To our customers,

RENESAS

# RENESAS

**Application Note**

**Multimedia Processor for Mobile Applications**

# ITU-R BT.656 Interface

---------------------------------------------------------------------------------------

**EMMA Mobile 1**

# PREFACE

**Purpose**　　　　　The purpose of this document is to introduce the usage of EMMA Mobile 1 ITU-R BT.656 (NTS) interface.

**Organization**　　　This document includes the following:

- Chapter 1. Overview
- Chapter 2. Usage of NTS Interface
- Chapter 3. Sample of NTS Operation
- Appendix NTS Driver Function

**Notation**　　　　　Here explains the meaning of following words in text:

　　　　　　　　　**Note**　　　　　Explanation of item indicated in the text

　　　　　　　　　**Caution**　　　Information to which user should afford special attention

　　　　　　　　　**Remark**　　　Supplementary information

**Related document**　The following tables list related documents.

### Reference Document

| Document Name | Version/date | Author | Description |
|---|---|---|---|
| S19268EJ1V0UM00_1chip.pdf | 1st edition | NECEL | User's Manual |
| S19265EJ1V0UM00_ASMUGIO.pdf | 1st edition | NECEL | User's Manual |
| S19257EJ1V0UM00_ITU-R.pdf | 1st edition | NECEL | User's Manual |
| S19907EJ1V0AN00_GD.pdf | 1st edition | NECEL | GD Spec |
| S19901EJ1V0AN00_SPI.pdf | 1st edition | NECEL | Application Note |
| S19905EJ1V0AN00_I2C.pdf | 1st edition | NECEL | Application Note |
| ADV7179KCP.pdf | Rev. A | Analog Devices | Specification |
| | | | |

**Disclaimers**

- **The information contained in this document is subject to change without prior notice in the future. Refer to the latest applicable data sheet(s) and user's manual when designing a product for mass production.**

- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this documents or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customers' equipment shall be done under the full responsibility of the customer. NEC Electronics assume no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

Note)
1. "NEC Electronics" as used in this document means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
2. "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above)
3. All trademarks or registered trademarks are the property of their respective owners. Registered trademarks ® and trademarks™ are not noted in this document.

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1  Overview

## 1.1 Introduction

In this document, the below contents of EMMA Mobile 1 ITU-R BT.656 (NTS) interface will be described.

1) the normal process procedure of NTS interface
2) usage sample of NTS

As additional, the EMMA Mobile 1 NTS driver interface of EMMA Mobile 1 evaluation program will be explained.

About detail of NTS interface, please refer to "**EMMA Mobile 1 ITU-R BT.656 Interface User's Manual**".

## 1.2 Development Environment

● Hardware environment of this project is listed as below.

**Table 1-1 Hardware Environment**

| Name | Version | Maker |
|---|---|---|
| EMMA Mobile 1 evaluation board (PSKCH2Y-S-0016-01) | - | NEC Electronics |
| PARTNER-Jet ICE ARM | M20 | Kyoto Microcomputer Co. Ltd |

● Software used in this project is listed as below.

**Table 1-2 Software Environment**

| Name | Version | Maker |
|---|---|---|
| GNUARM Toolchain | V4.3.2 | GNU |
| WJETSET-ARM | V5.10a | Kyoto Microcomputer Co. Ltd |

# Chapter 2  Usage of NTS Interface

EMMA Mobile 1 NTS interface supports 2 standards: NTSC and PAL.

Operation flows of NTS interface:

1) fetches YUV422 image data form a frame buffer;

2) converts it to ITU-R BT.656-compliant parallel data;

