AN3127
Application note

CEC networking
using STM32F100xx value line microcontrollers

Introduction

STM32F100xx value line microcontrollers feature a high-definition multimedia interface consumer electronics control (HDMI™-CEC) controller peripheral that supports the HDMI-CEC v1.3a protocol. The HDMI-CEC controller provides a hardware support of this protocol, and it supports the whole set of features offered with CEC devices.

This application note describes the CEC protocol software and hardware implementation based on the HDMI-CEC controller. A real application example is also provided to illustrate the software implementation.

Contents

List of tables

List of figures

Glossary of terms

Broadcast message: a message sent to logical address 15 that all devices are expected to receive.

Destination: the target device for a CEC message.

Follower: a device that has just received a CEC message and is required to respond to it.

Initiator: the device that is sending, or has just sent, a CEC message and, if appropriate, is waiting for a follower to respond.

Logical address: a unique address assigned to each device.

Menu-providing device: a non-display device that may render a menu on TV.

Playback device: a device that has the ability to play media, like a DVD player.

Recording device: a device that has the ability to record a source such as an internal tuner or an external connection.

Source device: a device that is currently providing an AV stream via HDMI.

Tuner device: a device that contains a tuner, e.g. an STB (set-top box) or a recording device.

TV: a device with a HDMI input that has the ability to display the input HDMI signal. Generally it has no HDMI output.

CEC: consumer electronics control.

DDC: display data channel.

E-DDC: enhanced display data channel.

EDID: extended display identification data.

E-EDID: enhanced extended display identification data.

HDMI: high-definition multimedia Interface.

(HDMI) source: a device with a HDMI output.

(HDMI) sink: a device with a HDMI input.

1 High-definition multimedia interface, consumer electronics control (HDMI-CEC)

1.1 Introduction

Consumer electronics control (CEC) is the appendix supplement 1 to the HDMI (high-definition multimedia interface) standard.

It is a protocol that provides high-level control functions between all of the various audiovisual products in a given environment. It is specified to operate at low speeds with minimum processing and memory overhead.

For more details, refer to the high-definition multimedia interface specification available from *www.hdmi.org*.

1.2 Frame description

The CEC bus is a single-wire protocol that can connect up to 10 audiovisual devices through standard HDMI cabling.

All transactions on the CEC line consist of an initiator and one or more followers. The initiator is responsible for sending the message structure and the data. The follower is the recipient of any data and is responsible for setting any acknowledgement bits.

A message is conveyed in a single frame that consists of a start bit followed by a header block and, optionally, an opcode and a variable number of operand blocks. Figure 1 shows a CEC frame format.

Figure 1. CEC frame format

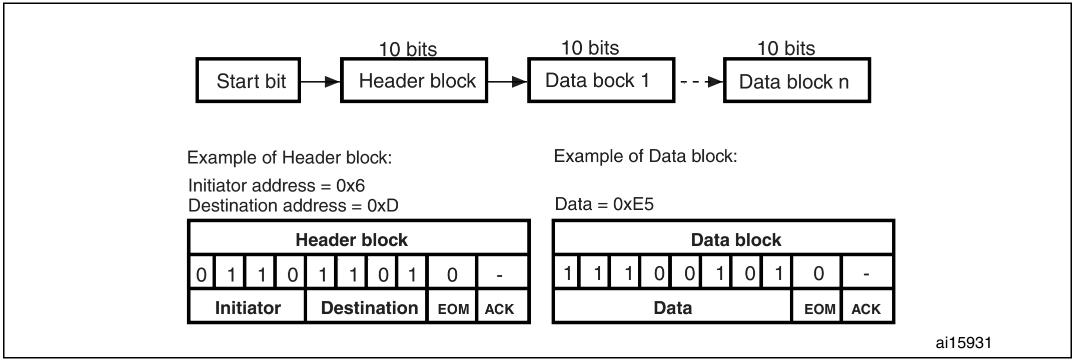


Figure 2. Message structure

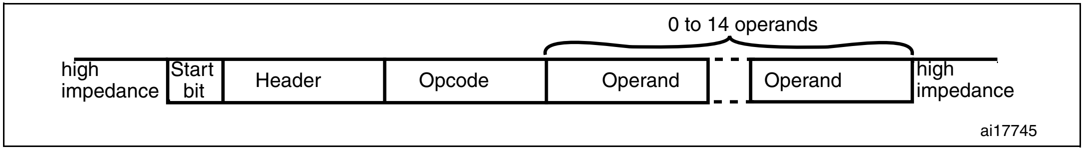
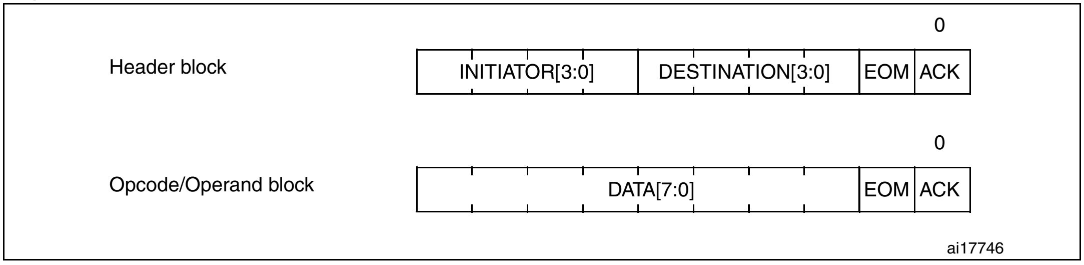


Figure 3. Blocks



All these blocks are made of an 8-bit payload (most significant bit transmitted first) followed by an end-of-message (EOM) bit and an acknowledge (ACK) bit.

The EOM bit is set in the last block of a message and kept cleared in all others. If a message contains additional blocks after the EOM, these blocks should be ignored. The EOM bit may be set in the header block to “ping” other devices and make sure that they are active.

The acknowledge bit is always brought to high impedance by the initiator. It can therefore be driven low by either the follower that has read its own address in the header, or the follower that needs to reject a broadcast message.

The header consists of the source logical address field, and the destination logical address field. Note that the address 0xF is specially used for broadcast messages.

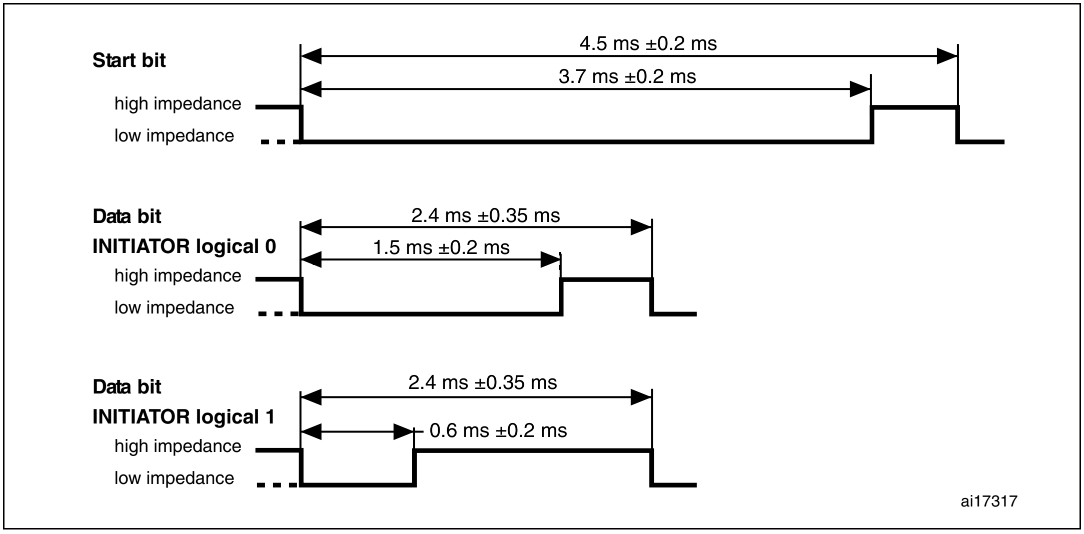
1.3 Bit timing

The format of the start bit is unique and identifies the start of a message. It should be validated by its low duration and its total duration.

All remaining data bits in the message, after the start bit, have a consistent timing. The high-to-low transition at the end of the data bit is the start of the next data bit except for the final bit where the CEC line remains high.

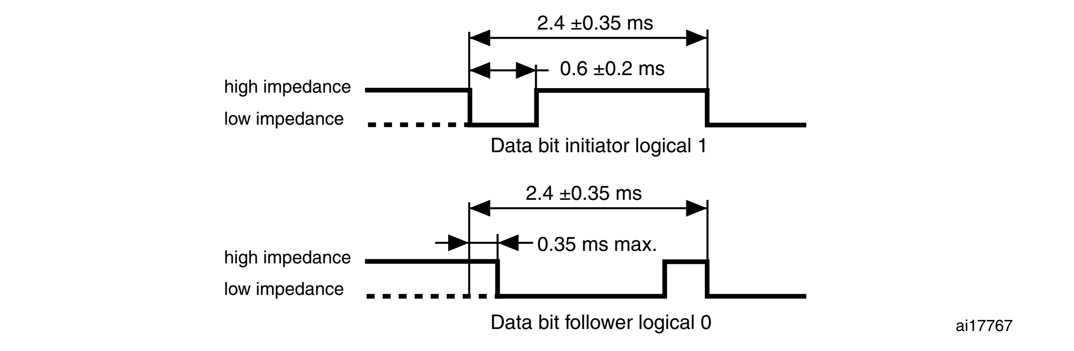
Figure 4 shows the timings of the start bit and the different data bits.

Figure 4. Bit timings



CEC *Figure 5* shows an example bit with both initiator and follower where the follower may assert the bit to logical 0 to acknowledge a data block. The initiator outputs a logical 1, thus allowing the follower to change the CEC state by pulling the control line low for the duration of the safe sample period.

