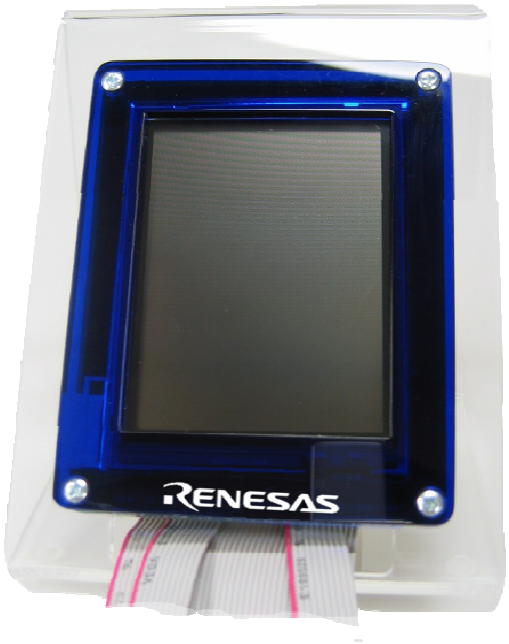
Renesas QVGA portrait TFT Display Board

18bit color, 8.9cm (3.5”) with touch screen

“Add-On QVGA Board“

(EB-EXT-QVGA-PT)

Quick Start-up Guide



Renesas Electronics (Europe)

Automotive Business Unit – Instrumentation

Release Date: 2011-02-03

Version: V1.1

Content:

1 Introduction

This document is a quick start-up guide for the “Renesas QVGA portrait TFT Display Board”, which is nick-named “Add-On QVGA Board” and has the Renesas order code EB-EXT-QVGA-PT.

It is recommended to study the NEC NL2432HC22-41K TFT module datasheet (document number: DOD-PP-0307 2nd edition) for further details.

Download the documents from the **Renesas Electronics (Europe)** web site.

URL: http://www2.renesas.eu/docuweb/

Search for keyword: ‘*EB-EXT-QVGA-PT*’

The board is designed to be used with our Dx4 (e.g. uPD70F3532, uPD703538) series dashboard controller evaluation and application boards.

1.1 Purpose of the board

- Extend application boards with touch-controlled TFT
- Simplify graphics project development
- Enable future usage with further evaluation boards

1.2 On-board devices

- 3.5" TFT-LCD with touch screen module: NL2432HC22-41K
- Power supply
    - Step-Up Converter for backlight
    - Linear regulator for display logic

2 Board operation

2.1 Overview

The Add-On QVGA Board consists of a NL2432HC22-41K TFT module and adapts its ZIF connector to three legacy standard header connectors on the backside. This connectors are designed to fit directly to our Dx4 (e.g. uPD70F3532, uPD703538) series dashboard controller evaluation and application boards. Fitting cables for all the three legacy connectors are part of the delivery package of this board.

2.1.1 Front view

The following picture gives an overview of the front view of the Add-On QVGA Board. The area directly over the active display area is not covered by the transparent plastic housing, to allow usage of the resistive touch screen.

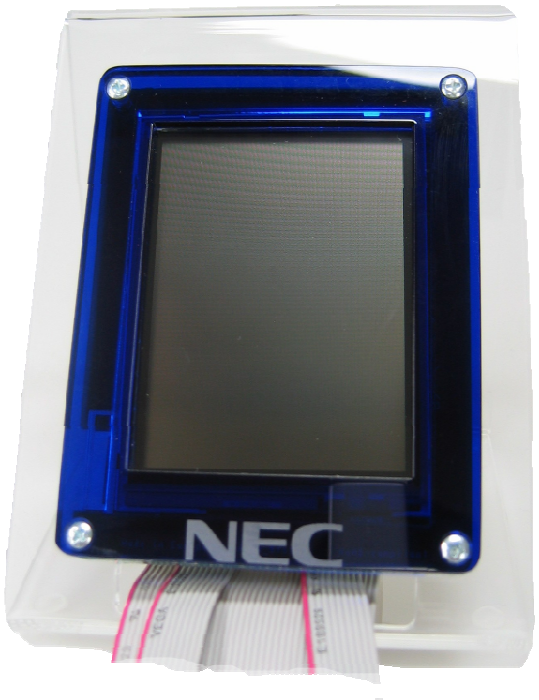


Figure 2-1 Add-On QVGA board front view

2.1.2 Rear view - Connectors and other elements

The following picture gives an overview of three connectors on the rear side of the Add-On QVGA Board;

- Video Input Connector
- CSI Control Connector
- Touch Screen Connector

For details on the connectors please refer to chapter 3.3 Connectors.

