAPTS: Select PTM0 Capture Trigger Source
  0: Undefined
  1: PTCK0 pin source

Note: When in the capture input mode this bit must be set high

• PT0CCLR: Select PTM0 Counter Clear Condition
  0: PTM0 Comparator P Match
  1: PTM0 Comparator A Match

### PTM0D Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>R/W</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>POR</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### D9~D0: PTM0 10-bit Counter Register

### PTM0A Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>R/W</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>POR</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### D9~D0: PTM0 10-bit CCRA Register
### Hardware Description

#### Capture Input Mode Flow

The input capture mode is often used for pulse width measurement. Its main function is to capture a changing event on an external signal and at that point save the current value of the internal counter. From the saved values the pulse width can be obtained.

To operate in capture input mode, PTM0 must set the PT0M[1:0] bits in the PTM0C1 register to "01". The external signal on the "PTCK0" pin is selected by setting the PT0CAPTS bit of the PTM0C1 register. Select the required trigger edge type, such as rising edge, falling edge, double edge or no trigger, by setting the PT0IO[1:0] bits in the PTM0C1 register. After the setup is complete, change the PT0ON bit from low to high which will start the counter.

When a valid edge transition occurs on the PTCK0 pin, the current value of the counter is stored in the CCRA register and generates a PTM0 CCRA interrupt. Regardless of which edge transition occurs on the PTCK0 pin, the counter will continue to run until the PT0ON
bit experiences a falling edge transition. When a CCRP compare match occurs, the counter will be reset to zero. Here the value of CCRP controls the maximum value of the counter. A PTM0 interrupt is also generated when a CCRP compare match occurs. The number of CCRP overflow interrupt signals can be stored to measure long pulse widths.

If the PT0IQ[1:0] bits are both set high, no capture operation will occur regardless of which edge transition occurs on the PTCK0 pin, however the counter will continue to run.

When the PTCK0 pin is shared with other functions, then PTM0 needs to take care if it is operating in the input capture mode. If the pin is set to an output, then any level shift on this pin may initiate an input capture operation. The PT0CCLR, PT0OC, and PT0POL bits are not used in this mode.

Application Circuit Diagram

Description:
1. In the capture input mode the PTM selects PTCK0 as the input signal.
2. The STM is output on the STP0 pin in the compare match output mode.
3. The STP0 pin is used to verify that the PTM capture signal is correct.

Software Description

MCU Initialisation Table

<table>
<thead>
<tr>
<th>Step</th>
<th>Content</th>
<th>Register</th>
<th>Setup Bits</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup HIRC frequency</td>
<td>SCC</td>
<td>CKS[2:0]=000</td>
<td>System clock select; fH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FHS=0</td>
<td>High frequency clock select: HIRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HIRC[1:0]=10</td>
<td>FHS=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HIRCC</td>
<td>HIRC frequency select: 8MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HIRCE0</td>
<td>HIRC oscillator; Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HIRCE0=1</td>
</tr>
<tr>
<td>2</td>
<td>Setup pins</td>
<td>PAS1</td>
<td>PAS1[7:6]=01</td>
<td>PA7 pin function setup: STP0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCS0</td>
<td>PCS0[5:4]=00</td>
<td>PC2 pin function setup: PTCK0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAC</td>
<td>PAC7=0</td>
<td>PA7 pin type setup: output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCC</td>
<td>PCC2=1</td>
<td>PC2 pin type setup: input</td>
</tr>
</tbody>
</table>
The PTM operates in the capture input mode with the following settings.