3) outputs to an external NTSC/PAL encoder IC

## 2.1 Normal Procedure of NTS Operation

Normal NTS operation procedure is shown as below.

```
                    ╭─────────────╮
                    │    START    │
                    ╰──────┬──────╯
                           ▼
                ┌─────────────────────┐
                │  NTS Initialization │
                └──────────┬──────────┘
                           ▼
            ┌─────────────────────────────┐
            │ External Encoder IC Setting │
            └──────────────┬──────────────┘
                           ▼
                ┌─────────────────────┐
                │  NTS Configuration  │
                └──────────┬──────────┘
                           ▼
                ┌─────────────────────┐
                │      NTS Start      │
                └──────────┬──────────┘
                           ▼
              ┌ ┌───────────────────┐ ┐
              │ │  NTS Processing   │ │
              └ └───────────────────┘ ┘
                           ▼
                ┌─────────────────────┐
                │      NTS Reset      │
                └──────────┬──────────┘
                           ▼
            ┌─────────────────────────────┐
            │  External Encoder IC Reset  │
            └──────────────┬──────────────┘
                           ▼
                    ╭─────────────╮
                    │     END     │
                    ╰─────────────╯
```
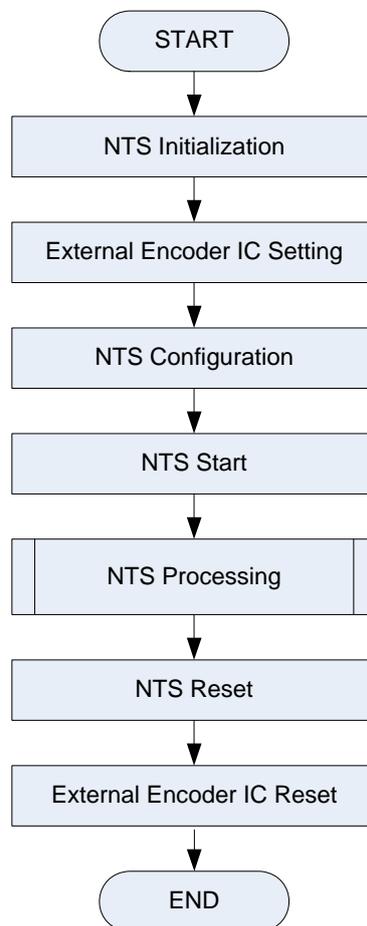
Figure 1-1 Normal NTS Operation Flow

## 2.2 Detail of Normal NTS Operation Procedure

### 2.2.1 NTS Initialization

When do NTS initializations, unreset NTS clock by setting ASMU register as below. It's necessary to switch the alternate pin to NTS function.

Related register:
RESETREQ0;
RESETREQ0ENA;
AHBCLKCTRL0;
APBCLKCTRL0;
GCLKCTRL2;
GCLKCTRL2ENA;
CHG_PINSEL_G64;
CHG_PINSEL_G80;
CHG_PULL_G08;

Explanation:
GIO P72 ~ P81 are alternated with NTS and other modules.
When switch the alternate pins to NTS function:
bit [31:16] of the register "CHG_PINSEL_G64" are set to "01 01 01 01 01 01 01 01";
bit [5:0] of the register "CHG_PINSEL_G80" are set to "01 01 01".
GIO P72 is alternated with NTS_CLK pin.

For NTS clock setting, configure "CHG_PULL_G72" as below:

| Signal | Setting | Function |
|--------|---------|----------|
| IE | 1 | Allows input |
| UPC | 0 | Not care |
| POENB | 1 | Disable Pull-Up/Down |

**Note**:
1. More detail about the CHG registers, please refer to "**EMMA Mobile 1 One Chip User's Manual**".
2. About the explanation of all the ASMU registers mentioned in this document, please refer to "**EMMA Mobile 1 ASMU/GIO Interface User's Manual**".

### 2.2.2 Configure the External Encoder IC

Configure the external encoder IC according to its' specification or user's manual.

### 2.2.3 Configure NTS module

NTS interface supports PAL mode and NTSC mode. So configure the NTS module according to the specified output mode;
Set NTS interrupt register as necessary.

Related registers:
 NTS_CONTROL;
 NTS_FRAMESEL;
 NTS_YAREAAD_A/B/C;
 NTS_UVAREAAD_A/B/C;
 NTS_HOFFSET;
 NTS_INTENCLR;
 NTS_FFCLR;
 NTS_INTENSET;

### 2.2.4 Start NTS Processing

Start NTS processing by setting NTS_OUT to the not-0 value.
Different setting value performs the different output mode.