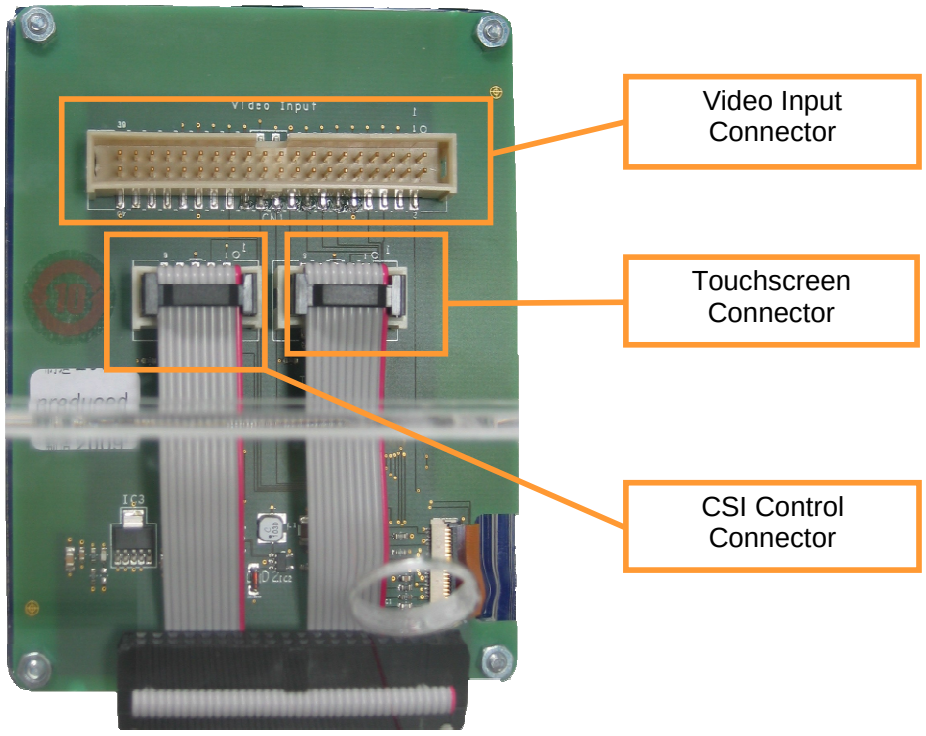


Figure 2-2 Add-On QVGA board rear view - connectors and other elements

2.1.3 Accessories

The following three cables are delivered together with the Add-On QVGA board as accessories.

- Video input cable

The video input cable is a 1:1 cable with three standard 100 mil connectors. Two of the connectors are 40-pin types, one is 30-pin. The cable is used to connect either a legacy 40-pin video connector (like on Dx4 application boards) or a legacy 30-pin video connector (like on RAPP board) to the 40-pin video input connector of the Add-On QVGA board.

The following picture shows the external connection side of the video input cable.