<table>
<thead>
<tr>
<th>Step</th>
<th>Content</th>
<th>Register</th>
<th>Setup Bits</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup PTM Clock Source</td>
<td>PTM0C0</td>
<td>PT0CK[2:0]</td>
<td>PTM Clock Source select: fH/64 PT0CK[2:0]=&quot;011&quot;</td>
</tr>
<tr>
<td>2</td>
<td>Setup PTM Operating Mode</td>
<td>PTM0C1</td>
<td>PT0M[1:0]</td>
<td>PTM Mode Select: Capture Input Mode PT0M[1:0]=&quot;01&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Setup PTM External Pin Function</td>
<td>PTM0C1</td>
<td>PT0IO[1:0]</td>
<td>PTM Trigger Signal: Dual Edge Input Capture PT0IO[1:0]=&quot;10&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Setup Input Capture Input Source</td>
<td>PTM0C1</td>
<td>PT0CAPTS</td>
<td>Capture Input Signal Select: PTCK0 PT0CAPTS=&quot;1&quot;</td>
</tr>
<tr>
<td>5</td>
<td>Setup PTM Interrupts</td>
<td>MF0</td>
<td>PTM0AE</td>
<td>Start PTM Comparator A Interrupt PTM0AE=&quot;1&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTC0</td>
<td>MF0E</td>
<td>Start PTM Multi-function Interrupt 0 MF0E=&quot;1&quot;</td>
</tr>
<tr>
<td>6</td>
<td>Setup PTM Interrupt Flags</td>
<td>INTC0</td>
<td>MF0F</td>
<td>Clear PTM Multi-function Interrupt 0 Flag MF0F=&quot;0&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MF0</td>
<td>PTM0AF</td>
<td>Clear PTM Comparator A Interrupt Flag PTM0AF=&quot;0&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MF0</td>
<td>PTM0PF</td>
<td>Clear PTM Comparator P Interrupt Flag PTM0PF=&quot;0&quot;</td>
</tr>
<tr>
<td>7</td>
<td>Setup Global Interrupt</td>
<td>INTC0</td>
<td>EMI</td>
<td>Start PTM Global Interrupt EMI=&quot;1&quot;</td>
</tr>
</tbody>
</table>

STM Compare Match Output Mode Setup:

<table>
<thead>
<tr>
<th>Step</th>
<th>Content</th>
<th>Register</th>
<th>Setup Bits</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup STM Clock Source</td>
<td>STM0C0</td>
<td>ST0CK[2:0]</td>
<td>STM Clock Source Select: fH/64 ST0CK[2:0]=&quot;011&quot;</td>
</tr>
<tr>
<td>2</td>
<td>Setup STM Operating Mode</td>
<td>STM0C1</td>
<td>ST0M[1:0]</td>
<td>STM Mode Select: Compare Match Output Mode ST0M[1:0]=&quot;00&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Setup STM External Pin Function</td>
<td>STM0C1</td>
<td>ST0IO[1:0]</td>
<td>STM Output Signal: Output Invert ST0IO[1:0]=&quot;11&quot; (Output Invert)</td>
</tr>
<tr>
<td>4</td>
<td>Setup Match Output Source Start Value</td>
<td>STM0C1</td>
<td>ST0OC</td>
<td>Match Output Signal Select: Initially High ST0OC=&quot;1&quot;(High)</td>
</tr>
<tr>
<td>5</td>
<td>Setup STM Counter Clear Condition</td>
<td>STM0C1</td>
<td>ST0CCCLR</td>
<td>Comparator Match Select: Comparator A Match ST0CCCLR=&quot;1&quot;</td>
</tr>
<tr>
<td>6</td>
<td>Setup Comparator A Match Value</td>
<td>STM0AH</td>
<td>Bit[1:0]</td>
<td>Setup Comparator Match Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STM0AL</td>
<td>Bit[7:0]</td>
<td>Setup Comparator Match Value</td>
</tr>
</tbody>
</table>
### Self-defined register name description:

<table>
<thead>
<tr>
<th>Item</th>
<th>Register</th>
<th>Bit Group</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>data0</td>
<td>2</td>
<td>Enter the capture signal width value</td>
</tr>
<tr>
<td>2</td>
<td>data1</td>
<td>2</td>
<td>Enter the capture signal width value</td>
</tr>
<tr>
<td>3</td>
<td>sum0</td>
<td>2</td>
<td>Enter the capture signal positive/negative pulse width value</td>
</tr>
<tr>
<td>4</td>
<td>count[20]</td>
<td>40</td>
<td>Store 20 positive pulse width values</td>
</tr>
<tr>
<td>5</td>
<td>count[20]</td>
<td>40</td>
<td>Store 20 negative pulse width values</td>
</tr>
<tr>
<td>6</td>
<td>fg</td>
<td>1 bit</td>
<td>First trigger flag</td>
</tr>
<tr>
<td>7</td>
<td>i</td>
<td>1 Variable</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>j</td>
<td>1 Variable</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>p</td>
<td>1 Variable</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>high</td>
<td>1 Variable</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>low</td>
<td>1 Variable</td>
<td></td>
</tr>
</tbody>
</table>

### Points to Note:

It is important to note the following during the design process:

1. Prevent the count value from exceeding a value of 1023, otherwise an additional register will be required to store the data.