**Note**:

| Name | Setting | Function |
|---|---|---|
| NTS_OUT | 00b | Output OFF (all-0 data is output) |
| | 01b | Blackback output |
| | 10b | Blueback output |
| | 11b | Normal output |

### 2.2.5 NTS Processing

After start the NTS processing, the source data are fetched and converted to the NTS compliant parallel data. Then output to the external encoder NTSC/PAL IC.

### 2.2.6 NTS Reset

Reset NTS interface by setting the related ASMU registers, when exit from NTS function.

Related register:
RESETREQ0;
RESETREQ0ENA;
AHBCLKCTRL0;
APBCLKCTRL0;
GCLKCTRL2;
GCLKCTRL2ENA;

### 2.2.7 Reset the External Encoder IC

It's also necessary to reset and power off the external NTSC/PAL encoder IC, when exit from NTS function.

About the reset operation of the external encoder IC, please refer to its' specification or user's manual.

# Chapter 3 Sample of NTS Operation

## 3.1 Outline of NTS Operation Sample

On EMMA Mobile 1 evaluation board (PSKCH2Y-S-0016-01), ADV7179 (manufacture: ANALOG DEVICES) is connected as the external NTSC/PAL encoder IC.

For show the usage of NTS interface with ADV7179, the below 2 samples are performed.

● NTSC mode; (NTSC and ADV7179 output)

● PAL mode; (PAL and ADV7179 output)

In the NTS samples, the data is transmitted as below:
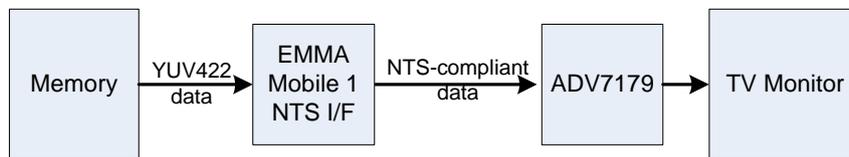


Figure 3-1 Data Flow Chart of NTS Samples

As shown in the figure, EMMA Mobile 1 NTS will fetch the source YUV422 image data, and convert to NTS compliant data, then output to ADV7179.

1 set of TV monitor is connected to display the processed data via ADV7179.

**Caution**:

Please make sure all hardware for NTS interface are on READY status, before perform the NTS samples.

## 3.2 Operation Flow of NTS Example

Operation flow chart of the NTS sample is shown as below.

```
                        ┌─────────────┐
                        │    START    │
                        └─────────────┘
                               │
          ┌────────────────────────────────────────┐
          │     Prepare source data for NTS sample  │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │            NTS Initialization           │
          │ 1] Open clockgate                       │
          │    GCLKCTRL2ENA[24:23] = 11b;           │
          │    GCLKCTRL2[24:23] = 11b;              │
          │    GCLKCTRL2ENA[24:23] = 00b;           │
          │ 2] cancel reset device                  │
          │    RESETREQ0ENA[19] = 1b;               │
          │    RESETREQ0[19] = 1b;                  │
          │    RESETREQ0ENA[19] = 0b;               │
          │ 3] enable auto clock control            │
          │    AHBCLKCTRL0[24] = 1b;                │
          │    APBCLKCTRL0[12] = 1b;                │
          │ 4] switch CHG register                  │
          │    CHG_PINSEL_G64[31:16] = 5555H        │
          │    CHG_PINSEL_G80[5:0] = 010101b        │
          │    CHG_PULL_G72[2:0] = 101b             │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │         External Encoder IC Setting     │
          │ 1] Reset ADV7179                        │
          │ 2] Init ADV7179 as NTSC mode output     │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │            NTS Configuration            │
          │      (According the different sample)    │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │               NTS Start                 │
          │   [ NTS_OUT = x ]   (x = 1, 2 or 3)     │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │             NTS Processing              │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │               NTS Reset                 │
          │ 1] Reset device                         │
          │    RESETREQ0ENA[19] = 1b;               │
          │    RESETREQ0[19] = 0b;                  │
          │    RESETREQ0ENA[19] = 0b;               │
          │ 2] Disable auto clock control           │
          │    AHBCLKCTRL0[24] = 0b;                │
          │    APBCLKCTRL0[12] = 0b;                │
          │ 3] Close clockgate                      │
          │    GCLKCTRL2ENA[24:23] = 11b;           │
          │    GCLKCTRL2[24:23] = 00b;              │
          │    GCLKCTRL2ENA[24:23] = 00b;           │
          └────────────────────────────────────────┘
                               │
          ┌────────────────────────────────────────┐
          │       Release the External Encoder IC   │
          │ 1] Reset ADV7179                        │
          └────────────────────────────────────────┘
                               │
                        ┌─────────────┐
                        │     END     │
                        └─────────────┘
```