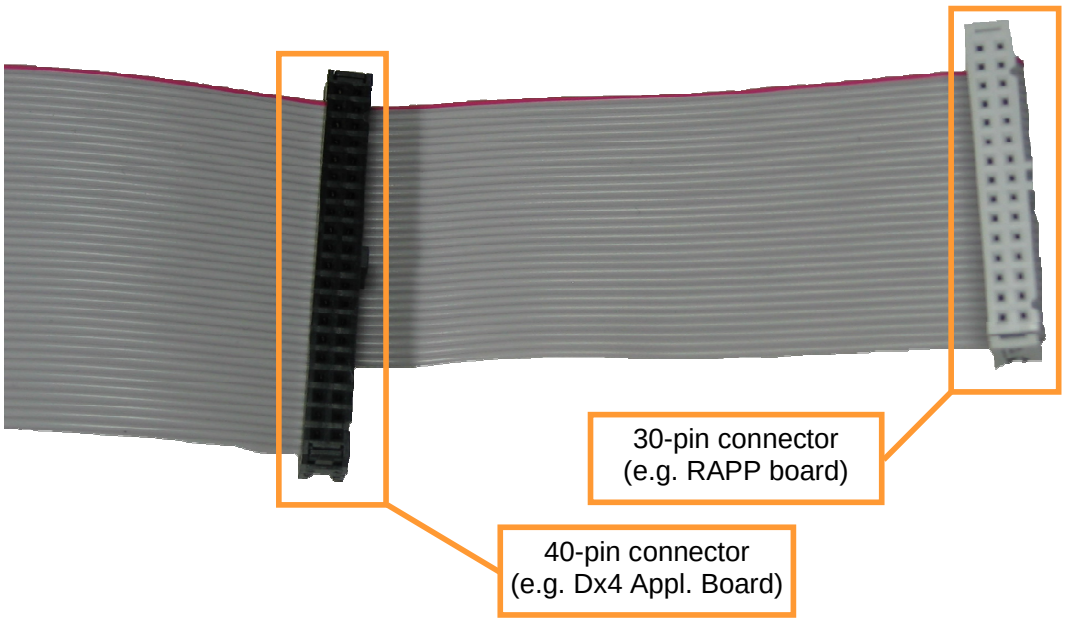


Figure 2-3 Video input cable

- CSI control cable

The CSI control cable is a 1:1 cable with two standard 100 mil connectors with 10-pins each. The cable is used to connect a CSI (SPI) of a microcontroller to the CSI control connector.

- Touch screen cable

The Touch screen cable is a 1:1 cable with two standard 100 mil connectors with 10-pins each. The cable is used to connect GPIO and ADC port pins of a microcontroller to the Touch screen connector.

The following picture shows the external connection side of the CSI control cable. Because in fact the Touch screen cable is identical to the CSI control cable, the picture shows the external connection side of the Touch screen cable too.

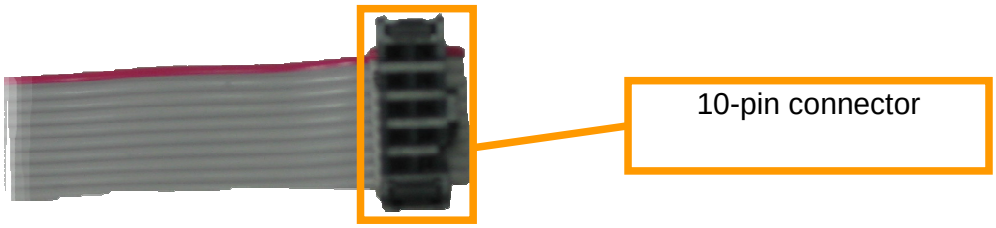


Figure 2-4 CSI control cable / Touch screen cable

2.2 Quick board startup sequence

Here is a recommendation how to start the board operation:

- Connect the video input connector CN1 to a video output (e.g. MVO)
- Connect the CSI control connector CN2 to a CSI (SPI). This is mandatory for board operation.

Connect the Touch Screen connector CN3 to GPIO and ADC[(1)].

- Power up the board you connected the Add-On QVGA board to.
- Initialize the NL2432HC22-41K TFT module by sending its CSI initialization sequence. Please refer to chapter 5.1 Initialization sequence for more information.
- Activate the video output (e.g. MVO). Please refer to chapter 5.2 Video output activation for more information.
- Select touch screen scan direction X or Y by GPIO and read out the touch screen position information by ADC[(1)].

[(1)]This is optional, in case the touch screen should be used.

3 Board specification

3.1 Power Supply

The display logic, display backlight as well as the touch screen is supplied directly out of the 3.3V that are applied via the video input connector. The Add-On QVGA Board drains approximately 150 mA out of this 3.3V rail. The board dissipates a power of about 500 mW.

Please note that the 3.3V power inputs (VDD) on all the three input connectors (CN1 video input, CN2 CSI control and CN3 touch control) are interconnected. You must not supply different voltages or voltages from different supply rails through this 3.3V inputs.