Figure 5. Follower acknowledge (ACK)



1.4 Device connectivity and addressing

1.4.1 CEC communication

By definition the HDMI system architecture consists of sources and sinks. A given device may have one or more HDMI input(s) and output(s). Each HDMI input on the device should follow all the rules for a HDMI sink and each HDMI output should follow all the rules for a HDMI source.

The DDC is used for configuration and status exchange between a single source and a single sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

Figure 6. CEC and DDC line connections

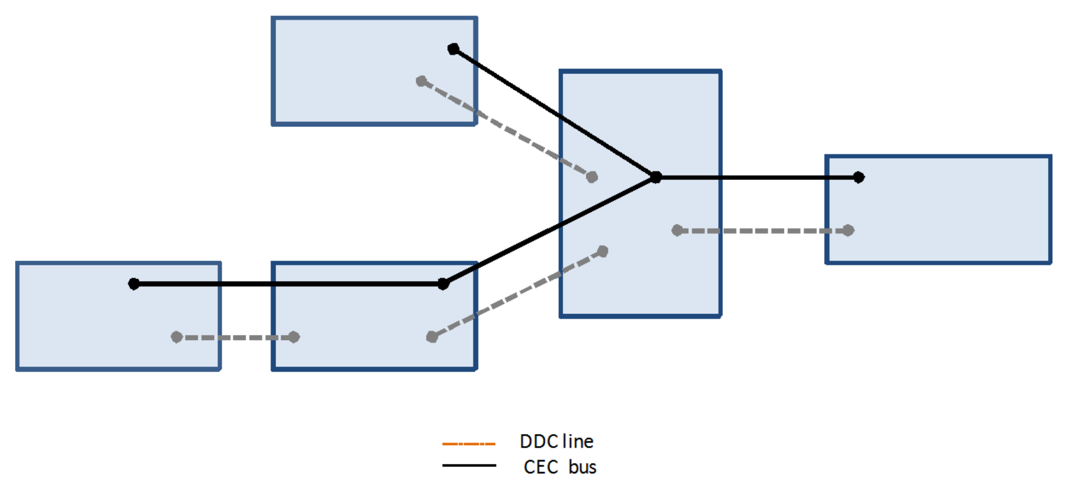
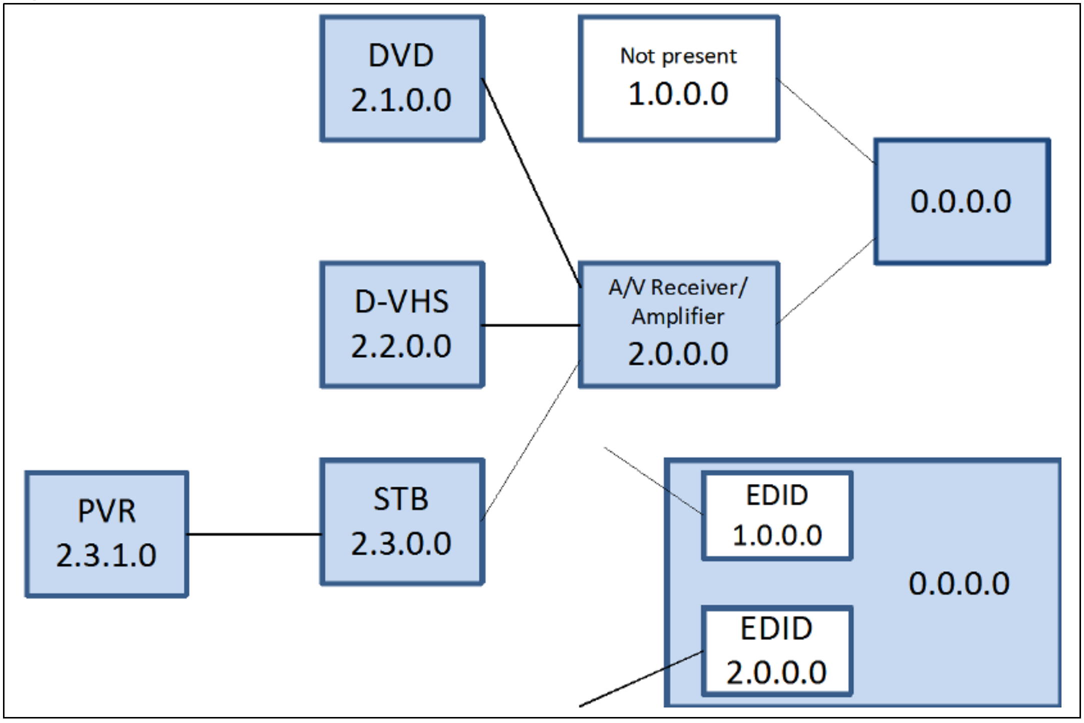


Figure 7. Addresses within a HDMI cluster



1.4.2 Enhanced DDC

The enhanced DDC described in this section is defined in VESA “ENHANCED DISPLAY DATA CHANNEL STANDARD Version 1 (September 2, 1999)”. All sinks are required to support these enhanced DDC features. If the E-EDID structure of a sink is longer than 256 bytes, it should support the segment pointer.

Timing

Data is synchronized with the SCL signal and timing should comply with the Standard Mode of the I2C specification (100 kHz maximum clock rate).

The I2C bus is a standard two-wire (clock and data) serial databus protocol. Refer to the I2C specification for details.

Note that a HDMI sink may hold off the DDC transaction by stretching the SCL line during the SCL-low period following the Acknowledge bit as permitted by the I2C specification. All HDMI sources should delay the DDC transaction while the SCL line is being held low.

Data transfer protocols

The source should use I2C commands to read information from a sink’s E-EDID with a slave address.

In Enhanced DDC, a segment pointer is used to address the E-EDID outside the 256 bytes normally addressable by the 0xA0/0xA1 addresses. The Enhanced DDC protocol sets the segment pointer before the remainder of the DDC command.

Segment pointer

Enhanced DDC allows access to up to 32 Kbytes of data. This is accomplished using a combination of the 0xA0/0xA1 address pair and a segment pointer. For each value of the segment pointer, 256 bytes of data are available at the 0xA0/0xA1 address pair. An unspecified segment pointer references the same data as when the segment pointer is zero.

Each successive value of the segment pointer allows access to the next two blocks of E-EDID (128 bytes each). The value of the segment pointer register cannot be read since it is reset at the completion of each command.

Enhanced DDC sink

The sink should be Enhanced DDC read compliant.

The sink should be capable of responding with EDID 1.3 data and up to 255 extension blocks, each 128 bytes long (up to 32 Kbytes of E-EDID memory) whenever the hot plug detect (HPD) signal is asserted.

The sink should be capable of providing E-EDID information over the Enhanced DDC channel whenever the +5 V power signal is provided. The information should be available within 20 ms after the +5 V power signal is provided.

Enhanced DDC source

The source should use Enhanced DDC protocols.

It should be capable of reading EDID 1.3 data at DDC address 0xA0.

The source reads Enhanced EDID extension data at DDC address 0xA0 using segment pointer 0x60.

1.4.3 Hot plug detect (HPD) signal

A HDMI sink should not assert high voltage levels on its HPD pin when the E-EDID is not available for reading. This requirement should be fulfilled at all times, even if the sink is powered off or in standby. The HPD pin may be asserted only when the +5 V power line from the source is detected. This ensures that the HPD pin is not asserted before the third make of the connector.

A source may use a high voltage level HPD signal to initiate the reading of E-EDID data. It does not indicate whether the sink is powered, or whether the HDMI input on the sink is selected or active.

A HDMI sink should indicate any change to the contents of the E-EDID by driving a low voltage level pulse on the HPD pin. This pulse should last at least 100 ms.

1.4.4 Physical address discovery

In order to allow CEC to address specific physical devices, all have a physical address. The bus connectivity is worked out whenever the hot plug detect (HPD) signal is de-asserted by the physical address discovery process. That process uses only the DDC/EDID (display data channel/extended display identification data) mechanism. Starting from the CEC root device which takes address 0b0000 (normally the TV), all sinks and repeaters, whether CEC-capable or not, determine the address their source device should take, and make it available in the source address field of the EDID vendor-specific data block

(EDID_EXT_VSDB_B0-1). Thereby, except for the root device, all devices have the physical address stored in the EDID of their connected sink.

The physical address of each node is determined through the physical address discovery process. This process is dynamic in that it automatically adjusts physical addresses as required as devices are physically or electrically added or removed from the device tree.

All sinks and repeaters should perform the steps of physical address discovery and propagation even if those devices are not CEC-capable. Sources are not required to determine their own physical address unless they are CEC-capable.