Figure 3-2 Operation Flow of NTS Sample

## 3.3 Detail of NTS Sample

### 3.3.1 Prepare Source Data for NTS Sample

Load YUV422 data to frame A and frame B for NTS sample.

Data stored in frame A is prepared for NTSC sample;

Data stored in frame B is prepared for PAL sample.

### 3.3.2 NTS Initialization

Set ASMU and CHG registers, as listed in figure 3-2.

### 3.3.3 ADV7179 Setting

Reset and Initialize ADV7179.

Reset ADV7179 by call SPI0 driver interface:

1) Initialize EMMA Mobile 1 SPI0 interface

2) Set GPIO10 of PMIC (reset pin) to output mode

3) Make GPIO10 output '0' and wait for stable

4) Make GPIO10 output '1'

Initialize ADV7179 by call SPI0 and I2C driver interface:

1) Initialize EMMA Mobile 1 I2C interface

2) Enable power output for ADV7179

3) Set ADV7179 with the default value of NTSC mode

**Note**:

Detail of SPI0 driver interface, please refer "**EMMA Mobile 1 SPI Application Note**".

Detail of SPI0 driver interface, please refer "**EMMA Mobile 1 I2C Application Note**".

### 3.3.4 NTS Configuration

For different sample, the configuration is different.

**Table 3-1 Configuration of NTS Sample**

| Sample | Module | Register | Setting |
|--------|--------|----------|---------|
| NTSC | NTS I/F | NTS_CONTROL | = 0;<br>{<br>UPSCALE = 0b: disable upscale function ;<br>OUTMODE = 0b: NTSC mode ;<br>CLKPOL = 0b: rising edge of NTS_CLKI ;<br>ENDIAN = 0b: Little endian ;<br>} |
| | | NTS_YAREAAD_A | = 0x3100_0000 |
| | | NTS_UVAREAAD_A | = 0x3110_0000 |
| | | NTS_HOFFSET | = 720 |
| | | NTS_FRAMESEL | = 1<br>{<br>AREASEL = 01b: frame buffer A ;<br>} |
| | ADV7179 | Set ADV7179 registers with NTSC setting values (**NOTE**), please refer **Table 3-2** | |
| PAL | NTS I/F | NTS_CONTROL | = 4;<br>{<br>UPSCALE = 0b: disable upscale function ;<br>OUTMODE = 1b: PAL mode ;<br>CLKPOL = 0b: rising edge of NTS_CLKI ;<br>ENDIAN = 0b: Little endian ;<br>} |
| | | NTS_YAREAAD_A | = 0x3120_0000 |
| | | NTS_UVAREAAD_A | = 0x3130_0000 |
| | | NTS_HOFFSET | = 720 |
| | | NTS_FRAMESEL | = 2<br>{<br>AREASEL = 10b: frame buffer B ;<br>} |
| | ADV7179 | Set ADV7179 registers with PAL setting values (**NOTE**), please refer **Table 3-2** | |