3.2 Interfaces

3.2.1 Parallel Video Input Interface

Standard parallel video interfaces with 3.3V (LVTTL) signal levels can be connected to the video input connector (CN1) of the Add-On QVGA Board. The board requires the connection of a parallel video interface with

- 18 color data signals
- Horizontal sync signal
- Vertical sync signal
- Pixel clock

Such kind of parallel video interface is provided for instance by our graphic application board of Dx4-H dashboard controllers.

3.2.2 CSI Control Interface

A CSI (SPI) interface with 3.3V (LVTTL) signal levels needs to be connected to the CSI Control Connector (CN2) for initialization of the Add-On QVGA Board. The board requires the connection of a CSI (SPI) with the following signals

- Serial clock
- Serial data out
- /Chip-select
- /Reset

3.2.3 Touch Screen Interface

A legacy touch screen interface is available on the Touch Screen Connector (CN3). It is used to read out position data of the resistive touch screen. A microcontroller needs to select the touch screen scan direction using GPIO port pins, while it reads the direction dependent position information that is sampled by its connected the ADC channels. For this the following signals can be connected

- Select X scan direction input 3.3V (LVTTL) level
- Select Y scan direction input 3.3V (LVTTL) level
- X position information analog output (0 - 3.3V)
- Y position information analog output (0 – 3.3V)

The usage of the Touch Screen Interface is not mandatory.

Remark: To allow detection of the untouched state of the Touch Screen, 22kOhm pull-down resistors are placed in parallel with capacitors C4 and C6 since Display revision V1.10. This ensures a defined voltage level of the analog X and Y position outputs in this case.

3.3 Connectors

3.3.1 Video Input (legacy 40-pin header)

This connector allows connecting a parallel video data interface like the video outputs of Dx4 (i.e. MVO and SVO).

Video input **CN1**					
Standard header, male 2x20					
Pin	**Name**	**Board connection**	**Pin**	**Name**	**Board connection**
1	R2	VO0R2	2	B2	VO0B2
3	VDD33	VDD33	4	N.C.	Not connected
5	B3	VO0B3	6	B4	VO0B4
7	B5	VO0B5	8	B6	VO0B6
9	B7	VO0B7	10	GND	GND
11	G2	VO0G2	12	G3	VO0G3
13	G4	VO0G4	14	G5	VO0G5
15	G6	VO0G6	16	G7	VO0G7
17	GND	GND	18	R3	VO0R3
19	R4	VO0R4	20	R5	VO0R5
21	R6	VO0R6	22	R7	VO0R7
23	VDD33	VDD33	24	N.C.	Not connected
25	HSYNC	VO0HSYNC	26	VSYNC	VO0VSYNC
27	CLK	VO0CLK	28	EN	VO0EN
29	GND	GND	30	GND	GND
31	VDD33	VDD33	32	R0	VO0R0
33	R1	VO0R1	34	G0	VO0G0
35	G1	VO0G1	36	B0	VO0B0
37	B1	VO0B1	38	N.C.	Not connected
39	N.C.	Not connected	40	GND	GND

3.3.2 CSI Control (legacy 10-pin header)

This connector allows connecting a legacy CSI (SPI) interface connector. This connector is used for the mandatory initialization of the NL2432HC22-41K TFT display.

CSI control **CN2**					
Standard header, male 2x5					
Pin	**Name**	**Board connection**	**Pin**	**Name**	**Board connection**
1	CSI_SO	VO0SO	2	GND	GND
3	CSI_SCK	VO0SCK	4	GND	GND
5	VDD33	VDD33	6	LIGHT_ENAB	ON_LM2735
7	CSI_CSZ	VO0CSZ	8	GND	GND
9	CSI_RESZ	RESETZ0	10	GND	GND

3.3.3 Touch Screen (legacy 10-pin header)

This connector allows connect to the resistive touch screen build into the NL2432HC22-41K TFT display.

Touch Screen **CN3**					
Standard header, male 2x5					
Pin	**Name**	**Board connection**	**Pin**	**Name**	**Board connection**
1	X_OUT	TOUCH_XL_IN	2	GND	GND
3	Y_OUT	TOUCH_YU_IN	4	GND	GND
5	VDD33	VDD33	6	N.C.	Not connected
7	X_ACTZ	TOUCH_ACTX_Z	8	GND	GND
9	Y_ACTZ	TOUCH_ACTY_Z	10	GND	GND

4 Board schematics

The Renesas QVGA portrait TFT Display schematics is a rather simple design, because it mainly adapts the video-, touch- and control signals from the three standard header connectors to the zero insertion force connector of the used NL2432HC22-41K display.

