AN4032
Application note

Interfacing an HD44780 2-line LCD display with the STM8SVLDISCOVERY

Introduction

This application note describes how to use a firmware example that drives a standard HD44780 2-line LCD display with the STM8SVLDISCOVERY.

The example uses an LCD display driver that has been developed to be as generic as possible, so it can be reused with any STM8 device.

The firmware package associated to this application note includes a pre-configured project template which facilitates development of your own application to use an HD44780 LCD display module.
The LCD interface offers many ways to easily create a user interface to display text messages and other application related information, whilst using very few internal hardware resources.

The HD44780 is a common LCD controller and is very widespread through various formats, 1 X 8, 2 x 16, 2 x 20, 4 x 20 and others. This example uses the very popular 2 x 16 format (2 lines of 16 characters) which is now offered at a very reasonable price. Other LCD display modules based on compatible ST7066 or KS0066U controllers could also be used.

This application example intentionally uses the least possible microcontroller pins as the interface bus to the LCD module is 4-bits wide.

This application example requires additional hardware:

- LCD module HD44780
- Potentiometer to adjust LCD contrast.

Reference documents

- STM8SVLDISCOVERY STM8S value line discovery (UM1482).
- STM8SVLDISCOVERY quick start guide (UM1480)

All these documents are available at http://www.st.com.

Contents

List of tables

List of figures

1 Application description

1.1 Hardware required

This application uses the STM8SVLDISCOVERY on-board LED LD1, an HD44780 or equivalent 2-line LCD display and a 10 kΩ to 20 kΩ potentiometer to adjust LCD contrast. Some LCD models include an LCD backlight feature but this application note does not deal with this implementation. The LCD display module used in this example is a WINTEK WD-C1602R which is fully compliant with an HD44780 controller, it has 16 characters X 2 lines and a character format of 5 X 8 dots. The LCD current consumption for logic is typically 2 mA.

1.2 STM8SVLDISCOVERY hardware settings

JP1 (power supply jumper) must be put in the 5V position, to provide the correct VDD for LCD module operation. There are no other specific settings for the STM8SVLDISCOVERY. The LCD can be:

- Connected directly with extra wires to the STM8SVLDISCOVERY CN1, CN2 and CN3 connectors.
- Interfaced with a daughterboard (connected via these same connectors, plus CN4 for mechanical support) that receives the LCD module and the potentiometer with a proper PCB layout (not detailed here).

1.3 HD44780 LCD display module description

The LCD display module includes 2 16-character lines and a 16-pin connector. Each character position is identified by its row number (1 or 2), and its column number (from 1 to 16). For example, when used in the software library: Location (2, 10) refers to the 10th character of the 2nd line. See *Table 1 on page 6* for connector details.

