GPL951XXUA CTS Driver Code User's Manual

V0.1 – Sep. 02, 2016

Important Notice

Generalplus Technology reserves the right to change this documentation without prior notice. Information provided by Generalplus Technology is believed to be accurate and reliable. However, Generalplus Technology makes no warranty for any errors which may appear in this document. Contact Generalplus Technology to obtain the latest version of device specifications before placing your order. No responsibility is assumed by Generalplus Technology for any infringement of patent or other rights of third parties which may result from its use. In addition, Generalplus products are not authorized for use as critical components in life support devices/ systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

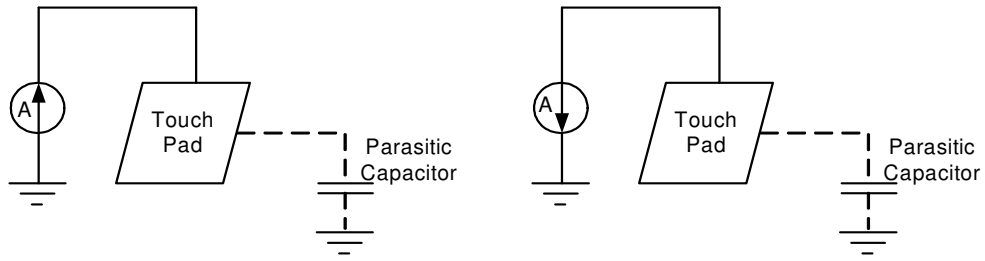
Revision History

Revision	Date	By	Remark
0.1	2016.09.02	Frank Kung	First Edition

1 Introduction

GPL951XXUA provides one Capacitor Touch Sensor Interface (CTS), this driver code can help users to develop CTS application easily. CTS Driver Code provides C-language and assembly functions for customer applications. This driver code also reserved some CTS driver control parameters to make user fine-tune CTS sensitive and error touch preventing.

A capacitor touch sensor circuit is showed as below:

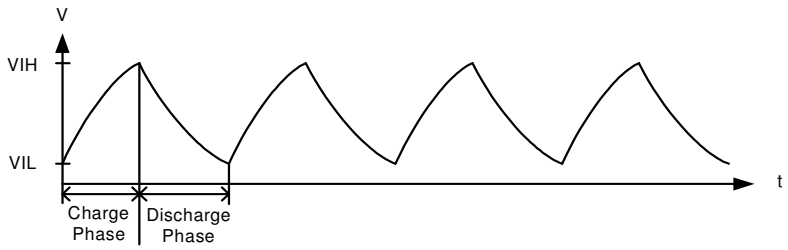


Charge Phase Discharge Phase

Where Charge and Discharge current is register control, Programmers can modify it in CTS_User.asm

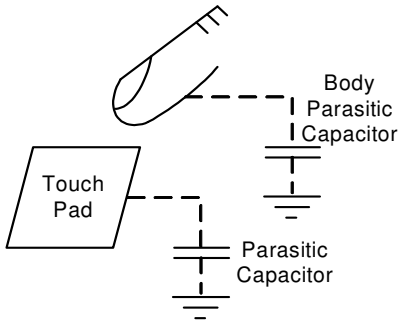
One CTS Scan Time is the period time length of one charge and discharge cycle.

The waveform is showed as below:

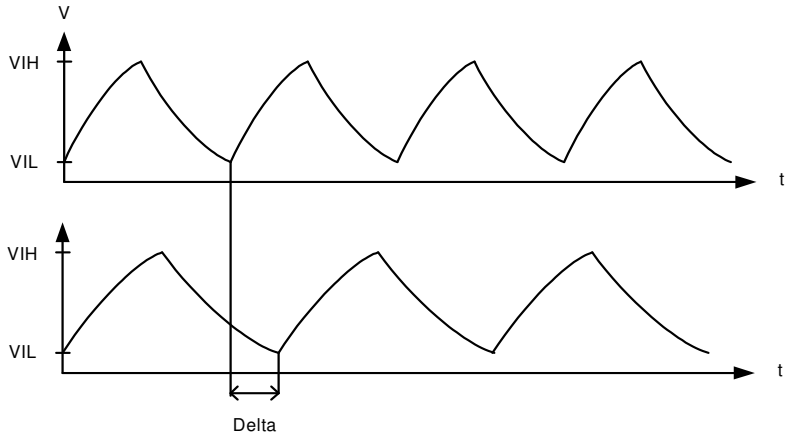


Where VIH and VIL are register control, Programmers can modify it in CTS_User.asm