The left side of the schematics show the three standard header connectors used to interface the display module by external graphics hardware. (e.g. Dx4 Application board) On the right upper corner the zero insertion force connector for the NL2432HC22-41K display can be found. In the right lower corner the power supply area is located.

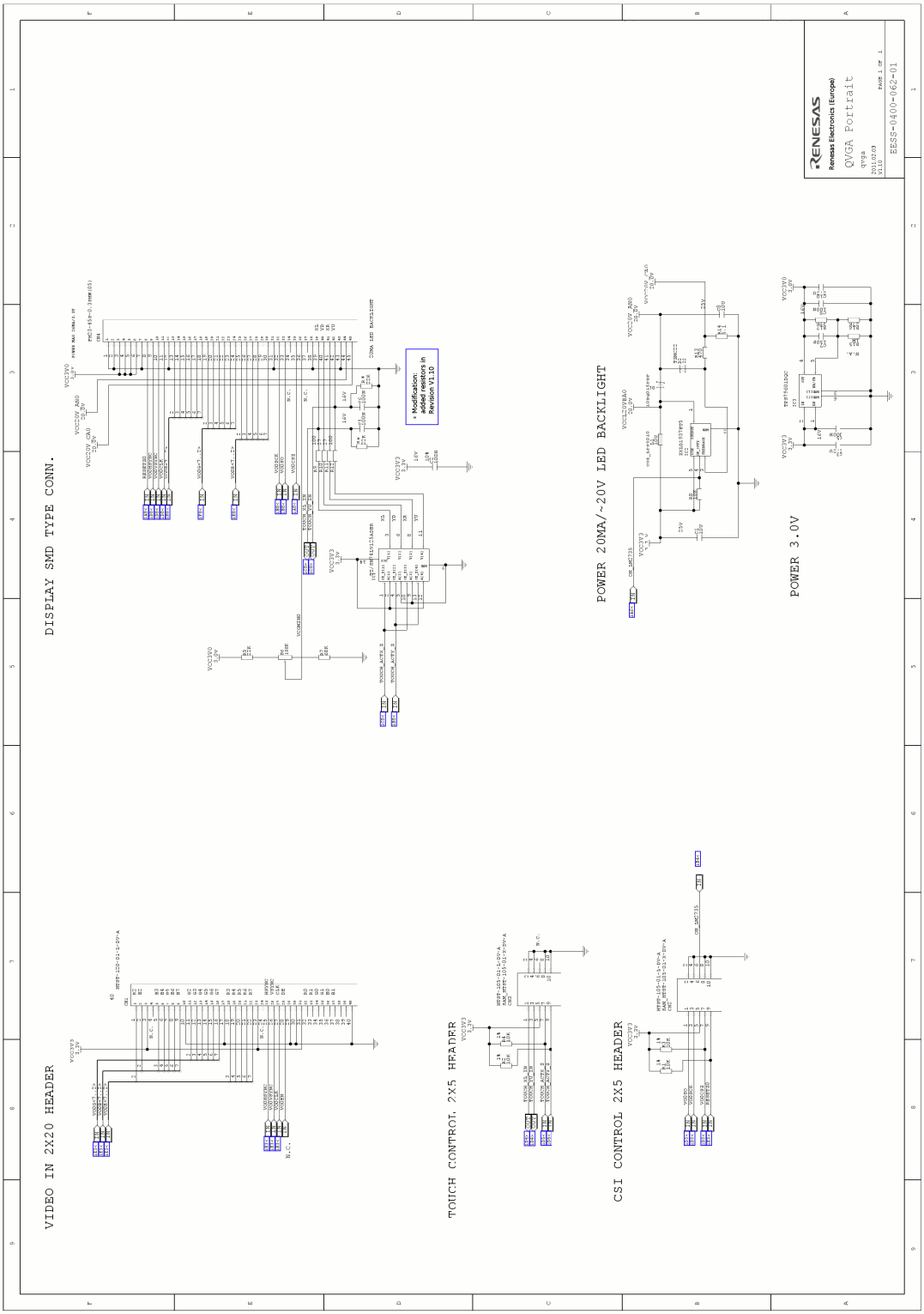


Figure 4-1 Add-On QVGA Board Schematics

5 Software

5.1 Initialization sequence

The init sequence is described in detail in the NL2432HC22-41K TFT module datasheet. The following shows a working source code for the CSI (SPI) initialization of the Add-On QVGA board on a Dx4-H microcontroller device. The function Csi_NecDisplay_Init() is located in the file 'csi_display_init.c' and is called before the parallel video output interface is activated.

```
/* csi_display_init.c
   Init NEC display NL2432HC22-41K via CSI sequence.
*/


/*********************************************************
  Section: Includes
*/
#include "typedef.h"
#include "portc_api.h"
#include "pclk_api.h"
#include "tick_api.h"
#include "csig_api.h"

/*********************************************************
  Section: Local Defines
*/

/*  CSI pins used for the displays
    display n on Dx4 eval Piggy:
    VOnSCK  - P0_15       CNTP1_18
    VOnSO   - P0_14       CNTP1_17
    VOnCS   - P1_4        CNTP1_23 (emulated by GPIO)
    VOnRst  - P1_0        CNTP1_19 (emulated by GPIO)
*/

#define CSIG0_CLOCK_SC_PORT         0
#define CSIG0_CLOCK_SC_PIN          15
#define CSIG0_DOUT_SO_PORT          0
#define CSIG0_DOUT_SO_PIN           14
#define CSIG0_GPIO_RESETZ_PORT      1
#define CSIG0_GPIO_RESETZ_PIN       0
#define CSIG0_GPIO_CHIPSELZ_PORT    1
#define CSIG0_GPIO_CHIPSELZ_PIN     4


/*********************************************************
  Section: Local Functions
*/

/*********************************************************
```

```
  Function: Csi_NecDisplay_Init
  Init NEC display NL2432HC22-41K via CSI.

  Parameters:
  void

  Returns:
  void
*/
void Csi_NecDisplay_Init(void) {
    /* Initialization of the NEC display
       NL2432HC22-41K via CSI needs to be done. */

    eeCsiG_InitCfg_t    mcfg;
    uint32_t tmp;

    eePortC_PinAssign_t params;
    eePortC_SetOutputBuffer_t out_buffer;

    /* power on sequence for NL2432HC22-41K display */
        /* write reg, register number, write data, data value */
    static const uint8_t SendDispOn[] = {
                0x00,   3, 0x01, 0x01,
                0x00,   1, 0x01, 0x00,
                0x00, 100, 0x01, 0x0F,
                   /* example: write data 0x0F to register R100 */
                0x00, 101, 0x01, 0x3F,
                0x00, 102, 0x01, 0x3F,
                0x00, 103, 0x01, 0x00,
                0x00, 104, 0x01, 0x00,