**Note**:

1. Default setting value sequence of ADV7179 for NTSC/PAL mode are listed in the below table.

**Table 3-2 Default Setting Value of ADV7179 for NTSC/PAL Mode**

| ADV7179 Register | | Mode | |
|---|---|---|---|
| Name | Address | NTSC | PAL |
| Mode Register 0 [MR0] | 00 | 0x00 | 0x05 |
| Mode Register 1 [MR1] | 01 | 0x10 | 0x10 |
| Mode Register 2 [MR2] | 02 | 0x00 | 0x00 |
| Mode Register 3 [MR3] | 03 | 0x00 | 0x00 |
| Mode Register 4 [MR4] | 04 | 0x10 | 0x00 |
| Reserved | 05 | 0x00 | 0x00 |
| Reserved | 06 | 0x00 | 0x00 |
| Timing Mode Register 0 [TR0] | 07 | 0x00 | 0x00 |
| Timing Mode Register 1 [TR1] | 08 | 0x00 | 0x00 |
| Subcarrier Freq Reg 0 | 09 | 0x1E | 0xCB |
| Subcarrier Freq Reg 1 | 0A | 0x7C | 0x8A |
| Subcarrier Freq Reg 2 | 0B | 0xF0 | 0x09 |
| Subcarrier Freq Reg 3 | 0C | 0x21 | 0x2A |
| Subcarrier Phase Reg | 0D | 0x00 | 0x00 |
| Closed Captioning Ext Data Byte 0 | 0E | 0x00 | 0x00 |
| Closed Captioning Ext Data Byte 1 | 0F | 0x00 | 0x00 |
| Closed Captioning Data Byte 0 | 10 | 0x00 | 0x00 |
| Closed Captioning Data Byte 1 | 11 | 0x00 | 0x00 |
| NTSC Pedestal Ctrl Reg 0 / PAL TTX Ctrl Reg 0 | 12 | 0x00 | 0x00 |
| NTSC Pedestal Ctrl Reg 1 / PAL TTX Ctrl Reg 1 | 13 | 0x00 | 0x00 |
| NTSC Pedestal Ctrl Reg 2 / PAL TTX Ctrl Reg 2 | 14 | 0x00 | 0x00 |
| NTSC Pedestal Ctrl Reg 3 / PAL TTX Ctrl Reg 3 | 15 | 0x00 | 0x00 |
| CGMS_WSS_0 [C/W0] | 16 | 0x00 | 0x00 |
| CGMS_WSS_1 [C/W1] | 17 | 0x00 | 0x00 |
| CGMS_WSS_2 [C/W2] | 18 | 0x00 | 0x00 |
| TeleText Request Ctrl Reg [TC07] | 19 | 0x00 | 0x00 |

More detail about ADV7179 registers, please refer "**ADV7179 Specification**".

### 3.3.5 Start NTS Processing

Start NTS processing by setting "NTS_OUT" with not-0 value.
For NTSC and PAL sample, set the register to "3"; it is normal output mode.

### 3.3.6 NTS Processing

After start the NTS processing, the source data are fetched and converted to the NTS compliant parallel data. Then output to ADV7179. It will be displayed in the connected TV monitor finally.

### 3.3.7 NTS Reset

After complete NTS NTSC and PAL function evaluation, reset NTS interface by setting ASMU registers as listed in figure 3-2.

### 3.3.8 Release ADV7179

Reset and power off ADV7179 as the below sequence.
1) Initialize SPI0
2) Set GPIO10 of PMIC (reset pin) to output mode
3) Make GPIO10 output '0'

## 3.4 ADV7179 Color Bar Function

As additional, ADV7179 color bar function is evaluated as an individual sample.

It will generate and output an internal color bar test pattern. It is important to note that when color bars are enabled, the ADV7179 is configured in a master timing mode.

Under this sample, ADV7179 is used only. So the operations of NTS are unnecessary.

Operation of this sample is as below:

1) ADV7179 Setting
   Same with "**3.3.3 ADV7179 Setting**"
2) Set ADV7179 to color bar mode
   Set bit[7] of ADV7179 Mode Register 1 with "1", to enable color bar function.
3) Check the color bar pattern displayed in TV monitor
4) Reset and power off ADV7179
   Same with "**3.3.8 Release ADV7179**".