Figure 1. HD44780 LCD display module

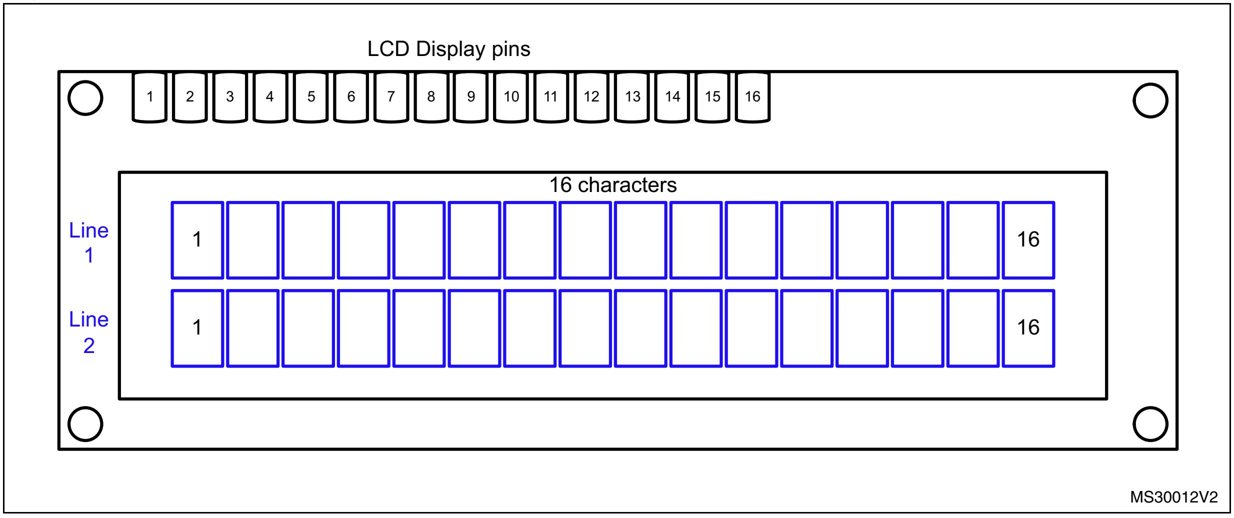


Table 1. LCD display module connector pin assignments

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GND	VCC	VEE or V0	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	NC	NC
0V	+5V	LCD power supply	Register Select 0: Command 1: Data	Read/Write 0: Write 1: Read	Enable LCD module	4-bit low order data bus				4-bit high order data bus				-	-

1.4 Hardware interface application schematics

The LCD display module can easily be interfaced with the STM8SVLDISCOVERY using the application schematics (*Figure 2*).

On the STM8SVLDISCOVERY side, the application uses 7 microcontroller I/Os, see the proposed hardware configuration in *Figure 2*. When required, users can easily modify the pin selection in the application software by editing the LCD port declarations in the *HD44780.h* header file. The only considerations to take into account are the pins' compatibility according to their function:

- For example, PE5 supplies LCD drive current due to its high sink capability (can drive more current than a standard port). If you change the MCU I/O which powers the LCD module, you must ensure that it features high sink capability.
- Some pins of the STM8S003K3 have no internal push-pull capability so they cannot be selected for the LCD port, control or power pins. Please read the pins description in the STM8S003K3 datasheet carefully if hardware configuration change is required.

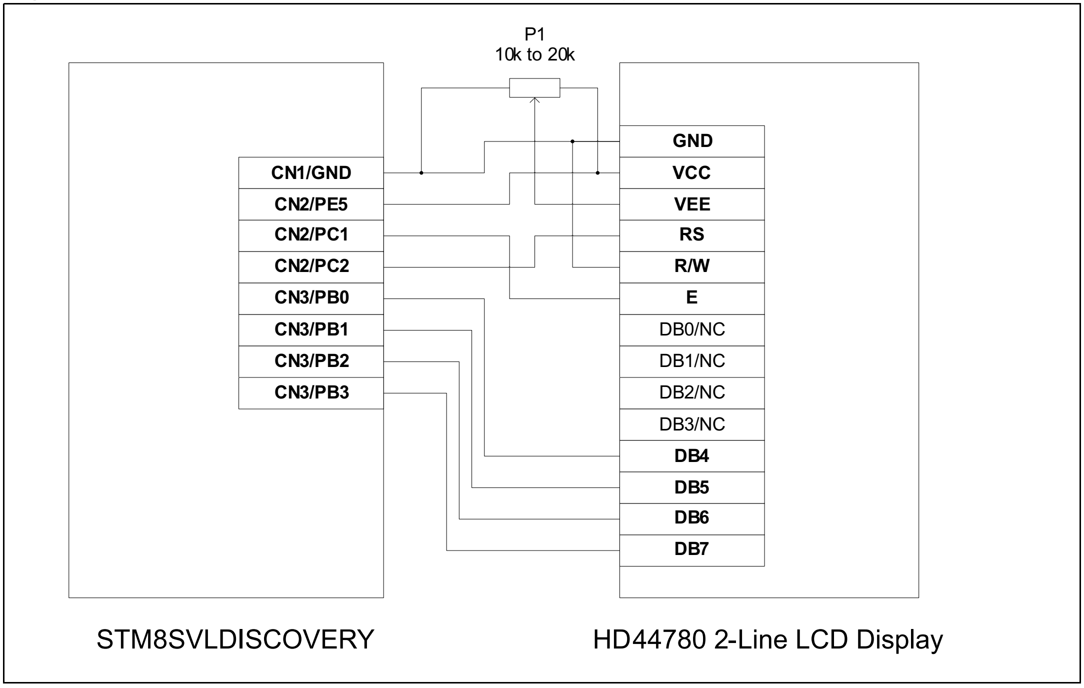
On the LCD display side:

- In order to display a character or send a command to the LCD, a byte corresponding to the character ASCII code or to the command byte value must be sent. Data can be sent from the MCU to the LCD in 4-bit or 8-bit operation.
  - In 8-bit mode, this operation is done in one send.
  - In 4-bit mode, this operation requires two sends of 4-bit data.