2. Prevent the input signal from being less than 1 clk or less than 20 instruction cycle times, otherwise the captured data will be incorrect.

Therefore special attention must be paid when selecting the PTM clock source. The following formula suggests how to select the PTM clock source:

According to the frequency of the input capture pin, the formula is as follows

\[
\text{Count Value} = \frac{\text{PTM Clock Source}}{\text{PTM0CK Input Frequency} \times \frac{1}{\text{Duty(\%)}}} 
\]

EX1: Suppose the desired count value is 125, the PTM0CK input frequency is 500Hz, the duty cycle is 50% and the system frequency is 8MHz. What should the PTM clock source be?

\[
\text{PTM Clock Source} = 125 \times 500Hz \times \frac{1}{50\%} = 125000Hz 
\]

\[
\text{PTM Clock Source} = \frac{f_{sys}}{X} \rightarrow \frac{8M}{X} = 125000 \rightarrow X = 64 
\]

ANS: PTM Clock Source should select \( f_{sys}/64 \).

EX2: PTM0CK Input Frequency is 500Hz, the duty cycle is 50%, the system frequency is 8MHz. What PTM Clock Source can be selected?

\[
1023 = \frac{\text{PTM Clock Source}}{500 \times \frac{1}{50\%}} \rightarrow \text{PTM Clock Source} = 1023000 
\]

ANS: PTM Clock Source can select a frequency less than 1.023MHz
The slower the input capture pin frequency then the slower the PTM Clock Source Input frequency, otherwise the count value will be greater than 1023.

**PTM Bi-directional Input Capture Mode Timing Diagram**

**Method 1:** After the PTM is started the counter will start counting, use CCRA to calculate the positive/negative pulse width

1. When the PT0ON bit changes from 0 to 1, the PTM 0 counter starts counting.
2. When the PTCK0 pin has a valid edge transition, the current value of the counter is stored in the CCRA register and a PTM0 Comparator A interrupt will be generated. Regardless of what edge transition occurs on the PTCK0 pin, the counter will continue to work until the PT0ON bit, experiences a falling edge.
3. The counter is reset to zero when a CCRP compare match occurs. In this way the CCRP value controls the maximum value of the counter. When a CCRP Comparator P compare match occurs, a PTM0 Comparator P interrupt will be generated. Longer pulse widths can be measured by counting the CCRP overflow interrupt signals.

**Graphical Data:**

<table>
<thead>
<tr>
<th>PT0ON</th>
<th>CCRA</th>
<th>CCRP</th>
<th>PTM0AF</th>
<th>PTM0PF</th>
<th>Pulse Width</th>
<th>Read Pulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0x0000</td>
<td>0x0000</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>Data0</td>
<td>0x0000</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>Data1</td>
<td>0x0000</td>
<td>1</td>
<td>0</td>
<td>Data1−Data0</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>Data1</td>
<td>0x0000</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>Data0</td>
<td>0x0000</td>
<td>1</td>
<td>0</td>
<td>Data1 + (0x400H−Data0)</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>Data1</td>
<td>0x0000</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Method 2: Using PTM ON/OFF to read positive/negative pulse width values

1. When the PT0ON bit changes from 0 to 1, the PTM 0 counter starts counting.

2. When the PTCK0 pin has a valid edge transition, the current value of the counter is stored into the CCRA register and a PTM0 Comparator A interrupt is generated. The counter will be restarted regardless of edge transitions on the PTCK0 pin. The counter starts counting from 0. As a result, each captured value is the pulse width value.

3. ** Using this method, it is important to note that when a valid edge transition appears on PTM0 to restart counting, 16 system instruction cycles are required. Therefore the value in CCRA must be compensated for by 16 system instruction cycles to obtain an accurate pulse width value.