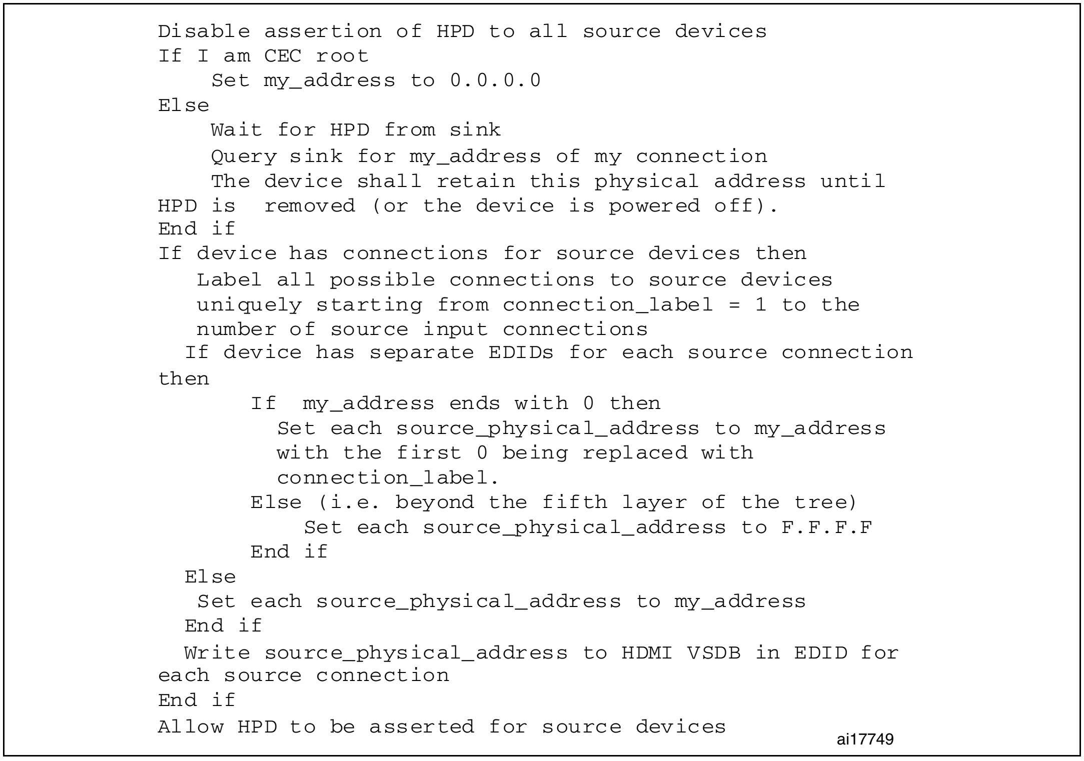
All addresses are 4 digits long allowing for a 5-device-deep hierarchy. All are identified in the form of n.n.n.n in the following description.

A sink or repeater that is acting as the CEC root device generates its own physical address: 0.0.0.0. The source or repeater reads its physical address from the EDID of the connected sink. The CEC line may be connected to only one HDMI output so a device with multiple HDMI outputs will read its physical address from the EDID on the CEC-connected output. Each sink and repeater is responsible for generating the physical address of all source devices connected to that device by appending a port number onto its own physical address and placing that value in the EDID for that port.

1.4.5 Discovery algorithm

The following algorithm is used to allocate the physical address of each device whenever HPD is de-asserted, or upon power-up:

Figure 8. Physical address discovery algorithm



```
Disable assertion of HPD to all source devices
If I am CEC root
    Set my_address to 0.0.0.0
Else
    Wait for HPD from sink
    Query sink for my_address of my connection
    The device shall retain this physical address until
HPD is  removed (or the device is powered off).
End if
If device has connections for source devices then
   Label all possible connections to source devices
   uniquely starting from connection_label = 1 to the
   number of source input connections
  If device has separate EDIDs for each source connection
then
       If  my_address ends with 0 then
         Set each source_physical_address to my_address
         with the first 0 being replaced with
         connection_label.
       Else (i.e. beyond the fifth layer of the tree)
           Set each source_physical_address to F.F.F.F
       End if
  Else
   Set each source_physical_address to my_address
  End if
  Write source_physical_address to HDMI VSDB in EDID for
each source connection
End if
Allow HPD to be asserted for source devices
```

Whenever a new physical address (other than F.F.F.F) is discovered, a CEC device should:

- allocate the logical address (see *Section 1.4.7: Logical address allocation*)
- report the association between its logical and physical addresses by broadcasting <Report Physical Address>

This process allows any node to create a map of physical connections to logical addresses.

1.4.6 Logical addressing

Apart from the physical address, each device appearing on the control signal line has a unique logical address. This address defines a device type as well as being a unique identifier. This address is 0 for a TV set with physical address 0b0000 and 14 or even 15 otherwise. It is defined in the CEC_OAR register and in the upper nibble of the first byte of the transmitted message. All CEC devices therefore have both a physical and a logical address, whereas non-CEC devices only have a physical address.

Table 1. Logical addresses

Address	Device
0	TV
1	Recording device 1
2	Recording device 2
3	Tuner 1
4	Playback device 1
5	Audio system
6	Tuner 2
7	Tuner 3
8	Playback device 2
9	Recording device 3
10	Tuner 4
11	Playback device 3
12	Reserved
13	Reserved
14	Free use
15	Unregistered (as initiator address) broadcast (as destination address)

1.4.7 Logical address allocation

Note that a logical address should only be allocated when a device has a valid physical address (i.e. not F.F.F.F), at all other times a device should take the ‘Unregistered’ logical address (15).

Only the device at physical address 0.0.0.0 may take logical address TV (0). A TV at any other physical address will take the ‘free use’ (14) address. If address 14 is already allocated it will take the ‘unregistered’ address (15).

Reserved addresses should not be used at present and are reserved for future extensions to this specification.

Where more than one possible logical address is available for the given device type (e.g. Tuner 1, Tuner 2, etc.), an address allocation procedure should be carried out by a newly connected device. The device takes the first allocated address for that device type and sends a <Polling Message> to the same address (e.g. Tuner 1 -->Tuner 1). If the <Polling Message> is not acknowledged, then the device stops the procedure and retains that address.

If the first address is acknowledged, then the device takes the next address for that device type and repeats the process (e.g. Tuner 2 .. Tuner 2). Again, if the message is not acknowledged, the device keeps that address.

This procedure continues until all possible 'type-specific' addresses have been checked; if no 'type-specific' addresses are available the device should take the unregistered address (15). Note that several physical devices might be sharing this address.

A device may lose its logical address when it is disconnected or switched off. However, it may remember its previous logical address, so that the next time it is connected or switched on, it can begin the polling process at its previous logical address and try each other allowable logical address in sequence before taking the unregistered address. For example if an STB that was previously allocated address Tuner 2 is reconnected, it would poll Tuner 2, Tuner 3, Tuner 4 and Tuner 1 before taking the unregistered address.

If a device loses its physical address at any time (e.g. it is unplugged) then its logical address should be set to unregistered (15).

1.5 STM32F100xx's HDMI-CEC controller

The STM32F100xx's HDMI-CEC controller provides a hardware support of the CEC protocol.

1.5.1 Main features

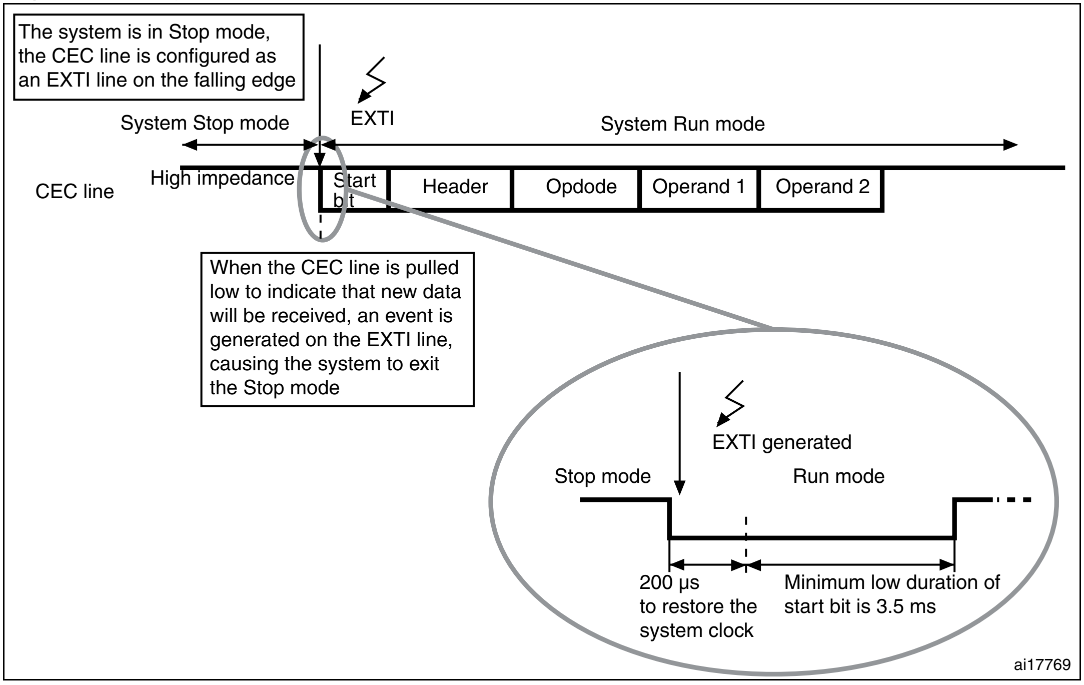
- Supports HDMI-CEC v1.3a
- Supports the whole set of features offered with CEC (devices may use all or only some of these features, depending on functionality):
  - One touch play - a device may be played and become the active source by pressing a single button.
  - System standby - enables devices to be set on standby by pressing a single button.
  - Preset transfer - the presets of a device can be autoconfigured to match those of the TV.
  - One touch record - used to make recordings by pressing a single button.
  - Timer programming - any device can program a timer to start/stop audio/video recording.
  - System information - allows devices to autoconfigure their language and country settings.
  - Deck control - allows a device to control and interrogate a playback device.
  - Tuner control - allows a device to control the tuner of another device.
  - Vendor-specific commands - allow vendor-defined commands to be used.
  - OSD display - allows a device to display text using the on-screen display of the TV.
  - Device menu control - allows a device to control the menu of another device.
  - Routing control - enables control of CEC switches for the streaming of a new source device.

1.5.2 HDMI-CEC advanced features

The STM32 HDMI-CEC controller provides an easy platform to build CEC firmware applications:

- dedicated address register containing up to 15 CEC device addresses
- Programmable high-resolution prescaler (14 bit) to obtain a 50 µs time base (HDMI specification resolution)
- advanced arbitration mechanism using SFT (signal free time) and header arbitration
- dedicated transmission and reception flags (TBTRF and RBTF) with interrupt capability
- error handling in a single dedicated register: bit timing errors, bit period errors, message errors and error bit generation
  - Tx block transfer finished error (TBTFE)
  - Line error (LINE)
  - Block acknowledge error (ACKE)
  - Start bit error (SBE)
  - Rx block transfer finished error (RTBFE)
  - Bit period error (BPE)
  - Bit timing error (BTE)
- error-free mode to allow safe communication with non-tolerant CEC devices. This mode can be selected using a dedicated configuration bit.
- Rx digital filtering
- CEC controller built around a flexible state machine to ensure safe communication, switching between the Tx and Rx modes, and easy error handling. There are 6 states: Disabled, Idle, RX, TX, RX_ERROR and TX_ERROR. Please refer to the reference manual (RM0041) available from *www.st.com* for more details.
- Wakeup-from-Stop source, with no data loss: the CEC line can be used as a wakeup event from the Stop mode. The STM32 Wakeup system is fast enough to ensure that there is no CEC data loss. CEC data received when the system is in the Stop mode cause the system to wake up from the low power mode, and in so doing, allow the CEC peripheral to easily retrieve them.

Figure 9. System exits Stop mode when data reception starts


The system is in Stop mode, the CEC line is configured as an EXTI line on the falling edge
EXTI
System Stop mode
System Run mode
CEC line
High impedance
Start bit
Header
Opdode
Operand 1
Operand 2
When the CEC line is pulled low to indicate that new data will be received, an event is generated on the EXTI line, causing the system to exit the Stop mode
EXTI generated
Stop mode
Run mode
200 µs to restore the system clock
Minimum low duration of start bit is 3.5 ms
ai17769

As the tolerance margin on the start bit is 200 µs, the software has to reconfigure the system clock within this time frame (see *Figure 9*). The system clock reconfiguration is necessary as after waking up from the Stop mode the CPU starts running on the high-speed internal oscillator (HSI 8 MHz).

To ensure that there is no data loss while the system is in the wakeup phase from the Stop mode, the following procedures can be selected:

- The software can choose to use the system PLL as the system main clock. The PLL source should be the HSI and not the HSE.
- For applications which do not need the full CPU performance, the software can choose to receive the first message just after waking up from Stop mode at the HSI frequency. In this case, there is no need to reconfigure the system clock after the CPU exits the Stop mode. However, just before entering the Stop mode the CEC prescaler should be reprogrammed to be in line with the HSI frequency by:
  - disabling the CEC peripheral. To do so, write PE=0 and wait until PE goes low
  - writing the CEC prescaler register (CEC_PRER = 399)
  - enabling the CEC peripheral. This is done by writing PE=1.
  - requesting Stop mode entry

Note: 1 *The HSE cannot be used after wakeup from Stop mode as it takes some time to start up, and CEC data would be lost.*

2 *Any data sent on the CEC bus cause the device to wake up even if the destination address is not matched.*

3 *After the first message is received, the software can restore the initial clock configuration to continue in Run mode or re-enter the Stop mode.*

Refer to the STM32F100xx reference manual (RM0041) for further details.

2 Hardware environment

To be fully compliant with the stringent HDMI 1.3a specification, specially when the CEC device is in the power-off state, a fully integrated ESD protection should be added externally to the STM32F100xx device. For that purpose, the HDMI2C1-5DIJ ensures level-shifting and serves as the signal booster for control links of HDMI 1.3 transmitters. The HDMI2C1-5DIJ is a bidirectional isolation buffer, integrating hysteresis and signal boosters for maximum system robustness and signal integrity.

In addition to this circuit, the 27 kΩ pull-up resistor on the CEC line should be externally connected to the STM32F100xx.

With these few hardware additions, the STM32F100xx CEC solution (CEC peripheral plus ESD protection circuit) is fully compliant with the HDMI 1.3 standard.

The solution is already implemented on Value line evaluation boards. Two HDMI connectors are available for each Value line evaluation board (CN15, CN16 for STM32100B-EVAL and CN2, CN3 for STM32100E-EVAL). The CEC, SCL, SDA and HPD signals are supported and connected to STM32F100xx through HDMI2C1-5DIJ.

The external hardware component is only needed when the CEC power-off state is needed in the application. If the power-off state is not needed, you can directly connect the STM32F100xx CEC line to the CEC bus.

2.1 HDMI connector

Only a vendor-approved, HDMI connector should be used. The list of approved HDMI connectors is available from the HDMI site (*www.hdmi.org*).

The differential impedance of this connector should be equal to 100 Ω ±15%. This can be checked by using time domain reflectometry (TDR) equipment.

Figure 10. HDMI cable

Table 2. HDMI connector pinout

Pin number	Signal name	Pin number	Signal name
1	TMDS Data 2+	11	TMDS Clock Shield
2	TMDS Data 2 Shield	12	TMDS Clock-
3	TMDS Data 2-	13	CEC
4	TMDS Data 1+	14	No Connect
5	TMDS Data 1 Shield	15	DDC Clock
6	TMDS Data 1-	16	DDC Data
7	TMDS Data 0+	17	Ground
8	TMDS Data 0 Shield	18	+5 V power
9	TMDS Data 0-	19	Hot Plug Detect
10	TMDS Clock+	-	-

2.1.1 I2C bus

The display data channel (DDC) I/Os and wires (SDA, SCL, DDC/CEC Ground), should meet the requirements specified in the I²C bus specification, version 2.1 (Section 15 for “Standard-Mode” devices).

Discussions about high-capacitance environments found in the I²C Specification section (17.2 “Switched pull-up circuit for Fast-mode I²C-bus devices”) also apply to the HDMI environment.

This section covers the electrical specifications for the I²C (defined by the CTS and by the I²C bus specification). Timing specifications are not dealt with here. The HDMI standard specifies that the maximum clock rate is 100 kHz.

The I²C bus is a standard two-wire (SCL for clock and SDA for data) serial databus protocol. The electrical specifications for I²C bus device I/Os are power-supply-dependent. An I²C bus device with fixed input levels of 1.5 V and 3 V can have its own supply voltage. A pull-up resistor has to be connected to a 5 V ±10% supply.

When devices with fixed input levels are mixed with devices with input levels dependent on V_{CC}, the latter have to be connected to a common 5 V ±10% supply line with pull-up resistors connected to their SDA and SCL pins.

The ST reference board implemented in this application note has 3.3 V-tolerant I/Os and a HDMI-compatible I²C bus dependent on a +5 V V_{CC} supply. For the implementation, it is recommended to use the setup shown in *Figure 11*, with the HDMI2C1-5DIJ chip.

Figure 11. Application schematic

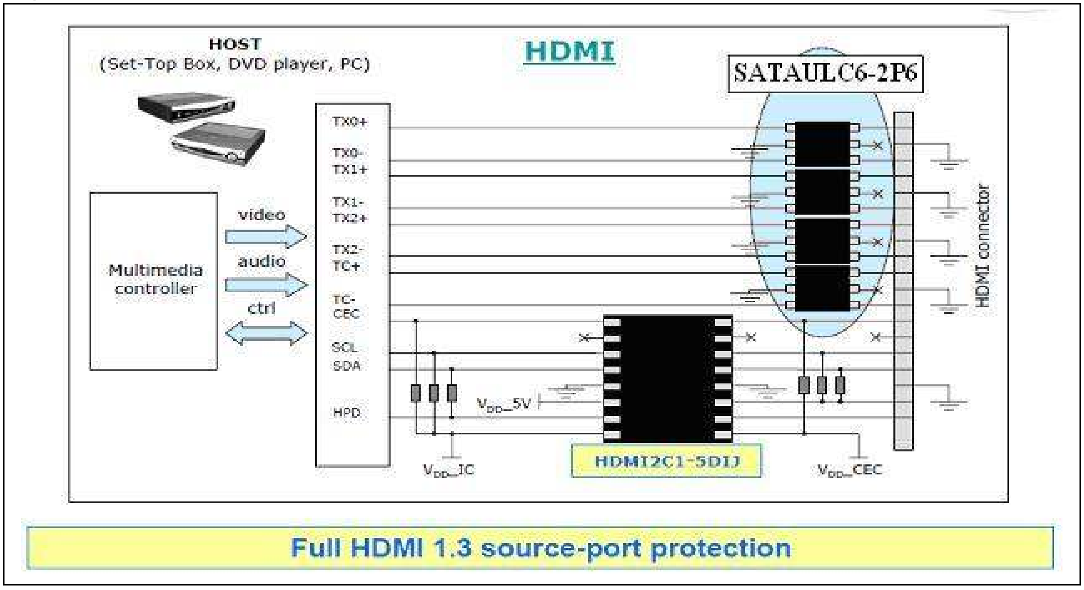


Figure 12. Electrical schematic proposal for STM32100B-EVAL

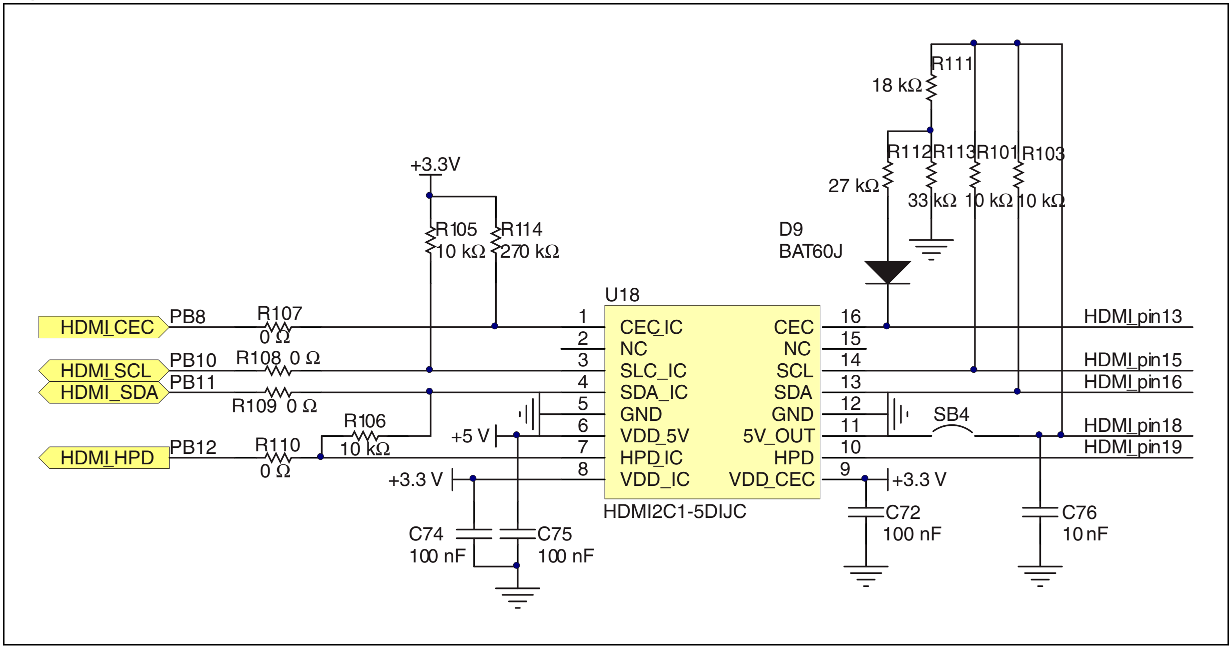
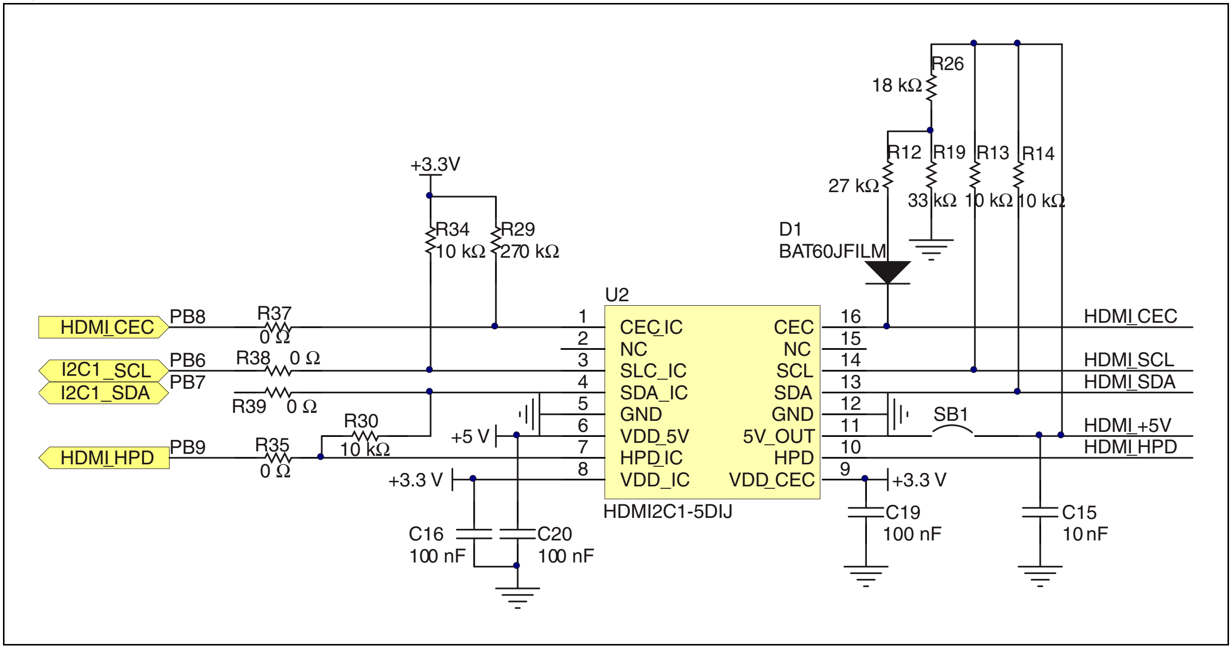


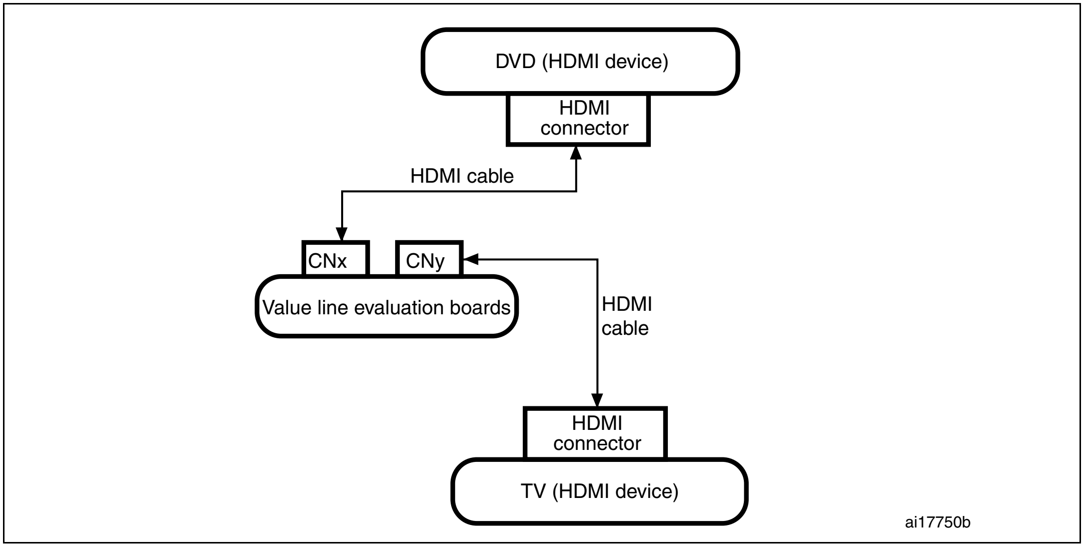
Figure 13. Electrical schematic proposal for STM32100E-EVAL



2.2 Hardware connection

To run the CEC demonstration, you have to connect the HDMI connector of the Value line evaluation board (STM32F100B-EVAL or STM32F100E-EVAL) to another HDMI device (TV, recording device, etc.) using the HDMI cable (see *Figure 10*)

Figure 14. Example of a hardware connection



Note: You can connect the Value line evaluation board to other Value line evaluation boards using HDMI cables, or a simple wire between all CEC device lines (PB8). If you choose the second solution, do not forget to connect the grounds of all the boards together.

Table 3. STM32F100xx and HDMI-CEC connection

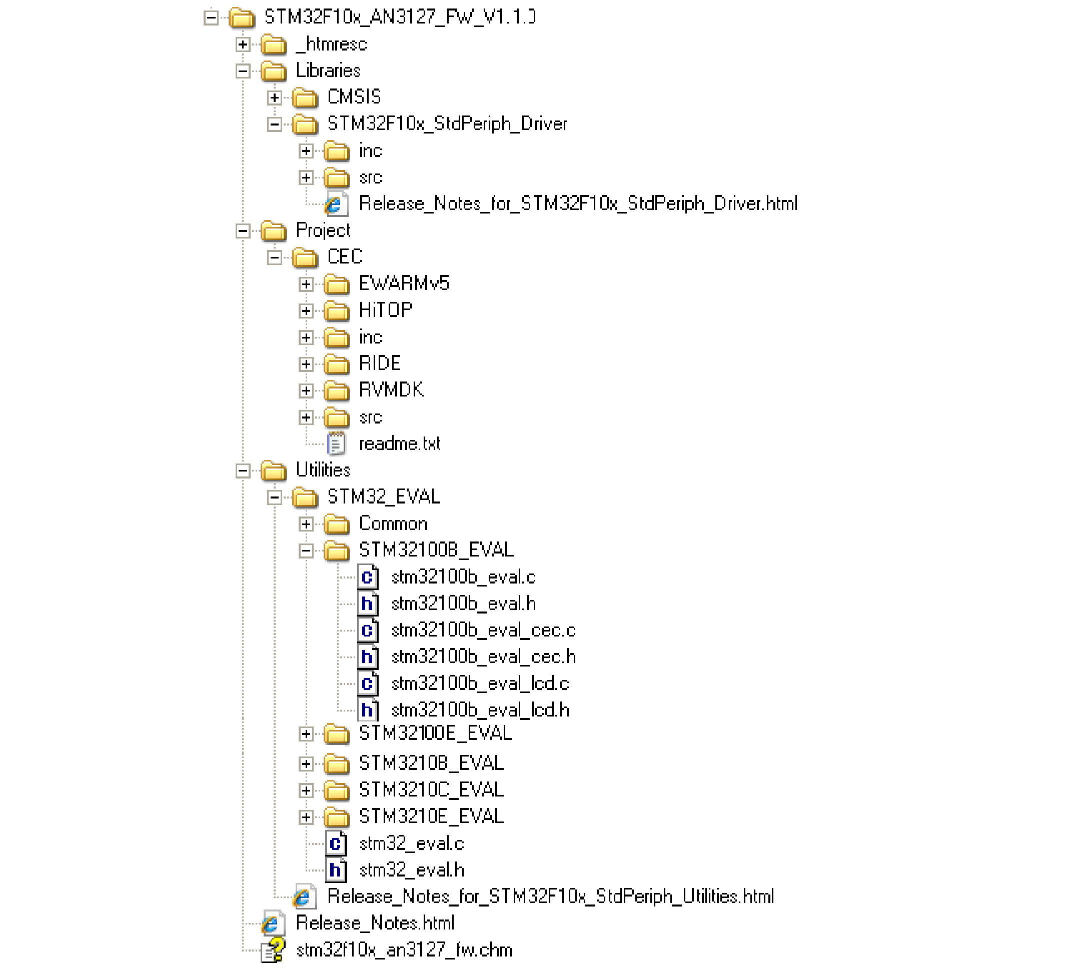
HDMI-CEC pins	STM32100B-EVAL	STM32100E-EVAL	Configuration	Remap
CEC line pin	PB8	PB8	Output open drain	PB10
CEC_I2C_SCL	PB10	PB6	Output open drain	
CEC_I2C_SDA	PB11	PB7	Output open drain	
CEC_HPD_PIN	PB12	PB9	Output push pull	

3 Firmware description

3.1 Package directories

When unzipped, the package has the structure shown in the figure below.