                0x00, 105, 0x01, 0x30,
                0x00, 106, 0x01, 0x04,
                0x00, 107, 0x01, 0x37,
                0x00, 108, 0x01, 0x17,
                0x00, 109, 0x01, 0x00,
                0x00, 110, 0x01, 0x40,
                0x00, 111, 0x01, 0x30,
                0x00, 112, 0x01, 0x04,
                0x00, 113, 0x01, 0x37,
                0x00, 114, 0x01, 0x17,
                0x00, 115, 0x01, 0x00,
                0x00, 116, 0x01, 0x40,
                0x00,   2, 0x01, 0x40,
                0x00,  75, 0x01, 0x04,
                0x00,  76, 0x01, 0x01,
                0x00,  77, 0x01, 0x01,
                0x00,  80, 0x01, 0x00,
                0x00,  81, 0x01, 0x00,
                0x00,  82, 0x01, 0x2E,
                0x00,  83, 0x01, 0xC4,
                0x00,  86, 0x01, 0x15,
                0x00,  87, 0x01, 0xED,
                0x00,  95, 0x01, 0x3F,
                0x00,  96, 0x01, 0x22,
                0x00,  25, 0x01, 0x76,
                0x00,  26, 0x01, 0x54,
                0x00,  27, 0x01, 0x67,
                0x00,  28, 0x01, 0x60,
                0x00,  29, 0x01, 0x04,
                0x00,  30, 0x01, 0x1C,
                0x00,  31, 0x01, 0xA9,
```

```
            0x00,  32, 0x01, 0x00,
            0x00,  33, 0x01, 0x20,
            0x00,  24, 0x01, 0x77,
            0x00,  59, 0x01, 0x01,
            0x00,   0, 0x01, 0x00};

/* power off sequence for NL2432HC22-41K display */
          /* write, register number, write, data value */
static const uint8_t SendDispOff[] = {
            0x00,   0, 0x01, 0x08,
            0x00,  24, 0x01, 0x00,
            0x00,   1, 0x01, 0x08};

/* Assign Pins */
/*  CSIG0SC (clock)
    VOnSCK on port CSIG0_CLOCK_SC_PORT.CSIG0_CLOCK_SC_PIN */
params.Port       = CSIG0_CLOCK_SC_PORT;
params.Pin        = CSIG0_CLOCK_SC_PIN;
params.Funct      = "CSIG0SC";
params.DirectCtrl = EE_PORTC_DIRECT_CONTROL_OFF;
params.Direction  = EE_PORTC_OUT;
eePortC_AssignPin(&params);

/*  CSIG0SO (data out)
    VOnSO on port CSIG0_DOUT_SO_PORT.CSIG0_DOUT_SO_PIN */
params.Port       = CSIG0_DOUT_SO_PORT;
params.Pin        = CSIG0_DOUT_SO_PIN;
params.Funct      = "CSIG0SO";
params.DirectCtrl = EE_PORTC_DIRECT_CONTROL_OFF;
params.Direction  = EE_PORTC_OUT;
eePortC_AssignPin(&params);

/*  set port pin for reset for CSIG via GPIO
    first make sure the pin is set to inactive-level
    VOnRst on port CSIG0_GPIO_RESETZ_PORT.CSIG0_GPIO_RESETZ_PIN */
eePortC_SetPin(CSIG0_GPIO_RESETZ_PORT, CSIG0_GPIO_RESETZ_PIN);
out_buffer.Pin         = CSIG0_GPIO_RESETZ_PIN;
out_buffer.BuffType    = EE_PORTC_BUFF_STANDARD;
out_buffer.Strength    = EE_PORTC_STRENGTH_STRONG;
out_buffer.Limit       = EE_PORTC_LIMIT_HIGHER ;
out_buffer.OpenDrain   = EE_PORTC_OPENDRAIN_OFF;
out_buffer.Diagnostic  = EE_PORTC_DIAGNOSTIC_OFF;