Graphical Data:

<table>
<thead>
<tr>
<th></th>
<th>PT0ON</th>
<th>CCRA</th>
<th>CCRP</th>
<th>PTM0AF</th>
<th>PTM0PF</th>
<th>Pulse Width</th>
<th>Read Pulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0x0000</td>
<td>0x0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Negative pulse</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>Data0</td>
<td>0x0000</td>
<td>1</td>
<td>0</td>
<td>Data0 + Counter Error</td>
<td>Positive pulse</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>Data1</td>
<td>0x0000</td>
<td>1</td>
<td>0</td>
<td>Data1 + Counter Error</td>
<td>Negative pulse</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>Data1</td>
<td>0x0000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Negative pulse</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>Data0</td>
<td>0x0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Positive pulse</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0x0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Negative pulse</td>
</tr>
</tbody>
</table>
Flowchart

Method 1

Overall flow chart

Step 1: Initialisation: Clear the RAM data, set the HIRC frequency, pin initialisation, PTM settings, STM settings, then go to step 2.

Step 2: The PTM is enabled, the counter starts counting and step 3 is executed.

Step 3: Determine whether STM0AF is equal to 1 and whether p is greater than or equal to 1,
   If Yes: go to step 4
   If No: Go to step 3

Step 4: Execute the STM LOOP: Input the positive pulse width and negative pulse width captured by the input to the STM as output matching data, so that the STM outputs the same signal on the PTCK0 pin. Step 3 is executed.

PTM ISR: When the PT0CK pin has a high edge or low edge signal, it will capture the count value into the CCRA register and calculate the pulse width value.

---

Flowchart:

START
   INIT RAM
   INIT HIRC
   INIT IO
   INIT PTM
   INIT STM
   PTM ON
   STM0AF = 1 & p ≥ 1 ?
      NO
      PTM ISR
         Calculate pulse width value
         RETURN
      YES
      STM LOOP
PTM Interrupt Flowchart

Step 1: Clear the PTM0AF flag: as the PTM0 Comparator A Match interrupt bit is contained within the multi-function interrupt it will not be automatically reset. It must therefore be reset by the application program. Execute step 2.

Step 2: Check if fg is equal to 0, check if it’s the first interrupt
   If Yes: Execute step 3
   If No: Execute step 5

Step 3: Store the PTM0 Comparator A value into the data0 register. Execute step 4.

Step 4: Set fg as 1. Execute step 18.

Step 5: Store the PTM0 Comparator A value into the data1 register. Execute step 6.

Step 6: Check if data1 is greater than data0.
   If Yes: Execute step 7
   If No: Execute step 8

Step 7: data1 - data0, then store into the sum0 register. Execute step 9.

Step 8: data1 + 400H-data0 value, then store into the sum0 register. Execute step 9.
   Note: In normal operation, data0 will be less than data1. Because data0 has been captured by the previous trigger, data1 will be greater than data0. However when data0 is greater than data1, this means that the PTM0 counter has overflowed and the counter will restart counting from 0. The maximum value of the PTM0 register is the same as the CCRP register. When the CCRP register has a value of 000H, the PTM0 will have a maximum value of 400H.

Step 9: data1 is stored into data0. This step stores the previous interrupt count value into data0. The new interrupt count value will be stored into data1. Execute step 10.

Step 10: Check if PTCK0 is equal to 1
   If Yes: this indicates a rising edge trigger, the result is the negative pulse width.
   Execute step 11
   If No: this indicates a falling edge trigger, the result is the positive edge pulse width.
   Execute step 13.

Step 11: Store the sum0 value into countl matrix variable. Execute step 12.

Step 12: Variable low +1. Then check if its greater than 19. If it is then the low variable should be cleared to zero. Execute step 15.

Step 13: Store the sum0 value into counth matrix variable. Execute step 14.

Step 14: Variable high +1. Check if it is greater than 19, if it is then the High variable should be cleared to zero. Execute step 15.

Step 15: Check if variable High and variable Low is equal to 2 and if p is equal to 0
   If Yes: Execute step 16
   If No: Execute step 18

Step 16: Set STM0AF to be 1. Execute step 17.

Step 17: Set p to be 1. Execute step 18.

Step 18: Return
Using the HT66F45x0 PTM Bidirectional Capture Input Application Note

PTM ISR

PTM0AF = 0

fg = 0 ?