  This application uses 4-bit operation, so only DB4 to DB7 need to be connected (LCD 4-bit data bus) and DB0 to DB3 are not connected.
  If the LCD port is changed, the user must assign the lower nibble of a new MCU port to the LCD data bus. For example, in this application example MCU port B is the LCD port with PB0 to PB3 pins. If a different MCU port was chosen, the choice of possible LCD port pins is limited to Px0 to Px3 (due to firmware management), where x can be A, B, C, etc… according to the pins availability on the device.
- The $R/\overline{W}$ pin of the LCD display module is connected to GND (Write is active).
  Only Write operations to LCD memory are performed in this example. Data are exclusively written into LCD and the firmware provides adequate timing needed for each command without polling the internal LCD busy flag.

A potentiometer is connected to VEE/V0. Contrast can be adjusted accordingly.

Figure 2. Hardware interface of the LCD display and the STM8SVLDISCOVERY



Note: After power ON, the LCD display must be initialized once only. During a debug session, user can be led to reset the MCU and then to force a second initialization while the LCD has not been powered OFF. To prevent the LCD display from an unexpected initialization, the LCD pin VCC is connected to the MCU pin PE5 to manage by software the LCD power. This ensures that an LCD power OFF is performed before any LCD initialization attempt.

Once your code has been completely developed and debugged, it is possible to fix the LCD pin VCC by connecting it permanently to the +5V power supply and by removing PE5 pin management in software. This will free-up one more pin on the microcontroller side.

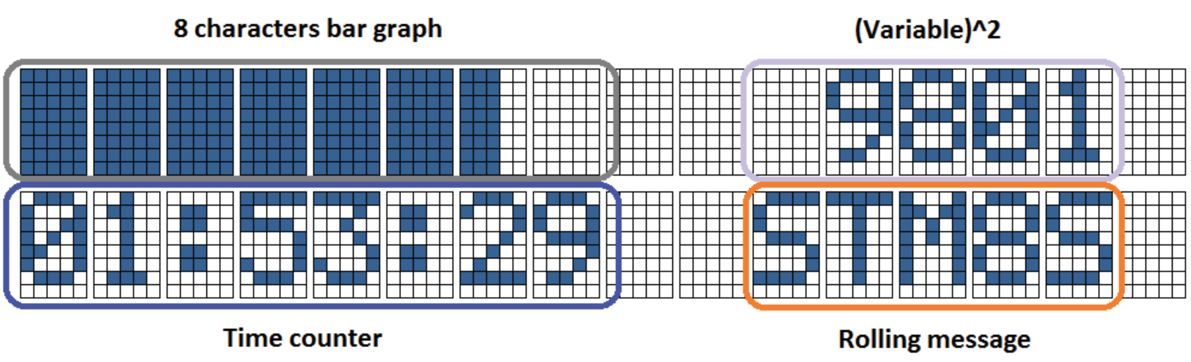
1.5 Application principle

1.5.1 Overview

This application note demonstrates some of the LCD functions provided in the library and explains how they can be combined to create your own menus for the final application. Once the hardware has been connected to the STM8SVLDISCOVERY and the software example loaded in the STM8S device memory, a welcome message “STM8SVLDISCOVERY” is displayed on the LCD screen. The user is then invited to press the on-board push-button to continue the demonstration. This brief LCD demonstration simultaneously refreshes 4 changing items such as shown in *Figure 3*:

- an 8-character bar graph using custom characters defined in the LCD’s memory
- a time counter that counts minutes, seconds and hundredths of seconds
- the square value of an incrementing variable
- a rolling text message that is updated every second

Figure 3. HD44780 LCD display demonstration



This firmware example uses some of the basic functions available in the HD44780 driver (initializing the LCD, printing characters strings to the LCD, using LCD display ON/OFF, setting cursor position, clearing the LCD display, etc…).

Up to 8 user characters (5 x 8 dots each) can be managed to create special characters or even limited graphics by combining them, as for instance, on a 4 x 2 characters matrix displayed on 2 lines. *Section 1.5.3* describes how to create your own characters.