Figure 15. Package directory structure



```
STM32F10x_AN3127_FW_V1.1.0
    _htmresc
    Libraries
        CMSIS
        STM32F10x_StdPeriph_Driver
            inc
            src
            Release_Notes_for_STM32F10x_StdPeriph_Driver.html
    Project
        CEC
            EWARMv5
            HiTOP
            inc
            RIDE
            RVMDK
            src
            readme.txt
    Utilities
        STM32_EVAL
            Common
            STM32100B_EVAL
                stm32100b_eval.c
                stm32100b_eval.h
                stm32100b_eval_cec.c
                stm32100b_eval_cec.h
                stm32100b_eval_lcd.c
                stm32100b_eval_lcd.h
            STM32100E_EVAL
            STM3210B_EVAL
            STM3210C_EVAL
            STM3210E_EVAL
            stm32_eval.c
            stm32_eval.h
        Release_Notes_for_STM32F10x_StdPeriph_Utilities.html
    Release_Notes.html
    stm32f10x_an3127_fw.chm
```

_htmresc folder: contains all the html page resources of the package.

Libraries folder: contains all CMSIS files and the STM32F10x standard peripheral's Drivers:

- ***CMSIS*** subfolder: contains the STM32F10xxx CMSIS files: device peripheral access layer and core peripheral access layer
- ***STM32F10x_StdPeriph_Driver*** subfolder: contains all the subdirectories and files that make up the core of the library:
  - ***inc*** subfolder: contains the header files of the peripheral drivers
  - ***src*** subfolder: contains the source files of the peripheral drivers

Project folder: contains preconfigured projects and the source and header files of the CEC demonstration

- **src** subfolder: contains the source files
  - *main.c*: file in which the system clocks and interrupts are configured
  - *stm32f10x_it.c*: file in which the CEC events (transmitter/receiver) and error (acknowledge failure, bus error, overrun, arbitration loss) interrupts are handled.
  - *cec_display.c*: displays the CEC demonstration messages on the LCD of Value line Evaluation board, or on HyperTerminal
- **inc** subfolder: contains the header files
  - *stm32f10x_it.h*: headers of the interrupt handlers
  - *stm32f10x_conf.h*: configuration file
  - *cec_display.h*: header file of *cec.display.c*
- EWARMv5, RVMDK, HiTOP and RIDE subfolders: contain tool-dependent preconfigured projects and workspaces

Utilities folder: STM32_EVAL implement an abstraction layer that interacts with the human interface resources-buttons, LEDs, LCD and COM ports (USARTs)-available on the STMicroelectronics Value line evaluation boards

The stm32_eval.c driver provides a common API to interact with buttons, LEDs and COM ports, while the definitions of these hardware resources are made in the header file of the Value line evaluation board (*stm32100b_eval.h* or *stm32100e_eval.h*).

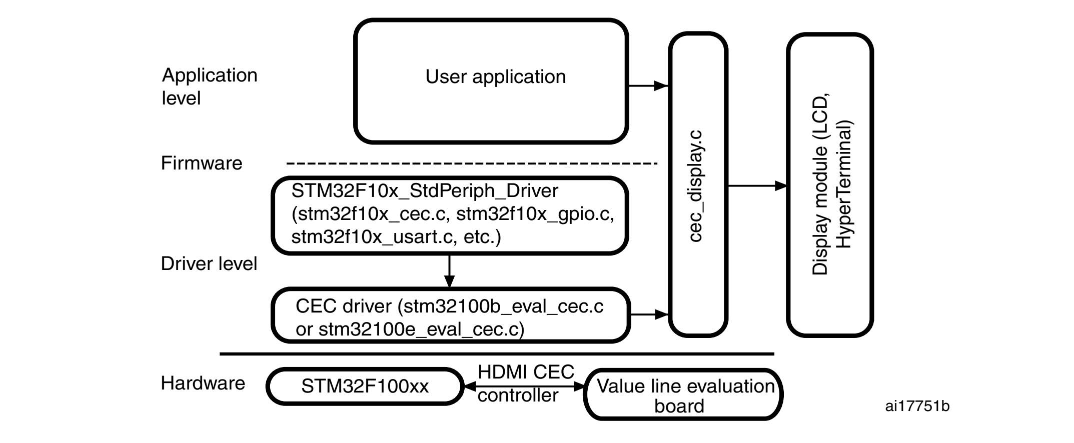
Moreover, a common API is provided to manage the LCD (*stm32100b_eval_lcd.c* or *stm32100e_eval_lcd.c*).

Likewise, a full firmware API is provided to manage the Value line board's HDMI-CEC (*stm32100b_eval_cec.c* or *stm32100e_eval_cec.c*). See *Table 4: High-level functions* for a list of the functions.

3.2 Firmware architecture

Figure 16 shows the firmware architecture, the different blocks and levels are explained below.

Figure 16. Firmware architecture


Application level
User application
Firmware
STM32F10x_StdPeriph_Driver (stm32f10x_cec.c, stm32f10x_gpio.c, stm32f10x_usart.c, etc.)
Driver level
CEC driver (stm32100b_eval_cec.c or stm32100e_eval_cec.c)
cec_display.c
Display module (LCD, HyperTerminal)
Hardware
STM32F100xx
HDMI CEC controller
Value line evaluation board
ai17751b

- The Value line evaluation board (STM32100B-EVAL or STM32100E-EVAL) is designed as a complete development platform for STMicroelectronics' ARM™ Cortex®-M3 core-based Value line microcontroller.
- ***STM32F10x_StdPeriph_Driver***: STM32F10x standard peripheral library
- **User application**: Value line boards (STM32100B-EVAL or STM32100E-EVAL) CEC demonstration
- **Display module**: It can be an LCD or Terminal programs. It is used to display the CEC demonstration messages.
- **CEC driver** (***stm32100b_eval_cec*** or ***stm32100e_eval_cec***): the application may interface CEC devices directly through the driver application layer. The driver functions are summarized in *Table 4: High-level functions*.

High-level functions

These are the functions that can simply be called by the final application to execute all needed configurations and perform high-end functionalities (like sending and receiving messages).

Table 4. High-level functions

Function name	Description
`HDMI_CEC_Init`	Initializes the HDMI-CEC controller: ● Enables the CEC and GPIOx clocks ● Configures the CEC line ● Calls the `PhysicalAddressDiscovery` function ● Calls the `LogicalAddressAllocation` function ● Calls the `HDMI_ReportPhysicalAddress` function
`HDMI_CEC_TransmitMessage`	Transmits messages (Header, opcode and operands)
`HDMI_CEC_GetErrorStatus`	Gets the error status register (ESR) register status
`HDMI_CEC_ProcessIRQSrc`	Processes all the interrupts (RBTF, RERR, TBTRF, TERR) that are high
`PhysicalAddressDiscovery`	Algorithm used to discover the physical address
`LogicalAddressAllocation`	Allocates the device’s logical address based on the logical address allocation algorithm

Table 4. High-level functions (continued)

Function name	Description
`HDMI_ReportPhysicalAddress`	Reports the physical address to all other devices, thus allowing any device to create a map of the network
`HDMI_CEC_CommandCallBack`	Handles the CEC command’s receive callback

4 Value line evaluation board CEC demonstration

The demonstration delivered with this application note is intended to run on the Value line evaluation boards (STM32100B-EVAL or STM32100E-EVAL) and it shows how to configure the HDMI-CEC peripheral and how to create a CEC network providing high-level communication between different devices using CEC protocol messages.

4.1 CEC demonstration overview

The CEC demonstration contains 2 different configurations:

1. Use the joystick push-buttons and display the CEC demonstration messages on the LCD of the Value line evaluation boards (STM32100B-EVAL or STM32100E-EVAL).
2. Use the keyboard keys (r: right, l: left, u: up, d: down and s: select) and display the CEC demonstration messages on HyperTerminal.

Note: To use the second configuration, you have to comment the following defines in the cec_display.c file:

```
#define LCD_DISPLAY
```

```
#define USE_JOYSTICK
```

In the following sections, the SEL, RIGHT, LEFT, UP and DOWN buttons mean the SEL, RIGHT, LEFT, UP and DOWN joystick push-buttons or the “s”, “r”, “l”, “u” and “d” keyboard keys, respectively, depending on the selected configuration.

Figure 17: CEC demonstration flowchart shows the menu system of the CEC demonstration. After system reset, you should use the UP, DOWN, RIGHT, LEFT and SEL buttons to select the device type of the CEC device. Then you should press the SEL button to enter the CEC menu.

The LEFT and RIGHT buttons should be used to navigate between the available follower addresses and CEC commands. The SEL button should be used to select the follower address or command to be sent.

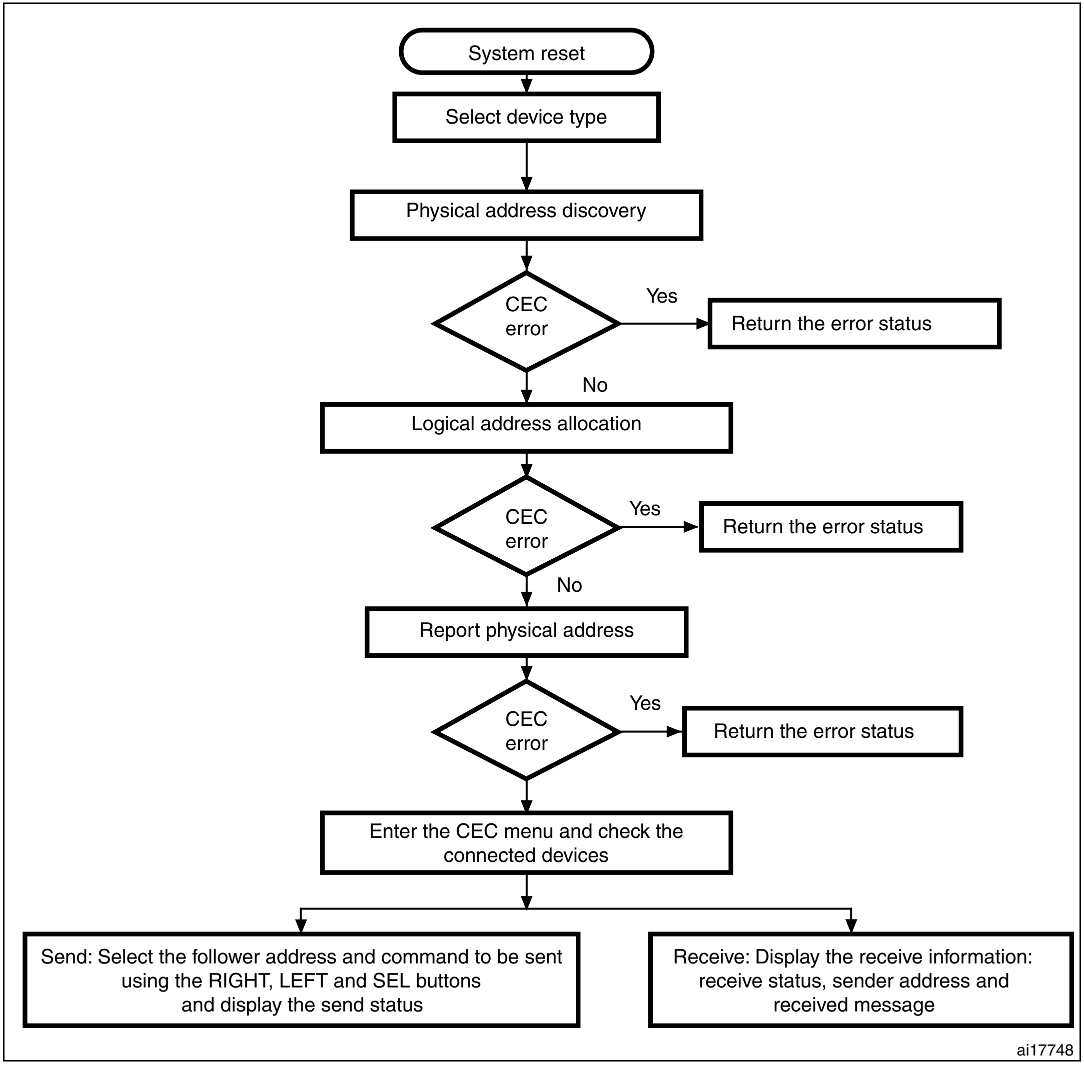
To run the CEC demonstration:

1. Load the CEC demonstration and reset the device
2. Select the device type of the CEC device using the joystick push-buttons or the keyboard keys (see *Section 4.2: Device type selection*)
3. If the selected device type is not available, the following message is displayed: “Unregistered Device”.
   In this case you have to reset the device and select another device type.
4. The physical and logical addresses are displayed on the selected display module (LCD or HyperTerminal).
5. Use the SEL button to enter the CEC menu
6. Use the LEFT and RIGHT buttons to navigate between the follower addresses. Use the SEL button to select the desired follower address (see *Figure 20: CEC menu*)

Note: *Only the addresses of connected devices are available. If a new device is connected, its logical address is added to the list of available follower addresses.*

7. Use the LEFT and RIGHT buttons to navigate between the available CEC commands. Use the SEL button to select the command to be sent (*Figure 24: Select CEC command*)
8. The send status is displayed on the selected display module. Repeat from step 6 if you want to select another follower and send another command. Reset the device if you want to select a different device type.

Figure 17. CEC demonstration flowchart


System reset
Select device type
Physical address discovery
CEC error
Yes
Return the error status
No
Logical address allocation
CEC error
Yes
Return the error status
No
Report physical address
CEC error
Yes
Return the error status
Enter the CEC menu and check the connected devices
Send: Select the follower address and command to be sent using the RIGHT, LEFT and SEL buttons and display the send status
Receive: Display the receive information: receive status, sender address and received message
ai17748

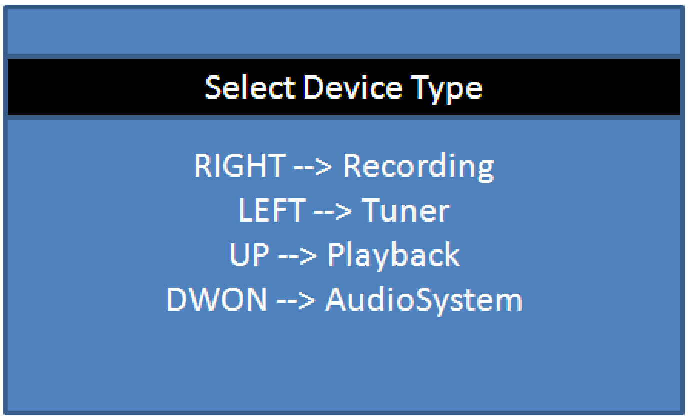
4.2 Device type selection

To select the device type of the CEC device, do the following:

- RIGHT button: select *recording* as the device type
- LEFT button: select *tuner* as the device type
- UP button: select *playback* as the device type
- DOWN button: select *audio system* as the device type

The selected device type is then used in the logical address allocation algorithm to allocate the logical address of the CEC device.

Figure 18. Device type selection



Select Device Type

RIGHT --> Recording
LEFT --> Tuner
UP --> Playback
DWON --> AudioSystem

4.3 Physical address discovery

The algorithm defined in *Section 1.4.5: Discovery algorithm* is used to allocate the physical address of each device. If no CEC error is generated, the CEC device executes the logical address allocation algorithm.

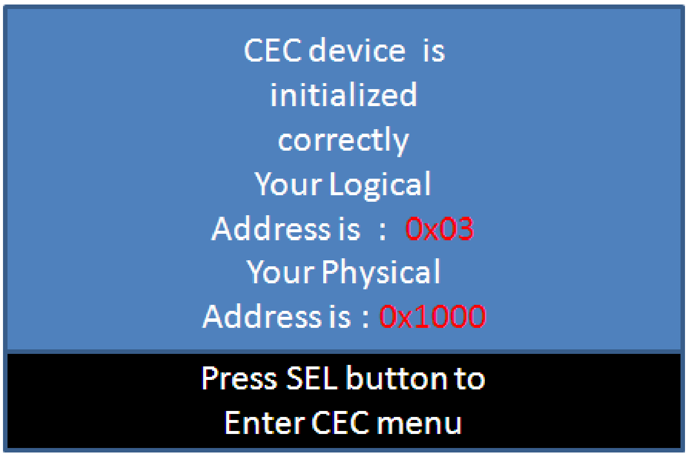
4.4 Logical address allocation

The selected device type is used to allocate the logical address as described in *Section 1.4.7: Logical address allocation*. If no CEC error is generated, the physical and logical addresses are displayed on the LCD. You should then press the SEL push-button to enter the CEC menu.

Note: 1 If the selected device type is not available, the following message is displayed: "Unregistered Device".
In this case you have to reset the device and select another device type.

2 Whenever a new physical address is discovered, the CEC device reports the association between its logical and physical addresses by broadcasting <Report Physical Address>.

Figure 19. Physical and logical addresses display



CEC device is
initialized
correctly
Your Logical
Address is : 0x03
Your Physical
Address is : 0x1000

Press SEL button to
Enter CEC menu

4.5 Checking the connected devices

Before displaying the available follower addresses, the CEC device checks the connected devices by sending a <Polling Message> to all logical addresses. If the <Polling Message> is acknowledged, then the device retains that address. After that step, only the addresses of the connected devices are displayed.

If no device is connected, only the broadcast address is available.

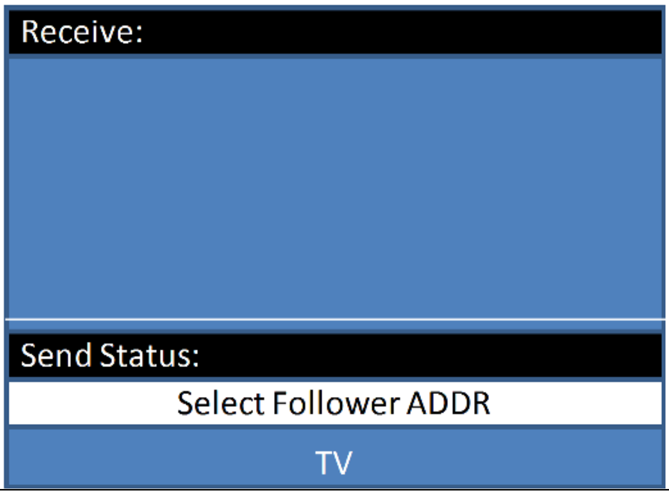
If a new device is connected after the connected device check step, its address is automatically added to the list of available addresses.

4.6 Displaying CEC send/receive information on the LCD

The LCD screen is divided into two parts as shown in *Figure 20: CEC menu*:

- a subscreen that shows the CEC receive information: receive status, sender address
- a subscreen used to select the follower address and the command to send

Figure 20. CEC menu



Receive:

Send Status:

Select Follower ADDR

TV

4.6.1 Receive information subscreen

The following information is displayed when a new message is received:

- Receive status
- Sender address
- Number of bytes (including the sender address)
- Message opcode
- Data (operands)

Figure 21. Receive flowchart

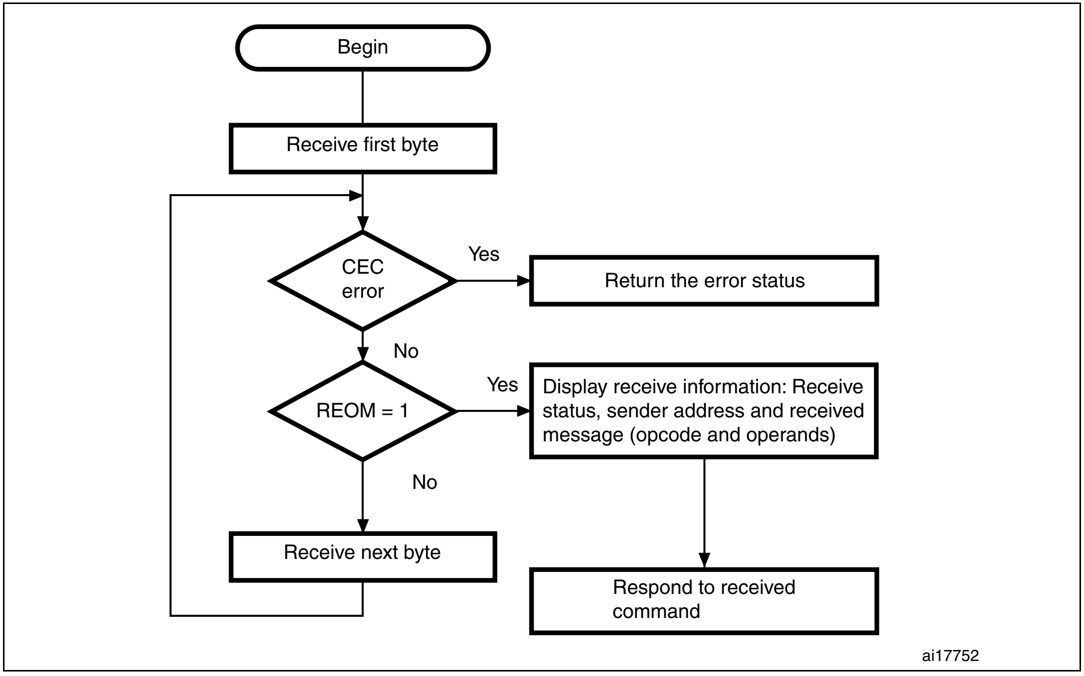
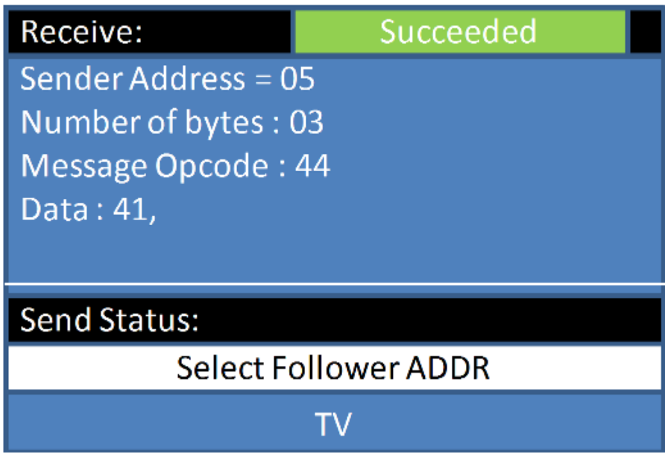

Begin
Receive first byte
CEC error
Yes
Return the error status
No
REOM = 1
Yes
Display receive information: Receive status, sender address and received message (opcode and operands)
No
Receive next byte
Respond to received command
ai17752

Figure 22: Receive information subscreen shows that the device has correctly received the frame from the sender with address 0x5.

- Sender address 0x5
- The number of bytes received is 0x3 (header + opcode + data)
- Message opcode: 0x44
- Data: 0x41

Figure 22. Receive information subscreen



Receive: Succeeded

Sender Address = 05

Number of bytes : 03

Message Opcode : 44

Data : 41,

Send Status:

Select Follower ADDR

TV

Note: The STM32F100xx Value line CEC responds only to the following commands (to other commands, it sends a feature abort):

- Standby
- Get CEC version
- Give physical address
- Give OSD name

4.6.2 Send information subscreen

This subscreen is used to navigate among the connected device addresses by using the RIGHT and LEFT buttons. After selecting the follower address, you have to select the command to be sent to the selected follower address with the LEFT, RIGHT and SEL buttons. Once you have selected a command, the CEC device sends this command to the selected follower address, and displays the transmission status.

You can then select a new follower address and a new command.

Figure 23. Send flowchart

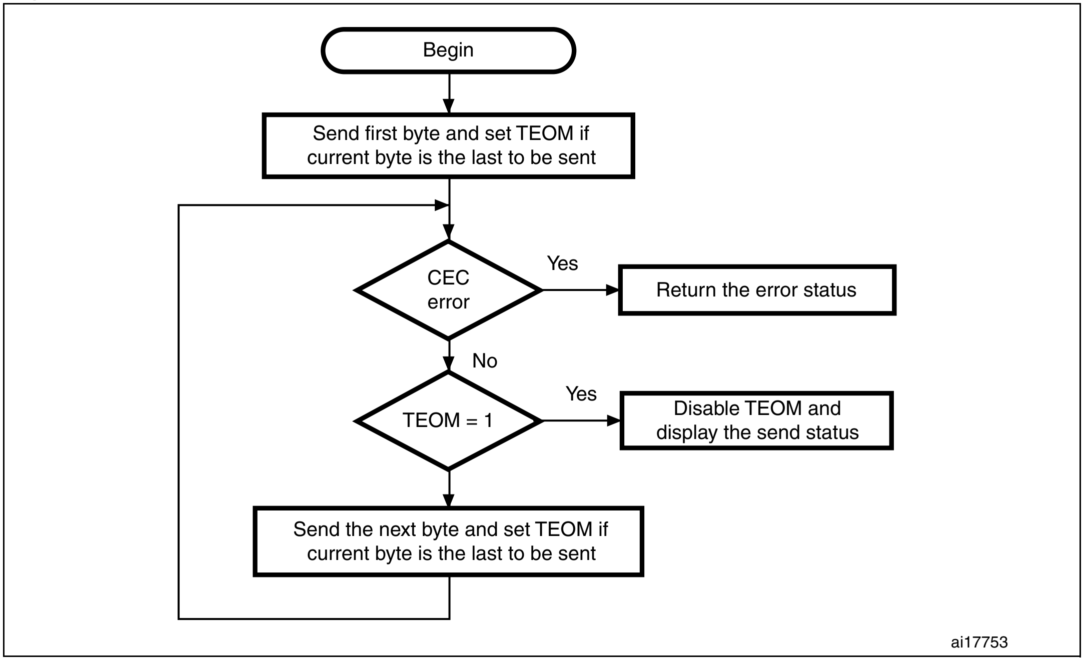
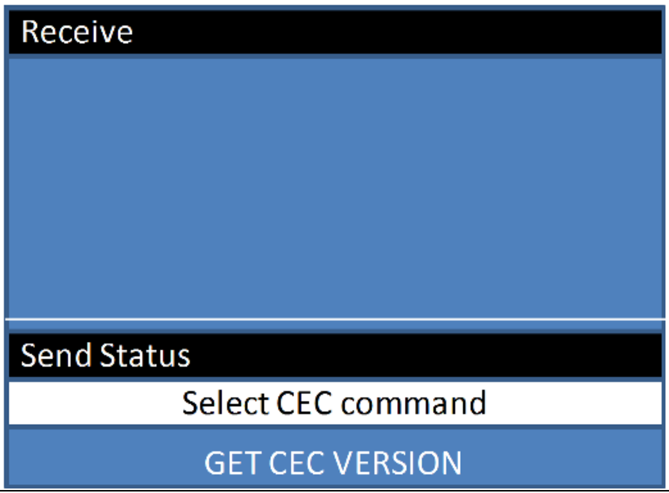

Begin
Send first byte and set TEOM if current byte is the last to be sent
CEC error
Yes
Return the error status
No
TEOM = 1
Yes
Disable TEOM and display the send status
Send the next byte and set TEOM if current byte is the last to be sent
ai17753

Figure 24. Select CEC command


Receive

Send Status

Select CEC command

GET CEC VERSION

5 Conclusion

This application note gives details on the HDMI-CEC controller available on the Value line evaluation boards (STM32100B-EVAL or STM32100E-EVAL) and the CEC demonstration firmware running on this evaluation board.

Note: The HDMI-CEC communication can be enhanced by adding Infrared protocols so that the user commands can be generated in a more flexible way. The STM32 Value line devices can easily handle this configuration using the embedded CEC and timer peripherals (the STM32 timer is used to decode the infrared protocols with the associated software). This combination (CEC communication / Infrared protocols) is implemented on STM32100E-EVAL firmware demo.

6 Revision history

Table 5. Document revision history

Date	Revision	Changes
01-Mar-2010	1	Initial release.
11-Oct-2010	2	Updated for high-density value line devices.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com