NO

YES

Store PTM comparator A value into variable data0

fg = 1

Store PTM comparator A value into variable data1

data1 > data0 ?

NO

YES

sum0 = data1 - data0

sum0 = data1 + (0x0400H - data0)

data0 = data1

PTCK0 = 1 ?

NO

YES

counth[low] = sum0

low ++

low > 19 ?

NO

YES

low = 0

high = 2

& low = 2

& p = 0 ?

NO

YES

STM0AF = 1

p = 1

high > 19 ?

NO

YES

high = 0

high = 0

high = 0

high = 0

counth[high] = sum0

high ++
STM Output Flowchart

Step 1: Clear the STM0AF flag: as the STM0 Comparator A compare match interrupt bit is contained within the multi-function interrupt, the interrupt flag will not be automatically reset. Therefore the interrupt bit has been reset by the application program. Execute step 2.

Step 2: Check if it is equal to 0
   If Yes: Execute step 3
   If No: Execute step 5

Step 3: Store data0 into the CCRA register. Execute step 4.

Step 4: i+1. Execute step 10

Step 5: Store data1 into the CCRA register. Execute step 6.


Step 8: Check if j is greater than 19
   If Yes: Execute step 9
   If No: Execute step 10

Step 9: Set j to 0. Execute step 10

Step 10: Check if p is equal to 1
   If Yes: Execute step 11
   If No: Execute step 13


Step 13: Return
Method 2

Overall Flowchart

Step 1: Initialisation: Clear the RAM data, setup the HIRC frequency, initialise the pins, setup the PTM, setup the STM and execute step 2.

Step 2: Enable the PTM: The counter starts counting. Execute step 3.

Step 3: Check if STM0AF is equal to 1 and that p is greater than 1
  If Yes: Execute step 4
  If No: Execute step 3

Step 4: Execute the STM LOOP: the positive captured pulse width and the captured negative pulse width are provided to the STM for output matching data, the STM produces the same signal as the PTCK0 pin. Execute step 3.

PTM ISR: After the PT0CK pin receives a rising or falling edge, the PTM will be restarted and the CCRA register value will be stored into data0 or data1.
PTM Interrupt Flowchart

Step 1: Clear the PTM0AF flag: as the PTM0 Comparator A interrupt bit is stored within the multi-function interrupt, the interrupt flag will not be automatically reset. Therefore it has to be reset by the application program. Execute step 2.

Step 2: Reset the PTM: Clear PTM0ON to 0 and then set to 1. Execute step 3.

Step 3: Check if the variable fg is equal to 0. Check if it is the first trigger.
   If Yes: Execute step 4
   If No: Execute step 5


Step 5: Check if PTCK0 is equal to 1. Check if it is a rising edge trigger.
   If Yes: Execute step 6
   If No: Execute step 8

Step 6: Save the PTM0 Comparator A value into the data1 register. Execute step 7.

Step 7: Variable Low +1. Then determine if it is greater than 19, if it is then clear the Low variable to 0. Execute step 10.

Step 8: Save the PTM0 Comparator A value into the data0 register. Execute step 9.

Step 9: Variable High +1. Then determine if it is greater than 19, if it is then clear the High variable to 0. Execute step 10.

Step 10: Check if the variable High and variable Low are equal to 2 and if variable p is equal to 0.
   If Yes: Execute step 11
   If No: Execute step 13
   Step 11: Set STM0AF to 1. Execute step 12.


Step 13: Return
STM Output Flowchart

Step1: Clear the STM0AF flag: Because the STM0 Comparator A matches the interrupt bit in the multi-function interrupt, the interrupt flag will not be automatically reset. It must be cleared by the application program. Then step 2 is executed.

Step2: Check if i is equal to 0
- If yes: Execute step 3
- If no: Execute step 5

Step3: Store data0 into the CCRA register. Execute Step4

Step4: i + 1. Execute Step10

Step5: Store data1 into the CCRA register. Execute Step6

Step6: i - 1. Execute Step7

Step7: j + 1. Execute Step8

Step8: Check if j is greater than 19
- If yes: Execute step 9
- If no: Execute step 10

Step9: Setup j is equal to 0. Execute step 10.

Step10: Check if p is equal to 1
- If yes: Execute step 11
- If no: Execute step 13


Step12: Setup p to be equal to 2. Execute step 13.

Step13: Return
Test Data

When the PTCK0 Input Frequency is 500Hz, the period is 2ms. When the duty cycle is increased from 20% to 80% in steps of 10%, what is the positive/negative pulse width of these 7 data? What is the count value captured by the PTM? The following table will offer a solution and show the calculation formula.

<table>
<thead>
<tr>
<th>Duty (%)</th>
<th>Positive Pulse Width (µs)</th>
<th>Negative Pulse Width (µs)</th>
<th>Positive Width Count Value</th>
<th>Negative Width Count Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>400</td>
<td>1600</td>
<td>32H</td>
<td>C8H</td>
</tr>
<tr>
<td>30</td>
<td>600</td>
<td>1400</td>
<td>4BH</td>
<td>AFH</td>
</tr>
<tr>
<td>40</td>
<td>800</td>
<td>1200</td>
<td>64H</td>
<td>96H</td>
</tr>
<tr>
<td>50</td>
<td>1000</td>
<td>1000</td>
<td>7DH</td>
<td>7DH</td>
</tr>
<tr>
<td>60</td>
<td>1200</td>
<td>800</td>
<td>96H</td>
<td>64H</td>
</tr>
<tr>
<td>70</td>
<td>1400</td>
<td>600</td>
<td>AFH</td>
<td>4BH</td>
</tr>
<tr>
<td>80</td>
<td>1600</td>
<td>400</td>
<td>C8H</td>
<td>32H</td>
</tr>
</tbody>
</table>

Calculation

\[
\text{Period} \times \text{Duty} \times (1-\text{Duty})
\]

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Period × Duty</th>
<th>Period × (1-Duty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTM Clock Source</td>
<td>PTM Clock Source</td>
<td></td>
</tr>
<tr>
<td>PTCK0 Input Frequency ( \times ) ( \frac{1}{\text{Duty}} )</td>
<td>PTCK0 Input Frequency ( \times ) ( \frac{1}{1-\text{Duty}} )</td>
<td></td>
</tr>
</tbody>
</table>

Measurement 1: PTCK0 Input Frequency=500Hz; Duty=20%

- **Signal Waveform**: CH1: PTCK0 output signal, VIN=5V, Positive pulse width = 403µs, Negative pulse width = 1597µs
- **Signal Data**: CH2: STP0 output signal, VOUT=5V, Positive pulse width = 400µs, Negative pulse width = 1600µs
- **Register read value**: counth=0x0032/0x0031, countl=0x00C8/0x00C7

Measurement 2: PTCK0 Input Frequency=500Hz; Duty=30%

- **Signal Waveform**: CH1: PTCK0 output signal, VIN=5V, Positive pulse width = 602µs, Negative pulse width = 1398µs
- **Signal Data**: CH2: STP0 output signal, VOUT=5V, Positive pulse width = 600µs, Negative pulse width = 1401µs
- **Register read value**: counth=0x004B/0x004A, countl=0x00AF/0x00AE
### Measurement 3: PTCK0 Input Frequency=500Hz; Duty=40%

<table>
<thead>
<tr>
<th>Signal Waveform</th>
<th>Signal Data</th>
</tr>
</thead>
</table>
| ![Signal Waveform](signal_waveform1.png) | **CH1:** PTCK0 output signal  
  $V_{in}=5V$  
  Positive pulse width=802µs  
  Negative pulse width=1198µs  
  **CH2:** STP0 output signal  
  $V_{out}=5V$  
  Positive pulse width=800µs  
  Negative pulse width=1200µs  
  Register read value:  
  $\text{counth}=0x0064/0x0063$  
  $\text{countl}=0x0095/0x0096$ |

### Measurement 4: PTCK0 Input Frequency=500Hz; Duty=50%

<table>
<thead>
<tr>
<th>Signal Waveform</th>
<th>Signal Data</th>
</tr>
</thead>
</table>
| ![Signal Waveform](signal_waveform2.png) | **CH1:** PTCK0 output signal  
  $V_{in}=5V$  
  Positive pulse width=1001µs  
  Negative pulse width=998µs  
  **CH2:** STP0 output signal  
  $V_{out}=5V$  
  Positive pulse width=1000µs  
  Negative pulse width=1001µs  
  Register read value:  
  $\text{counth}=0x007C/0x007D$  
  $\text{countl}=0x007C/0x007D$ |