eePortC_SetOutputBuffer(CSIG0_GPIO_RESETZ_PORT, out_buffer);

/*  set port pin for chip select for CSIG via GPIO
    first make sure the pin is set to inactive-level
    VOnCS on port CSIG0_GPIO_CHIPSELZ_PORT.CSIG0_GPIO_CHIPSELZ_PIN*/
eePortC_SetPin(CSIG0_GPIO_CHIPSELZ_PORT, CSIG0_GPIO_CHIPSELZ_PIN);
out_buffer.Pin         = CSIG0_GPIO_CHIPSELZ_PIN;
out_buffer.BuffType    = EE_PORTC_BUFF_STANDARD;
out_buffer.Strength    = EE_PORTC_STRENGTH_STRONG;
out_buffer.Limit       = EE_PORTC_LIMIT_HIGHER ;
out_buffer.OpenDrain   = EE_PORTC_OPENDRAIN_OFF;
out_buffer.Diagnostic  = EE_PORTC_DIAGNOSTIC_OFF;
eePortC_SetOutputBuffer(CSIG0_GPIO_CHIPSELZ_PORT, out_buffer);
/* End assign Pins */

/* Init CSIG0 */
/*  Set master config (mcfg) */
/* divider 16 = 500k baud */
mcfg.Baudrate   = eePclk_GetActClkFreq(EE_PCLK_CSIG0) / 16;
```

```
    /* eeBsp_Printf("\nBaud rate is: %d \n", mcfg.Baudrate); */
    mcfg.Direction  = EE_CSIG_MSB;              /* MSB first */
    mcfg.ClockPhase = EE_CSIG_CLK_ACTIVE_HIGH;
    mcfg.DataPhase  = EE_CSIG_DATA_PHASE_RISE;

    mcfg.DataLength = 8;                        /* 1byte = 8bit */
    mcfg.Mode  = EE_CSIG_MASTER;                /* master mode */
    eeCsiG_Init(0, &mcfg, (eeCsiG_OptFeat_t*)0 );
    /* eeCsiG_SetIsrCallback(0, EE_CSIG_INT_TIRE, _unusedCsiGIsr);
       eeCsiG_SetIsrCallback(0, EE_CSIG_INT_TIR,  _unusedCsiGIsr);
       eeCsiG_SetIsrCallback(0, EE_CSIG_INT_TIC,  _unusedCsiGIsr); */
    eeCsiG_Enable(0, EE_CSIG_TX_ONLY);          /* only TX direction */

    /* reset display
        assert reset, wait 2ms
        reset, wait 2ms after reset deassertion
    */
    eePortC_ClearPin(CSIG0_GPIO_RESETZ_PORT, CSIG0_GPIO_RESETZ_PIN);
    eeTick_WaitMS(0, 2);
    eePortC_SetPin(CSIG0_GPIO_RESETZ_PORT, CSIG0_GPIO_RESETZ_PIN);
    eeTick_WaitMS(0, 2);

    /* activate (low active) chip select */
    eePortC_ClearPin(CSIG0_GPIO_CHIPSELZ_PORT, CSIG0_GPIO_CHIPSELZ_PIN);
    eeTick_WaitMS(0, 2);

    /* transmit display de-init sequence via CSI */
    /* Note: the de-init is important, because without deinit first,
       the display stays off after Reset without power off */
    for (tmp = 0; tmp < sizeof(SendDispOff); tmp++) {
        if ((tmp % 4) == 0) {
            eeTick_WaitMS(0, 25);
        }
        eeCsiG_Write(0, (void*)(&SendDispOff[tmp]), 1, TRUE);
    }

    /* transmit display init sequence via CSI */
    for (tmp = 0; tmp < sizeof(SendDispOn); tmp++) {
        if ((tmp == sizeof(SendDispOn)-4)
              || (tmp == sizeof(SendDispOn)-8)) {
            eeTick_WaitMS(0, 20);
        }
        eeCsiG_Write(0, (void*)(&SendDispOn[tmp]), 1, TRUE);
    }

    /* de-activate (low active) chip select */
    eeTick_WaitMS(0, 2);
    eePortC_SetPin(CSIG0_GPIO_CHIPSELZ_PORT, CSIG0_GPIO_CHIPSELZ_PIN);
}
```