1.5.2 LCD interface

This LCD interface allows users to control the HD44780 module to display text, messages or other information related to the needs of the application. The LCD port 4-bit wide data bus is managed by 4 lines connected to PB0 to PB3 microcontroller pins. In addition, there are 3 control pins RS, R/$\overline{\text{W}}$ and E.

- R/$\overline{\text{W}}$ is set to write mode (0) as data is always sent from the microcontroller to the LCD module and not in the opposite direction.
- RS is the register select pin. This pin allows the LCD module to decide if data coming from the microcontroller must be interpreted as an LCD command, or if it must be treated as text data to be displayed on the LCD screen.
- E enables LCD access for each MCU to LCD transaction. As soon as a communication starts from the microcontroller side, this pin must be put high to enable the LCD. It returns to low level once the procedure is completed.

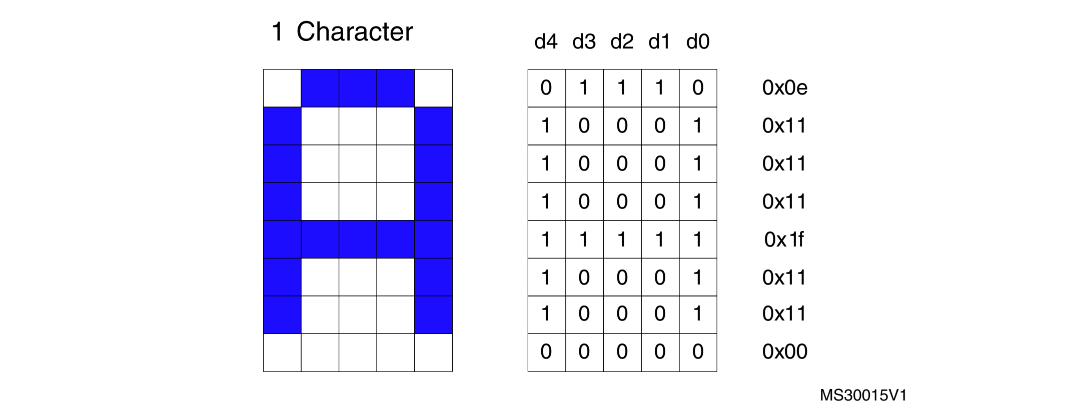
1.5.3 Custom characters definition

This section describes how users can create their own custom characters.

Each custom character must be defined in the LCD display module CGRAM (Character Generator RAM) of the HD44780 device.

- For 5 X 8 pixel based LCD characters, up to 8 characters can be stored in the CGRAM.
- A character is defined by 8 bytes (only 5 bits are considered, the others are not set) representing the character pattern for each row.
- A displayed dot is stored as 1 in the character bytes, otherwise it is 0.
- These 8 bytes can be stored in a character table that will be transferred in the CGRAM memory of the HD44780 controller.

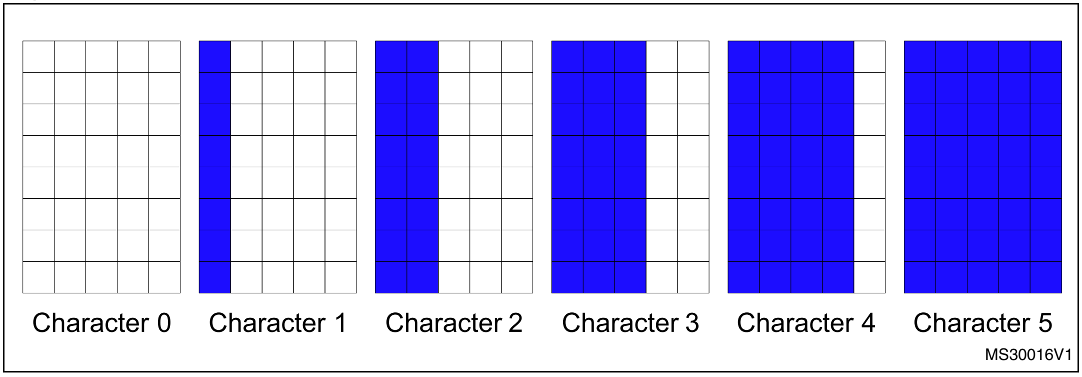
Figure 4. Example of A character definition



d4	d3	d2	d1	d0	
0	1	1	1	0	0x0e
1	0	0	0	1	0x11
1	0	0	0	1	0x11
1	0	0	0	1	0x11
1	1	1	1	1	0x1f
1	0	0	0	1	0x11
1	0	0	0	1	0x11
0	0	0	0	0	0x00

To define the application custom characters related to the bar graph used in our example, we created the following 6 characters stored in a single tab.