When user is touching Touch Pad, the total capacitor value is circuit parasitic capacitor + user body parasitic capacitor, show as below



The waveform changing with and without touching is showed as below:



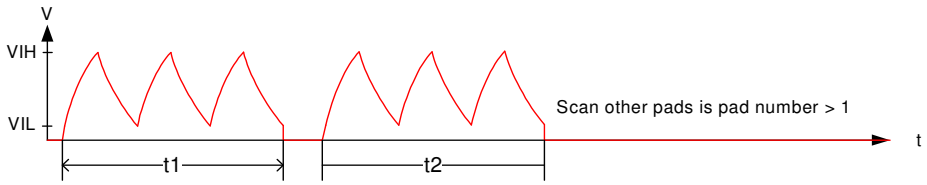
The CTS driver code will refer to Delta value to judge this touch pad is touched or not.

In driver code, user can modify variable R_ScanNumber, R_ThresholdOn, R_ThresholdOff and R_ScanCycleNum.

R_ScanNumber is the charge/discharge count for one cycle. R_ScanNumber variable number is equal to touch pad number to define each touch pad charge/discharge count.

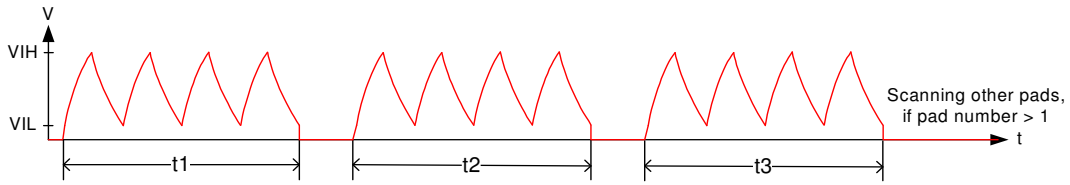
R_ ScanCycleNum is the cycle number of charge/discharge cycle count. The total charge/discharge time (R_CtsSum) is (t of R_ ScanNumber) * R_ ScanCycleNum.

Example1: R_ScanNumber = 3, R_ScanCycleNum = 2



This pad total charge/discharge time R_CtsSum is t1 + t2

Example2: R_ScanNumber = 4, R_ScanCycleNum = 3



This pad total charge/discharge time R_CtsSum is t1 + t2 + t3

R_TriggerLevel = R_CtsSum - R_CtsRefer

R_ThresholdOn is the threshold of on level, when R_TriggerLevel >= R_ThresholdOn means touch pad is touched.

R_ThresholdOff is the threshold of off level, when R_TriggerLevel < R_ThresholdOff means touch pad is not touched.

2 Function List for Assembly

CTS Driver Function List			
Function	**Parameter**	**Return**	**Description**
F_CTS_Version	None	R1	CTS Driver code version V1.0, R1=0x0100
F_CTS_Initial	None	None	CTS Driver variable and HW Initialize
F_CTS_ScanStart	None	None	CTS Scanning start
F_CTS_ISR	None	None	CTS Driver interrupt service
F_CtsForceCali	R1: Calibration Count	None	Force CTS Driver Calibration
F_CTS_ServiceLoop	None	None	CTS Service loop

CTS User's Function List			
Function	**Parameter**	**Return**	**Description**
F_CTS_User_Init	None	None	CTS HW initial
F_CTS_Para_User_Initial	None	None	CTS Default Parameters initial
F_CTSSetScanNum	R4: Pad Index	None	Set CTS Scan number
F_CTSChSel	R4: Pad Index	None	Set CTS Scan IO selection register
F_CTSStart	None	None	CTS Scanning Start
F_CTSAllChOff	None	None	Stop All CTS Scan and Clear Scan IO setting