# Appendix A. NTS Driver Function

## A.1 NTS API function list

The following table shows the NTS interface functions:

**Table A-1 NTS Driver Function List**

| Type | Function Name | Function Detail |
|------|---------------|-----------------|
| Driver Function | em1_nts_init | Initialize NTS interface |
| | em1_nts_release | Release NTS interface |
| | em1_nts_setup | Configure NTS interface |
| | em1_nts_start | Start NTS processing |
| | em1_nts_irq_enable | Enable NTS Interrupt |
| | em1_nts_irq_handle | NTS Interrupt handler |

## A.2 Type Define

### A.2.1 Naming rule and coding rule

About naming rule and coding rule, please refer to "**GD_SPEC_EM1_AN&TP.pdf**"

### A.2.2 Structure

**Table A-2 Structure Define**

| Structure Name | Structure Member |
|----------------|------------------|
| NTS_SETUP_ST | uchar upscale |
| | uchar outmode |
| | uint clkpol |
| | uint endian |
| | uint buffer_a_y |
| | uint buffer_b_y |
| | uint buffer_c_y |
| | uint buffer_a_uv |
| | uint buffer_b_uv |
| | uint buffer_c_uv |
| | uint hoffset |
| | uint frame_sel |

## A.3 Function Detail

### A.3.1 Initialization Function

**[Function Name]**

em1_nts_init

**[Format]**

void em1_nts_init(void);

**[Argument]**

None

**[Function Return]**

None

**[Function Flow]**

```
                          ┌─────────────┐
                          │    START    │
                          └─────────────┘
                                 │
                                 ▼
                    ┌───────────────────────────┐
                    │   Open Gateclock for NTS   │
                    └───────────────────────────┘
                                 │
                                 ▼
                    ┌───────────────────────────┐
                    │ Enable NTS auto clock control │
                    └───────────────────────────┘
                                 │
                                 ▼
                    ┌───────────────────────────┐
                    │        Unreset NTS         │
                    └───────────────────────────┘
                                 │
                                 ▼
                    ┌───────────────────────────┐
                    │      Switch CHG to NTS     │
                    └───────────────────────────┘
                                 │
                                 ▼
                          ┌─────────────┐
                          │     END     │
                          └─────────────┘
```
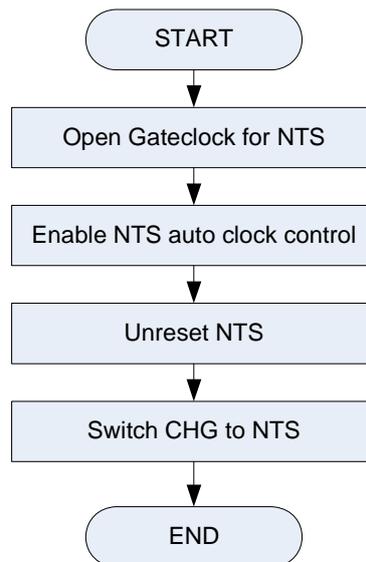
Figure A-1 NTS Initialization

**[Note]**

None

**A.3.2 Release Function**

**[Function Name]**

em1_nts_release

**[Format]**

void em1_nts_release(void);

**[Argument]**

None

**[Function Return]**

None

**[Function Flow]**

```
        ┌─────────────┐
        │    START    │
        └─────────────┘
               │
               ▼
    ┌──────────────────────┐
    │      Reset NTS       │
    └──────────────────────┘
               │
               ▼
    ┌──────────────────────────────┐
    │ Dsiable NTS auto clock control│
    └──────────────────────────────┘
               │
               ▼
    ┌──────────────────────┐
    │  Close NTS Gateclock │
    └──────────────────────┘
               │
               ▼
        ┌─────────────┐
        │     END     │
        └─────────────┘
```
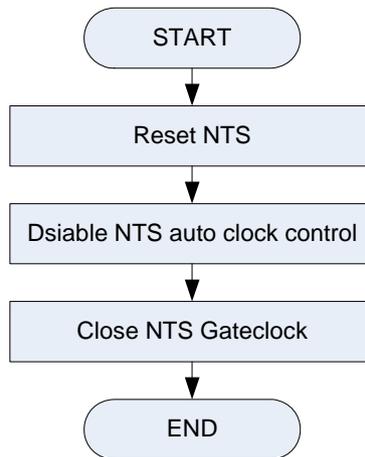
Figure A-2 NTS Release