Figure 5. Custom characters of application example



These characters are then written into the HD44780 LCD display module at the start address of CGRAM @0x40. These characters are defined from number 0 to 5 and can be called directly by printing their hexadecimal code.

For instance, the C instruction `LCD_printstring("\3\n")` displays Character 3 at the current cursor position. Displaying the bar graph prints a string containing the characters' numbers (progressively changing from Character 0 to Character 5) at the same LCD display location.

2 Software description

2.1 STM8S peripherals used by the application

This application example uses the following STM8S peripherals and settings:

- TIM4: generates a 1 ms time base, used for precise delays and other timing necessary for LCD display and menu display management.
  - TIM4 input clock is set to 16 MHz, and the prescaler to 128 (f= 125 kHz)
  - TIM4 period is set to124 thus producing a time base of 1 ms.
- GPIOs:

  STM8SVLDISCOVERY standard resources:
  - Port C (PC4) set as input floating (push-button B1)
  - Port D (PD0) set as output push-pull (LED LD1)

  HD44780 interface control resources: The LCD display module needs a port dedicated to the data bus (4-bit wide), a port dedicated to LCD control (E, RS) and a port to supply the LCD power (VCC pin):
  - Port B (PB0 to PB3) set as output push-pull is the LCDPort (data bus)[a]
  - Port C (PC1 and PC2) set as output push-pull is the LCDControlPort (RS and E pins)[a]
  - Port E (PE5) set as output push-pull is the LCDPwrPort (VCC of LCD)[a][b]

  The MCU I/Os selected to drive the HD44780 interface are configured through `#define` statements in the *HD44780.h* file as follows:

```
#define LCDPort GPIOB               LCD data bus assigned to PB
#define LCDControlPort GPIOC        LCD control pins E & RS assigned to PC
#define LCDPwrPort GPIOE            LCD power supply pin assigned to PE
#define LCDPwrPin GPIO_PIN_5        LCD power supply pin assigned to Px5
#define LCD_Enable GPIO_PIN_1       LCD enable pin assigned to Px1
#define LCD_RS GPIO_PIN_2           LCD RS pin assigned to Px2
```

- CLK (clock control): HSI (high speed internal 16 MHz) is selected as the main clock source. The SYSCLK prescaler is set to 1 so that CPU frequency is equal to 16 MHz.

[a] If the user wants to use other pins than the proposed ones, they must be configurable as output push-pull.

[b] If another power pin is used instead, it must have a high sink capability.

2.2 Configuring STM8S/A standard peripherals library

2.2.1 For use with this application

The *stm8s_conf.h* file of the STM8S/A standard peripherals library configures the library by enabling the peripheral functions used by the application. The following define statements must be present:

```
#include «stm8s_clk.h»
#include «stm8s_tim4.h»
#include «stm8s_gpio.h»
```

2.2.2 For other uses of the HD44780 LCD display driver

To reuse the HD44780 LCD display module files outside of this application project, especially for any other STM8x family project, the following files should be imported in the new project:

- *HD44780.c, HD44780.h, delay.c, delay.h.*
- TIM4 and GPIO libraries should also be imported unless other resources are selected:
  - As all delays are managed by the TIM4 peripheral, the TIM4 peripheral resource or other available timer should also be included in the project and initialized with the parameters of this application note example.
  - Similarly, ports of STM8x microcontroller must be managed with the GPIO standard library of the selected device.

    Please refer to the product datasheets to verify if the required peripherals or resources are available on the microcontroller being used in the new project.

This application software includes a pre-configured project template in the *Projects* folder of the application note resources named *HD44780_template.*
This provides an easy starting point to develop an HD44780 LCD based application. It includes HD44780 LCD display management and contains all the necessary software resources to drive an LCD display module with the minimum starting configuration.