3 Global Variable for Assembly

3.1 CTS Driver Global Variable

Name	Size(word)	Description
R_CtsSum	1 X Pad Number	Pad total scan time (Current value)
R_CtsReferL R_CtsRefer	2 X Pad Number	No Touch total scan time (Reference value).
R_SensorStatus	Bit Flag word number	Pad touch status (bit mask) 0: Touch release 1: Touched
R_CTS_Key	Bit Flag word number	Pad touch result (bit mask) 0: Touch release 1: Touched
R_TriggerLevel	Pad Number	CTS Sum - CTS Reference
R_ScanDoneFlag	1	CTS Scan completed Flag This flag is set by driver code, and cleared by user 0: CTS is scanning 1: CTS scan is completed
R_ScanNumber	Pad Number	Scan Number for each CTS IO
R_ThresholdOn	Pad Number	Touch On Threshold Level for each CTS IO
R_ThresholdOff	Pad Number	Touch Off Threshold Level for each CTS IO
R_ScanCycleNum	1	Scan Cycle Number

4 Global Table for Assembly

4.1 CTS Driver Global Table

Name	Size(word)	Description
T_TotalPadNum	2	CTS Pad number - 1
T_TraceUpStep	1	CTS Reference adaptive trace up step
T_TraceDownStep	1	CTS Reference adaptive trace down step
T_CalibrationCount	1	CTS driver calibration default count
T_ReferenceErrCnt	1	CTS reference level error de-bounce count
T_CtsKeyDebounceTime	1	CTS Touch de-bounce count
T_ScanLine	Pad Number	CTS Pad bit map
T_CTS_IOSel	Pad Number	CTS HW IO selection register setting
T_ScanNum	Pad Number	CTS Pad scan number setting value

5 Resource

5.1 CTS Driver Code

5.1.1 RAM Size

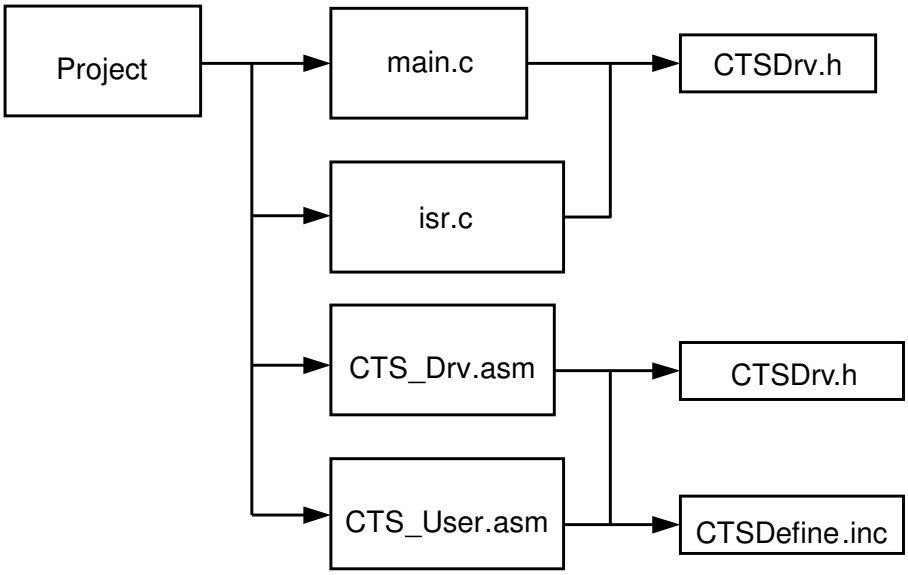
Overlap RAM (word)	Non-overlap RAM (word)
0	6 + 8xPad Number

5.1.2 ROM Size

Code (word)	DATA (word)
650 + 5 x Pad Number	0

6 Project Architecture for Assembly

6.1 Architecture



7 Application Interface for Assembly

7.1 CTS User Code

7.1.1 F_CTS_User_Init

Description	CTS HW initial
Parameters	None
Return Values	None
Destroy	R1, R2, R3
Remarks	This function will be call in function F_CTS_Initial
Example	F_CTS_User_Init: call F_CTS_Para_User_Initial r1 = 0xFFFF [P_IOF_Dir] = r1 [P_IOF_Attrib] = r1 r1 = C_IOF_CTS_En [P_IOF_Mux] = r1 r1 = C_CTS_SYS_DIV2 [P_CTS_DIV] = r1 r1 = C_CTS_LDO33_En [P_CTS_Ctrl3] = r1 r1 = C_CTS_ScanAllDoneIntEn [P_CTS_Ctrl2] = r1 r1 = C_CTS_Current_600u + C_CTS_Duration_450n + C_CTS_NF_200M + C_CTS_TriEn + D_CtsVrefSel_275_025 [P_CTS_Ctrl1] = r1 retf