5.2 Video output activation

The following shows an example source code for the activation of the Single Layer Video Output (SVO) of a DP4-H microcontroller with the setting for an attached Add-On QVGA board. The function VO_Activate() is located in the file ‘vo_activate.c’. It calls the Csi_NecDisplay_Init() function to ensure proper CSI (SPI) initialization of the Add-On QVGA board and selects the right display settings from the display data base (DDB). As shown the "NEC_QVGA_TOUCH_240x320" display setting needs to be selected for the Add-On QVGA board.

```
/* vo_activate.c
   Activate SVO with attached Add-On QVGA board.
*/

/*************************************************************
  Section: Includes
*/

#include "ddb_api.h"      /* display database with display timings */
#include "pl111_api.h"    /* SVO (Singlelayer video out) driver API */
#include "csi_display_init.h"


/*************************************************************
  Section: Local Functions
*/

void VO_Activate(void) {

    /* send CSI init sequence for NL2432HC22-41K */
    Csi_NecDisplay_Init()

    /* init SVO 0 and register it's master bus error interrupt */
    eePL111_Init(0);
    eePL111_EnableInt(0, EE_PL111_INT_MBE);

    /* set SVO 0 to colourmode RGB565,
               select used display "NEC_QVGA_TOUCH_240x320"
               and set framebuffer to address 0xf7800000 */
    eePL111_SetupDisplay(0,
                         EE_PL111_COLOR_MODE_16BPP_MODE565,
                         "NEC_QVGA_TOUCH_240x320",
                         (uint32_t)(0xf7800000));
    eePL111_EnableDisplay(0);
}
```

5.3 Touch screen control

There is no example code available for the touch screen control.