**[Note]**

None

### A.3.3 Setup Function

**[Function Name]**

em1_nts_setup

**[Format]**

DRV_RESULT em1_nts_setup (NTS_SETUP_ST* nts_st);

**[Argument]**

| Parameter | Type | I/O | Detail |
|-----------|------|-----|--------|
| nts_st | NTS_SETUP_ST * | I | All necessary setting for NTS |

**[Function Return]**

DRV_ERR_PARAM;

DRV_OK;

**[Function Flow]**



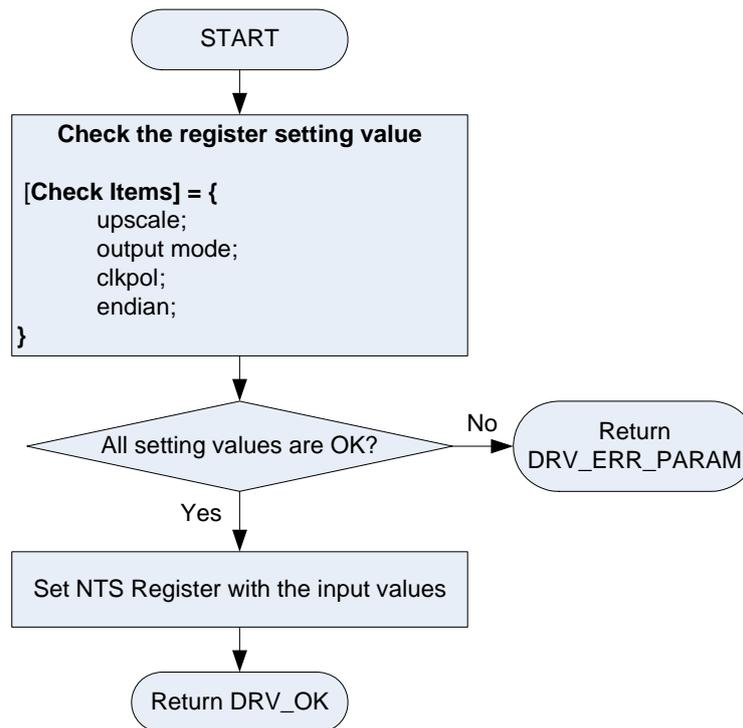Figure A-3 NTS Setup

**[Note]**

None

### A.3.4 Start Function

**[Function Name]**

em1_nts_start

**[Format]**

void em1_nts_start(uint display_mode);

**[Argument]**

| Parameter | Type | I/O | Detail |
|-----------|------|-----|--------|
| display_mode | uint | I | output mode :<br>  OFF;<br>  Blackback;<br>  Blueback;<br>  Normal; |

**[Function Return]**

None

**[Function Flow]**

START

Start NTS Processing
[   NTS_OUT = **x**   ]
(**x** = 1, 2, or 3)

END

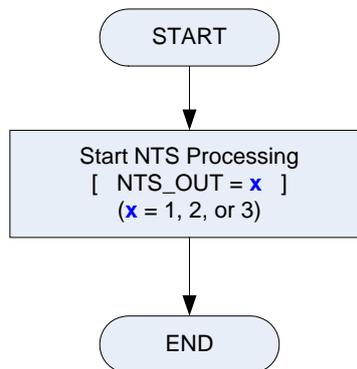Figure A-4 NTS Start

**[Note]**

None

## A.3.5 INT Enable Function

### [Function Name]

em1_nts_irq_enable

### [Format]

void em1_nts_irq_enable(void);

### [Argument]

None

### [Function Return]

None

### [Function Flow]

None

### [Note]

None

**A.3.6 INT Handler Function**

**[Function Name]**

em1_nts_irq_handle

**[Format]**

void em1_nts_irq_handle(void);

**[Argument]**

None

**[Function Return]**

None

**[Function Flow]**

None

**[Note]**

None

# ANNEX Modification History

| Number | Modification Contents | Author | Date |
|---|---|---|---|
| Ver 1.00 | New version | | Aug, 4, 2009 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |