



# TouchCore Family

Application Note #041 (AN041-V1.1)

AN

## [TouchCore 3.x] Current Reduce Solution

V1.1

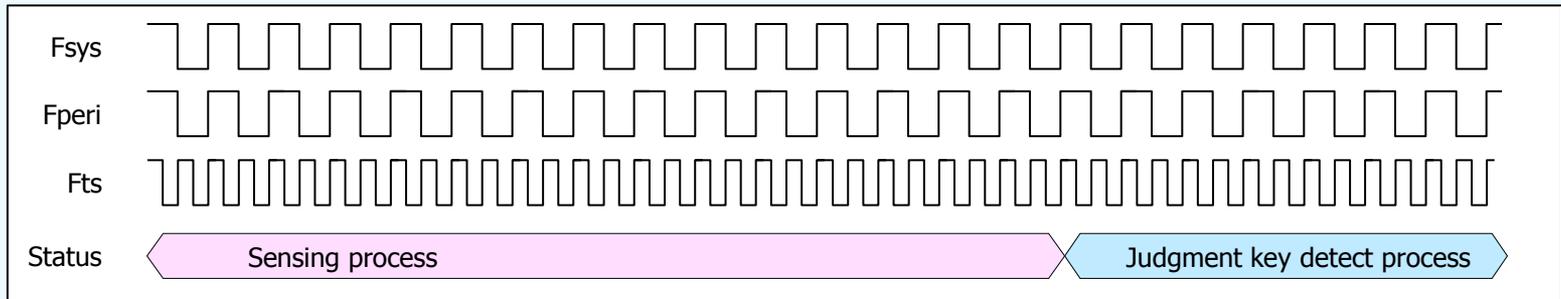
October 2010

- ◆ CORERIVER Semiconductor reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time.
- ◆ CORERIVER shall give customers at least a three month advance notice of intended discontinuation of a product or a service through its homepage.
- ◆ Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
- ◆ The CORERIVER products listed in this document are intended for usage in general electronics applications. These CORERIVER products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury..

# 1. Solution for current reduction

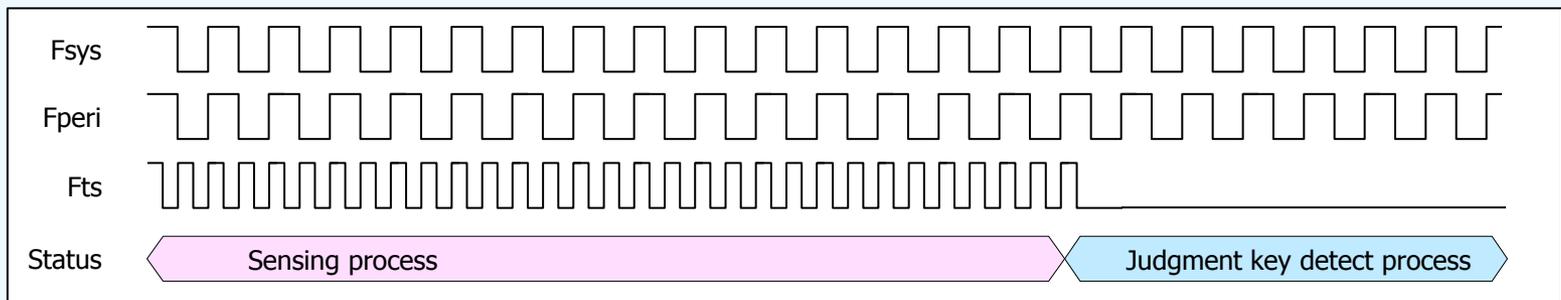
## ◆ Normal State : No Method for current reduction

- ✓ Current (on demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : Apprx. **13.40mA**



## ◆ Method #1 : Touch sampling clock(Fts) OFF just during do not sensing

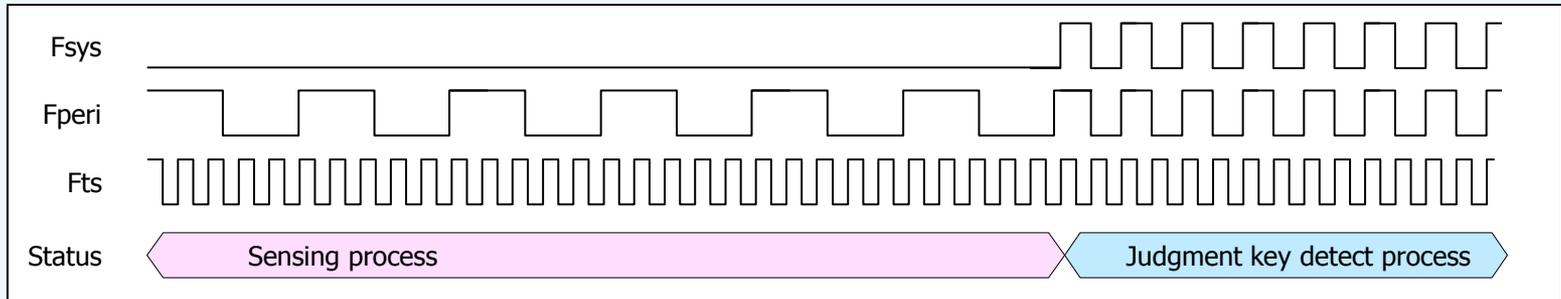
- ✓ Current (at demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : Apprx. **13.39mA**
- ✓ The current reduction is very low only with Method #1.  
But, if this method #1 is used with the other methods, the current reduction is more good.
- ✓ For use : tc30\_user.h → 8. Reduce the current → **#define TS\_CLOCK\_OFF TRUE**



# 1. Solution for reduce current

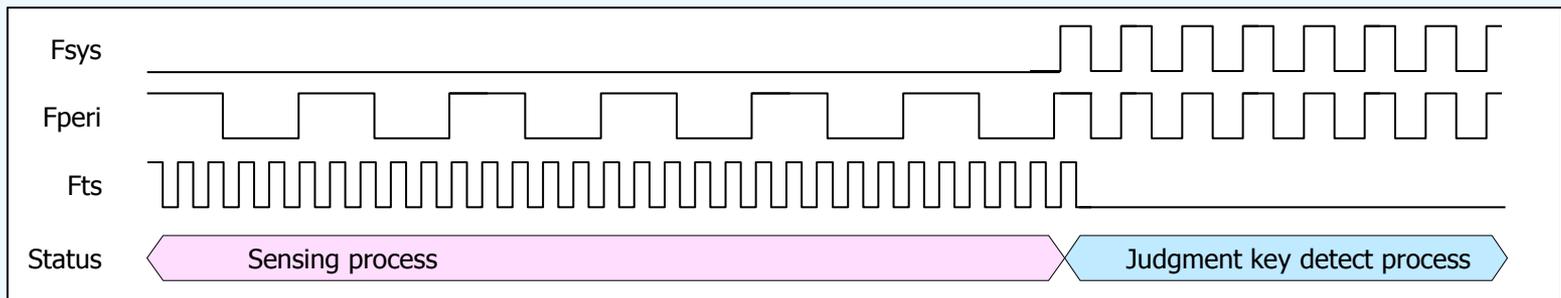
## ◆ Method #2 : Enter IDLE mode (before Fsys/Fperi change to 2MHz) while sensing

- ✓ Current (on demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : apprx. **9.03mA**
- ✓ For use : tc30\_user.h → 8. Reduce the current → **#define SYSTEM\_STATUS\_WHILE\_SENSING IDLE**



## ◆ Method #1 + #2 : Enter IDLE mode (before Fsys/Fperi change to 2MHz) while sensing with TS Clock off during sensing

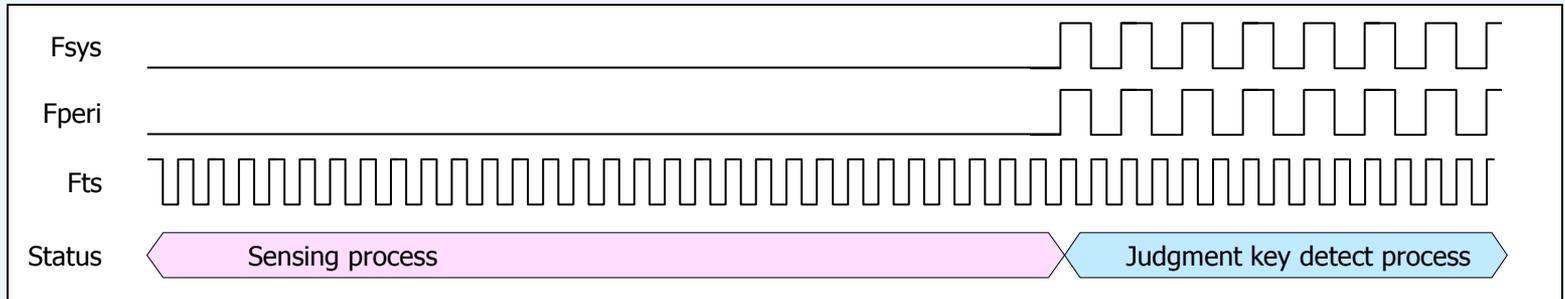
- ✓ Current (on demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : apprx. **9.05mA**
- ✓ For use : tc30\_user.h → 8. Reduce the current → **#define SYSTEM\_STATUS\_WHILE\_SENSING IDLE**  
tc30\_user.h → 8. Reduce the current → **#define TS\_CLOCK\_OFF TRUE**



# 1. Solution for reduce current

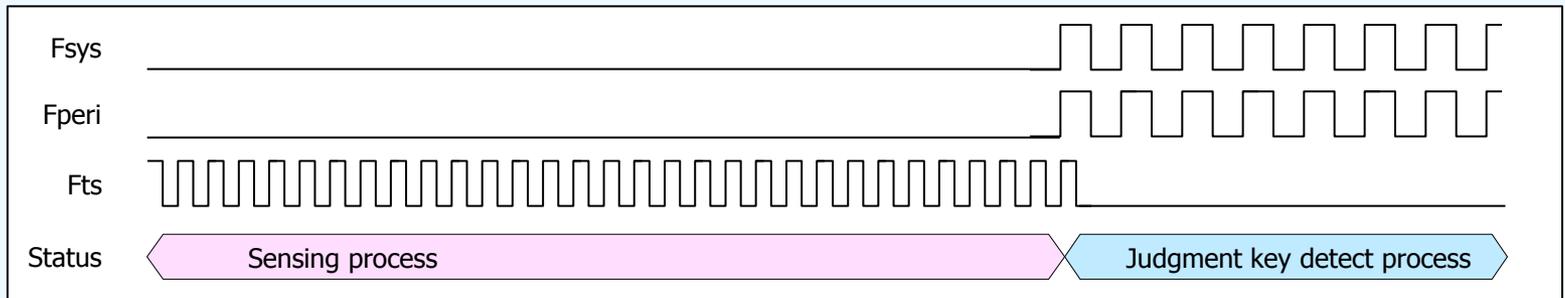
## ◆ Method #3 : Enter STOP mode while sensing

- ✓ Current (on demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : apprx. **9.30mA**
- ✓ For use : tc30\_user.h → 8. Reduce the current → **#define SYSTEM\_STATUS\_WHILE\_SENSING STOP**



## ◆ Method #1 + #3 : Enter STOP mode while sensing with TS Clock off during sensing

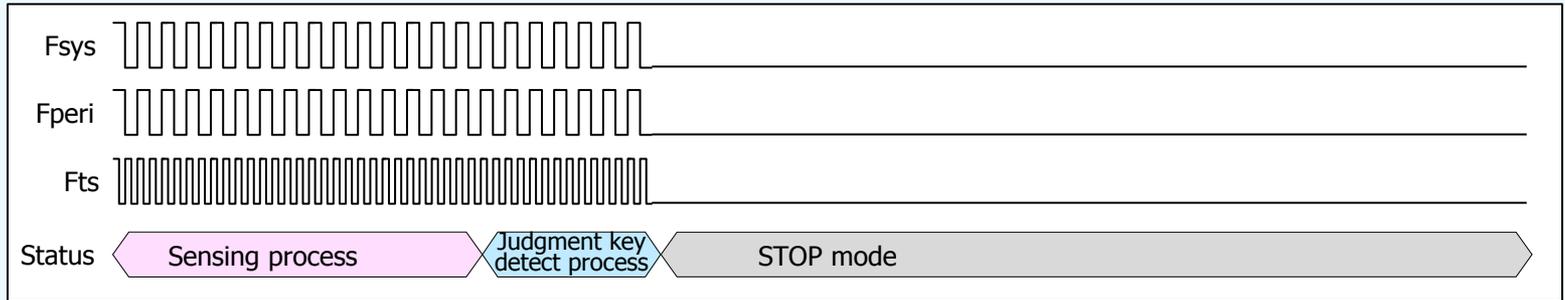
- ✓ Current (on demo B/D of CR (16key, 3.3V, Fsys 48MHz, Fts 96MHz)) : apprx. **7.28mA**
- ✓ For use : tc30\_user.h → 8. Reduce the current → **#define SYSTEM\_STATUS\_WHILE\_SENSING STOP**  
tc30\_user.h → 8. Reduce the current → **#define TS\_CLOCK\_OFF TRUE**



# 1. Solution for reduce current

## ◆ Method #4 : SNAPMODE

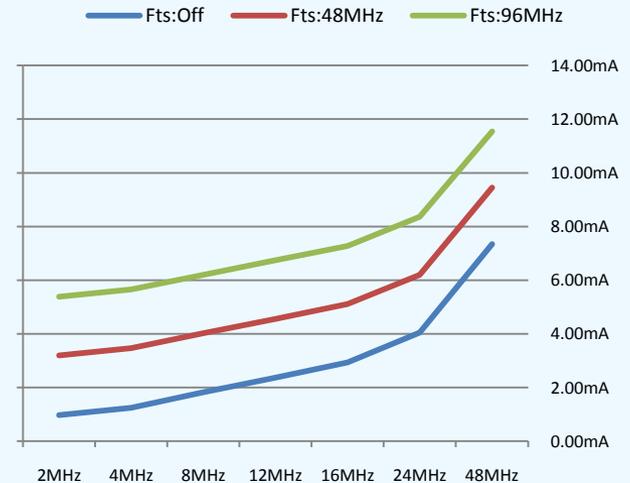
- ✓ Current : deferent through tune
- ✓ For use : tc30\_user.h → 8. Reduce the current → `#define SNAPMODE_ENABLE TRUE`



## ◆ Method #5 : Change System clock / Touch sensor clock

- ✓ Can reduce the current through change clock
- ✓ Refer to the below data for check reduce effect

Status Fsys \ Fts	Fts:Off	Fts:48MHz	Fts:96MHz	STOPMODE (with 32K RING)
2MHz	0.98mA	3.19mA	5.38mA	80uA
4MHz	1.25mA	3.47mA	5.65mA	80uA
8MHz	1.82mA	4.02mA	6.20mA	80uA
12MHz	2.37mA	4.56mA	6.74mA	80uA
16MHz	2.93mA	5.11mA	7.28mA	80uA
24MHz	4.05mA	6.20mA	8.36mA	80uA
48MHz	7.35mA	9.45mA	11.55mA	80uA



※ Internal LDO current : 40.5uA, 32K RING current 1.5uA

# 1. Solution for reduce current

## ◆ Method #6 : LVD\_DISABLE

- ✓ Current : ??
- ✓ For use : tc30\_user.h → 8. Reduce the current → `#define LVD_ENABLE FALSE`

## ◆ Method #7 : UNUSED PARI CLOCK DISABLE

- ✓ Current : ??
- ✓ For use :
- ✓ PCLKEN      &= ~UART\_CLKEN\_;
- ✓ PCLKEN      &= ~I2C0\_CLKEN\_;
- ✓ PCLKEN      &= ~TMR2\_CLKEN\_;

## ◆ Method #8 : Sensing essential Channel Only, during the snap mode

- ✓ Can reduce the current through Active run time make shorter

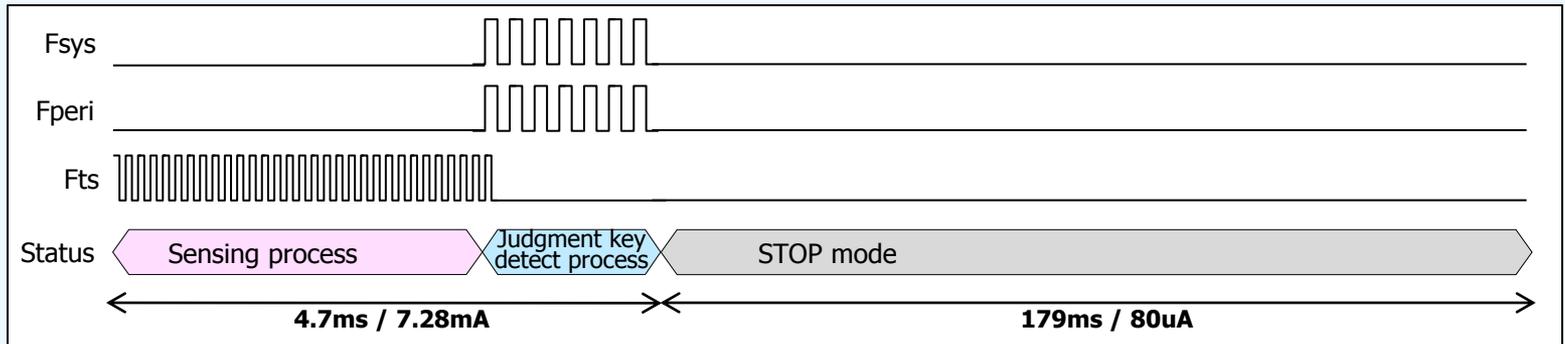
## ◆ Method #9 : Adjust TSPC, TSPD to reduce the Active run time

- ✓ Current : ??
- ✓ For use : tc30\_user.h → 8. Reduce the current →
- ✓ `#define TSPD_VAL_MODE_0           2000`
- ✓ `#define TSPC_VAL_MODE_0           0`
- ✓ `#define TSENCT_VAL_MODE_0       64`

## 2. Example

### ◆ Example (#1+#3+#4) : TS Clock OFF / STOP during sensing / SNAPMODE

- ✓ For use : tc30\_user.h → 8. Reduce the current → `#define SNAPMODE_ENABLE TRUE`  
tc30\_user.h → 8. Reduce the current → `#define SYSTEM_STATUS_WHILE_SENSING STOP`  
tc30\_user.h → 8. Reduce the current → `#define TS_CLOCK_OFF TRUE`



- ✓ Current =  $((4.7 \times 7280) + (179 \times 80)) / (4.7 + 179) = \text{apprx. } 264\mu\text{A}$

### ◆ Note : Can adjust current through STOP mode time and change system/touch clock

- ✓ Refer to the "Ref.WDT Time in 32K RING" for change STOP mode time in SNAPMODE
- ✓ Refer to the "(Change System clock / Touch sensor clock) content of Solution for reduce current" for change system/touch clock

## 3. Note

- ◆ **Note** : If you need to use IIC bus during the stop mode , then please follow the process below.
  - ✓ Enable the Start/Stop Interrupt ( for wakeup & run IIC ISR by the start condition)
  - ✓ Enter the STOP mode( watchdog enabled)
  - ✓ If it is woken by IIC interrupt & run the IIC ISR, then wait till the watchdog interrupt occurred.
  - ✓ Exit the STOP mode & restore the setting to run Active mode

Ex)

```
...
snap_status=1;
snap_counter=1;           // WDT interrupt signal wait flag
I2C1CFG |= 0x02;         // Enable Start/Stop Interrupt
WDIF = 0;      RWT = 1;
PCON = PCON | PD_;      // STOP mode Enter
_nop_();      _nop_();
snap_status=0;
I2C1CFG &= 0xFD;        // Disable Start/Stop Interrupt
while(snap_counter);   // WAIT snap_counter cleared in the WDT ISR
...
```

# Ref. WDT Time in 32k RING (For SNAP mode)

WDMOD=0			Interrupt Time-out		Reset Time-out	WDMOD=1			Interrupt Time-out		Reset Time-out (@22.88kHz)
WD2	WD1	WD0	(@22.88kHz)		(@22.88kHz)	WD2	WD1	WD0	(@22.88kHz)		
0	0	0	4096	<b>179.04ms</b>	$2^{12} + 512$ clocks	0	0	0	2097152	<b>1m 32s</b>	$2^{21} + 512$ clocks
0	0	1	8192	<b>358.07ms</b>	$2^{13} + 512$ clocks	0	0	1	4194304	<b>3m 3s</b>	$2^{22} + 512$ clocks
0	1	0	16384	<b>716.14ms</b>	$2^{14} + 512$ clocks	0	1	0	8388608	<b>6m 7s</b>	$2^{23} + 512$ clocks
0	1	1	32768	<b>1.43s</b>	$2^{15} + 512$ clocks	0	1	1	16777216	<b>12m 13s</b>	$2^{24} + 512$ clocks
1	0	0	65536	<b>2.86s</b>	$2^{16} + 512$ clocks	1	0	0	33554432	<b>24m 27s</b>	$2^{25} + 512$ clocks
1	0	1	131072	<b>5.73s</b>	$2^{17} + 512$ clocks	1	0	1	67108864	<b>48m 53s</b>	$2^{26} + 512$ clocks
1	1	0	262144	<b>11.46s</b>	$2^{18} + 512$ clocks	1	1	0	134217728	<b>97m 47s</b>	$2^{27} + 512$ clocks
1	1	1	524288	<b>22.92s</b>	$2^{19} + 512$ clocks	1	1	1	268435456	<b>195m 33s</b>	$2^{28} + 512$ clocks