### Measurement 5: PTCK0 Input Frequency=500Hz; Duty=60%

<table>
<thead>
<tr>
<th>Signal Waveform</th>
<th>Signal Data</th>
</tr>
</thead>
</table>
| ![Signal Waveform](signal_waveform3.png) | **CH1:** PTCK0 output signal  
  $V_{in}=5V$  
  Positive pulse width=1201µs  
  Negative pulse width=799µs  
  **CH2:** STP0 output signal  
  $V_{out}=5V$  
  Positive pulse width=1200µs  
  Negative pulse width=800µs  
  Register read value:  
  $\text{counth}=0x0096/0x0095$  
  $\text{countl}=0x0063/0x0064$ |
Measurement 6: PTCK0 Input Frequency=500Hz; Duty=70%

<table>
<thead>
<tr>
<th>Signal Waveform</th>
<th>Signal Data</th>
</tr>
</thead>
</table>
| CH1: PTCK0 output signal  
V_in=5V  
Positive pulse width=1400µs  
Negative pulse width=600µs  
CH2: STP0 output signal  
V_out=5V  
Positive pulse width=1400µs  
Negative pulse width=600µs  
Register read value:  
counth=0x00AF/0x00AE  
countl=0x004B/0x004A

Measurement 7: PTCK0 Input Frequency=500Hz; Duty=80%

<table>
<thead>
<tr>
<th>Signal Waveform</th>
<th>Signal Data</th>
</tr>
</thead>
</table>
| CH1: PTCK0 output signal  
V_in=5V  
Positive pulse width=1600µs  
Negative pulse width=400µs  
CH2: STP0 output signal  
V_out=5V  
Positive pulse width=1600µs  
Negative pulse width=400µs  
Register read value:  
counth=0x00C7/0x00C6  
countl=0x0032/0x0033

Software Attachment

- HT66F4530_PTM_Capture_AP_EX1_C.zip
- HT66F4530_PTM_Capture_AP_EX2_C.zip

Conclusion

This application note has introduced the PTM bidirectional input capture mode in the HT66F4530 and explained its working principles and software control flow. It used the STM output to compare the matching mode with a general data storage method to verify the data accuracy. However it should be noted that some restrictions remain within the sample program. Users can select an appropriate counting clock source according to the application input signal size. The second method requires special attention with regard to counting error compensation, otherwise there will be a 16 instruction cycle time error.
Reference Material

Consult the HT66F45x0 Data Sheet.
For more information consult the Holtek website at www.holtek.com.

Versions and Modification Information

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Issue Release and Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.5.8</td>
<td>陈淑娟(Chen, Shu-Juan)</td>
<td>First version</td>
</tr>
</tbody>
</table>

Disclaimer

All information, trademarks, logos, graphics, videos, audio clips, links and other items appearing on this website ('Information') are for reference only and is subject to change at any time without prior notice and at the discretion of Holtek Semiconductor Inc. (hereinafter 'Holtek', 'the company', 'us', 'we' or 'our'). Whilst Holtek endeavors to ensure the accuracy of the Information on this website, no express or implied warranty is given by Holtek to the accuracy of the Information. Holtek shall bear no responsibility for any incorrectness or leakage.
Holtek shall not be liable for any damages (including but not limited to computer virus, system problems or data loss) whatsoever arising in using or in connection with the use of this website by any party. There may be links in this area, which allow you to visit the websites of other companies. These websites are not controlled by Holtek. Holtek will bear no responsibility and no guarantee to whatsoever Information displayed at such sites. Hyperlinks to other websites are at your own risk.

Limitation of Liability

In any case, the Company has no need to take responsibility for any loss or damage caused when anyone visits the website directly or indirectly and uses the contents, information or service on the website.

Governing Law

This disclaimer is subjected to the laws of the Republic of China and under the jurisdiction of the Court of the Republic of China.

Update of Disclaimer

Holtek reserves the right to update the Disclaimer at any time with or without prior notice, all changes are effective immediately upon posting to the website.