7.1.2 F_CTS_Para_User_Initial

Description	CTS HW initial
Parameters	None
Return Values	None
Destroy	R1, R2, R3
Remarks	Users need to initialize Scan Number, On Threshold level, Off Threshold level and Scan Cycle Number. In example code, CTS parameter is get from default tables, user can modify the parameter data source by self.
Example	F_CTS_Para_User_Initial:

```
    r1 = seg16    T_ScanNum
    ds = r1
    r1 = D_SinglePadNum-1
L_Initial_Para:
    r2 = T_ScanNum
    r2 = r2 + r1
    r3 = DS:[r2]
    r2 = R_ScanNumber
    r2 = r2 + r1
    [r2] = r3

    r2 = T_OnThresholdLevel
    r2 = r2 + r1
    r3 = DS:[r2]
    r2 = R_ThresholdOn
    r2 = r2 + r1
    [r2] = r3

    r2 = T_OffThresholdLevel
    r2 = r2 + r1
    r3 = DS:[r2]
    r2 = R_ThresholdOff
    r2 = r2 + r1
    [r2] = r3

    r1 = r1 - 1
    jpl   L_Initial_Para

    r1 = seg16    T_ScanCycleCnt
    ds = r1
    r1 = T_ScanCycleCnt
    r2 = DS:[r1]
    [R_ScanCycleNum] = r2
    retf
```

7.1.3 F_CTSSetScanNum

Description	Set CTS Scan number
Parameters	R4: Pad Index
Return Values	None
Destroy	R2, R4
Remarks	CTS Driver code will call this function to set pad scan count
Example	F_CTSSetScanNum:

	r2 = R_ScanNumber r2 = r2 + r4 r2 = [r2] [P_CTS_CYCLE] = r2 retf

7.1.4 F_CTSChSel

Description	Set CTS Scan IO selection register
Parameters	R4: Pad Index
Return Values	None
Destroy	R2, R4
Remarks	Driver code will call this function to set CTS IO selection register
Example	F_CTSChSel: r2 = seg16 T_CTS_IOSel ds = r2 r2 = T_CTS_IOSel r2 = r2 + r4 r2 = ds:[r2] [P_CTS_CH] = r2 retf

7.1.5 F_CTSStart

Description	CTS Scanning Start
Parameters	None
Return Values	None
Destroy	R1
Remarks	CTS will call this function to clear CTS TM1 counter and Start CTS Scanning
Example	F_CTSStart: r1 = [P_CTS_Ctrl1] r1 = r1 | (C_CTS_En + C_CTS_Start) [P_CTS_Ctrl1] = r1 retf

7.1.6 F_CTSAllChOff

Description	Stop All CTS Scan and Clear Scan IO setting
Parameters	None
Return Values	None
Destroy	R1

Remarks	CTS Driver call will call this function to stop CTS Scan and Release CTS IO selection
Example	F_CTSAllChOff: r1 = [P_CTS_Ctrl1] r1 = r1 & ~(C_CTS_En) [P_CTS_Ctrl1] = r1 retf

8 Function List for C

CTS Driver Function List			
Function	**Parameter**	**Return**	**Description**
unsigned int F_CTS_Version(void);	None	CTS Driver Code Version	CTS Driver code version
void F_CTS_Initial(void);	None	None	CTS Driver variable and HW Initialize
void F_CTS_ScanStart(void);	None	None	CTS Scanning start
void F_CTS_ISR(void);	None	None	CTS Driver interrupt service
void F_CtsForceCali(unsigned int CalNum);	unsigned int CalNum Calibration Count	None	Force CTS Driver Calibration
void F_CTS_ServiceLoop(void);	None	None	CTS Service loop

9 Application Interface for C

9.1 CTS Driver Code

9.1.1 F_CTS_Version

Description	CTS Driver code version
Header File	CTSDrv.h
Syntax	unsigned int F_CTS_Version(void);
Parameters	None
Return Values	None
Remarks	Programmers can call this function to get CTS Driver library version

Example	#include CTSDrv.h
	int Version;
	int main(void) {
	Version = F_CTS_Version();
	}

9.1.2 F_CTS_Initial

Description	CTS Driver Variable and HW initialize
Header File	CTSDrv.h
Syntax	void F_CTS_Initial(void);
Parameters	None
Return Values	None
Remarks	Programmers must call this function to initial CTS driver before user CTS function

Example	#include CTSDrv.h
	int main(void) {
	F_CTS_Initial();
	}

9.1.3 F_CTS_ScanStart

Description	CTS Scanning start
Header File	CTSDrv.h
Syntax	void F_CTS_ScanStart(void);
Parameters	None
Return Values	None
Remarks	When this function is called CTS scan will start.

Example	#include CTSDrv.h
	int main(void) {
	F_CTS_Initial();
	F_CTS_ScanStart();
	}

9.1.4 F_CTS_ISR

Description	CTS Driver interrupt service
Header File	CTSDrv.h
Syntax	void F_CTS_ISR(void);
Parameters	None
Return Values	None
Remarks	Programmers must service this function in CTS TM0 ISR

Example	#include CTSDrv.h
	void IRQ6(void)
	{
	*P_CTS_Status = *P_CTS_Status;
	F_CTS_ISR();
	}

9.1.5 F_CtsForceCali

Description	Force CTS Driver Calibration
Header File	CTSDrv.h
Syntax	void F_CtsForceCali (unsigned int CalNum);
Parameters	unsigned int CaliCnt Calibration Count
Return Values	None
Remarks	Programmers can call this function to for CTS driver calibration

Example	#include CTSDrv.h
	int main(void) {
	F_CtsForceCali(20);
	}

9.1.6 F_CTS_ServiceLoop

Description	CTS Service loop
Header File	CTSDrv.h
Syntax	void F_CTS_ServiceLoop(void);
Parameters	None
Return Values	None
Remarks	Programmers must service this function in main loop to process CTS

Example	#include CTSDrv.h
	int main(void) {
	…
	While(1){
	F_CTS_ServiceLoop(void);
	…
	}

	}

10 Program Example

In maic.c

```
#include    ".\include\GPL951_Body.h"
#include    "CTSDrv.h"
void main(void)
{
      F_SYS_Initial();
       F_CTS_Initial();
       F_CTS_ScanStart();
       while(1)
       {
            *P_WatchDog_Clear = C_WDT_Clear;
            F_CTS_ServiceLoop();
            if(R_ScanDoneFlag != 0)
            {
                 if(R_CTS_Key != R_CtsPreKey)
                 {
                      i = R_CTS_Key;
                      R_CtsPreKey = R_CTS_Key;
                      *P_IOC_Data = R_CtsPreKey;
                 }
            }
       }
}
```

In isr.c

```
#include    CTSDrv.h
void IRQ6 (void)
{
       *P_CTS_Status = *P_CTS_Status;
       F_CTS_ISR();
}
```

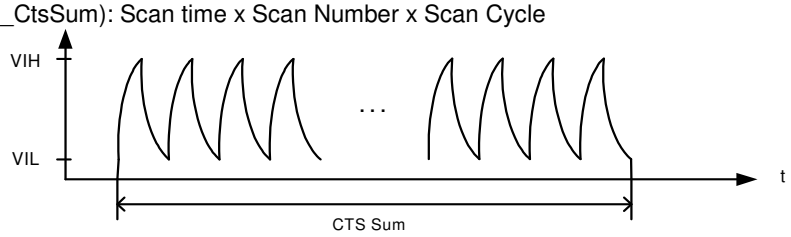
11 Driver Code Variable Definition

Below is CTS variables and constant definition:

Scan Number: This constant definition is used to control the CTS scan number for one cycle.

Scan Cycle: This constant definition is used to control the CTS scan cycle number.

CTS Sum (R_CtsSum): Scan time x Scan Number x Scan Cycle



CTS Reference (R_CtsRefer, R_CtsReferL): No touch CTS Sum average value.

Trigger Level (R_TriggerLevel): Touched and no touch charge CTS Sum delta

On Threshold Level: If CTS trigger level is great than this value, the correspond touch pad bit status will be set to1.

Off Threshold Level: If CTS trigger level is lower than this value, the correspond touch pad bit status will be set to 0.

Key Status (R_CTS_Key): It will show present Touch status. When touch pad is touched the correspond bit will be set to 1.

12 Special Note

1. In CTSDefine.inc, it has five user's define areas. Programmers need to define the parameters in these five areas before use CTS driver.

```
//==============User Define Area I ============================================
D_SinglePadNum:          .EQU   8        //CTS Pad Number
D_TraceUpStep:           .EQU   2        //Refence value trace up Max step definatin
D_TraceDownStep:         .EQU   5        //Refence value trace down Max step definatin
D_CalibrationCount:      .EQU   0x30     //Reference value calibration count
D_ReferenceErrCnt:       .EQU   0x20     //Maximmum value is 127. Reference value error debounce
                                         //time defination.
D_CtsKeydebounceTime:    .EQU   2        //CTS key debounce count
D_ScanCycleCnt           .EQU   60       //Scan cycle

//==============User Define Area II ===========================================
//              PadIndex,   IOIndex
.define         CTS_Pad0    0
.define         CTS_Pad1    1
.define         CTS_Pad2    2
.define         CTS_Pad3    3
.define         CTS_Pad4    4
.define         CTS_Pad5    5
.define         CTS_Pad6    6
.define         CTS_Pad7    7
.define         CTS_Pad8    8
.define         CTS_Pad9    9
.define         CTS_Pad10   10
.define         CTS_Pad11   11
.define         CTS_Pad12   12
.define         CTS_Pad13   13
.define         CTS_Pad14   14
.define         CTS_Pad15   15

//============ User Define area III ==========================================
//              PadIndex,       ScanTime
.define         ScanNum         60

.define         Pad0_ScanNum    ScanNum
.define         Pad1_ScanNum    ScanNum
.define         Pad2_ScanNum    ScanNum
.define         Pad3_ScanNum    ScanNum
.define         Pad4_ScanNum    ScanNum
.define         Pad5_ScanNum    ScanNum
.define         Pad6_ScanNum    ScanNum
.define         Pad7_ScanNum    ScanNum
```

```
.define     Pad8_ScanNum   ScanNum
.define     Pad9_ScanNum   ScanNum
.define     Pad10_ScanNum ScanNum
.define     Pad11_ScanNum ScanNum
.define     Pad12_ScanNum ScanNum
.define     Pad13_ScanNum ScanNum
.define     Pad14_ScanNum ScanNum
.define     Pad15_ScanNum ScanNum

//========== User Define area IV ========================================
//          PadIndex,       OnLevel
.define     TriggleLv       540
.define     OnLv            TriggleLv*7/10
.define     Pad0_OnLv       OnLv
.define     Pad1_OnLv       OnLv
.define     Pad2_OnLv       OnLv
.define     Pad3_OnLv       OnLv
.define     Pad4_OnLv       OnLv
.define     Pad5_OnLv       OnLv
.define     Pad6_OnLv       OnLv
.define     Pad7_OnLv       OnLv
.define     Pad8_OnLv       OnLv
.define     Pad9_OnLv       OnLv
.define     Pad10_OnLv      OnLv
.define     Pad11_OnLv      OnLv
.define     Pad12_OnLv      OnLv
.define     Pad13_OnLv      OnLv
.define     Pad14_OnLv      OnLv
.define     Pad15_OnLv      OnLv

//========== User Define area V =========================================
//          PadIndex,       OffLevel
.define     OffLv           TriggleLv*3/10
.define     Pad0_OffLv      OffLv
.define     Pad1_OffLv      OffLv
.define     Pad2_OffLv      OffLv
.define     Pad3_OffLv      OffLv
.define     Pad4_OffLv      OffLv
.define     Pad5_OffLv      OffLv
.define     Pad6_OffLv      OffLv
.define     Pad7_OffLv      OffLv
.define     Pad8_OffLv      OffLv
.define     Pad9_OffLv      OffLv
.define     Pad10_OffLv     OffLv
```

```
.define     Pad11_OffLv     OffLv
.define     Pad12_OffLv     OffLv
.define     Pad13_OffLv     OffLv
.define     Pad14_OffLv     OffLv
.define     Pad15_OffLv     OffLv
```