2.3 Application software description

2.3.1 Main application flowchart

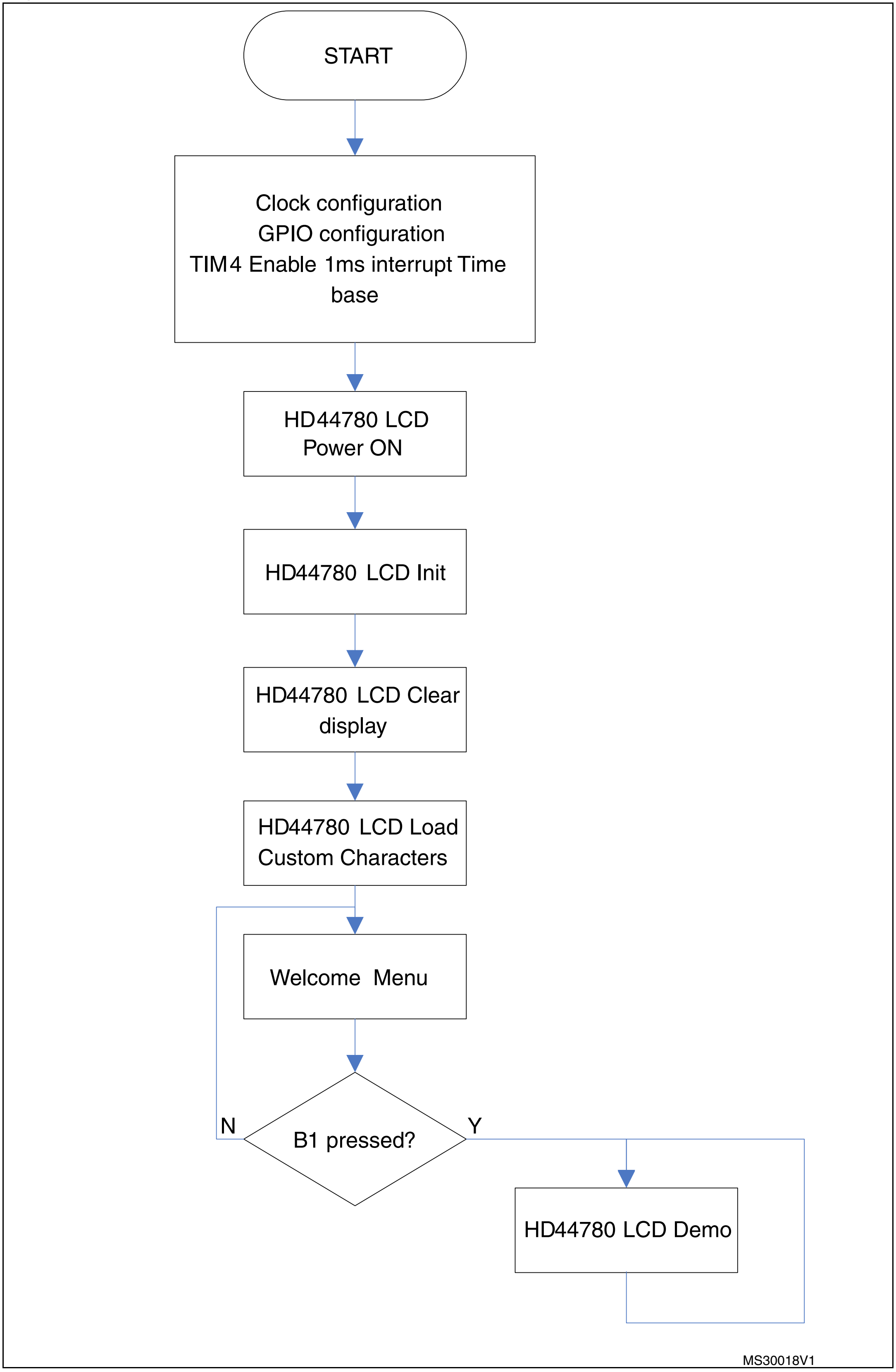
The main application initializes the peripherals that are used in this LCD display module demonstration in the following sequence:

1. Configures CLK (clock control), GPIO and TIM4 peripherals.
2. Enables the LCD power pin via the dedicated LCD power I/O (PE5 here).
3. After LCD power ON, a 100 ms delay is respected to guarantee proper LCD operation.
4. Initializes HD44780 LCD. Clears LCD screen.
5. Loads custom characters into the CGRAM of the LCD display device.
6. Displays welcome menu. Waits until B1 push-button is pressed to enter LCD demo.
7. Once the LCD demo is entered, the application remains in this state until STM8SVLDISCOVERY start-up.
   a) On the top left, an 8-character bar graph (*Figure 3*) is progressively filled then cleared.
   b) On the top right, the square value of a variable varying from 0 to 128 is permanently updated.
   c) On the bottom left, a time counter including minutes, seconds, tenths and hundredths of seconds is refreshed. This time counter is updated every 1 ms interrupt of TIM4. The total duration of this time counter is one hour maximum but it can be reset at any time by pressing B1.
   d) On the bottom right, the 5 words `STM8S`, `Value`, `Line`, `8-Bit` and `Micro` are displayed successively at a rate of 1 word per second.
   e) The display is turned ON and OFF periodically.

There is no particular connection between these 4 displayed items.

This example demonstrates the performance of the STM8SVLDISCOVERY. It shows that the LCD display module can be driven in different ways and that the STM8SVLDISCOVERY has the capability to refresh different kinds of information at the same time.

Figure 6. Main application flowchart



2.3.2 HD44780 LCD display module library description

The HD44780 LCD display module library includes 2 files: *HD44780.c* and *HD44780.h*. These 2 files are associated with *delay.c* and *delay.h* for timings that are necessary for LCD display management. As a reminder, all delays generated are linked to the TIM4 usage and the recommended settings. *Table 2* only describes the functions included in the *HD44780.c* file that make available actions that can be performed with the LCD display.

Table 2. HD44780.c functions relevant to LCD display

Function Name	Description	In Parameter(s)	Return value
void LCD_LOAD_CGRAM (char tab[], u8 charnum)	Load custom characters into HD44780 CGRAM. Preceding this function, an LCD command instruction with CGRAM starting address as argument must be sent (LCD_CMD(address)).	**char tab[]:** table containing custom characters data (8 bytes a character). **u8 charnum:** number of characters that will be loaded	
void LCD_PWRON(void)	Set HIGH the LCD power pin assigned to LCD VCC power line (LCD ON).	None	None
void LCD_PWROFF(void)	Set LOW the LCD power pin assigned to LCD VCC power line (LCD OFF).	None	None
void LCD_ENABLE(void)	Set HIGH the LCD Enable pin assigned to LCD E line for 2ms (then return to LOW).	None	None
void LCD_CMD (unsigned char cmd_data)	Send command to LCD (instruction, no text data).	**unsigned char cmd_data:** command value	None
void LCD_INIT(void)	Initialize the LCD display module in 4-bit mode. This function can only be called after LCD power ON. Manages mandatory delays between each command.	None	None
void LCD_printchar (unsigned char ascode)	Print 1 character to LCD display module.	**unsigned char ascode:** character code	None
void LCD_printstring (unsigned char *text)	Print one character string to the LCD display module (string must end by \n terminal).	**unsigned char *text:** characters string ended by \n	None
void LCD_CLEAR_DISPLAY (void)	Clear LCD display and set cursor at home position.	None	None
void LCD_2ndROW(void)	Set cursor to first column of 2nd row.	None	None
void LCD_HOME(void)	Set cursor to home position, first column of 1st row.	None	None
void LCD_LSHIFT(void)	Shift display left. All characters on screen are shifted one position to the left.	None	None
void LCD_RSHIFT(void)	Shift display right. All characters on screen are shifted one position to the right.	None	None
void LCD_DISP_ON(void)	LCD display is turned ON.	None	None
void LCD_DISP_OFF(void)	LCD display is turned OFF.	None	None

Table 2. HD44780.c functions relevant to LCD display (continued)

Function Name	Description	In Parameter(s)	Return value
void LCD_LOCATE (u8 row, u8 column)	Cursor is set at location defined by row and column. Eg: LCD_LOCATE(1,4) sets cursor on first line in 4th position. If position is out of range, then no action. Physical address of 1st line is 0x00; 2nd line is 0x40	**u8 row:** ROW number (must be between 1 to 2) **u8 column:** COLUMN number (must be included between 1 to 16)	None
void LCD_printf (const char *fmt, ...)	Print characters strings with specified format (***does not support all formats depending on compiler limitations***). Needs to include stdarg.h library in header file.	**const char *fmt, ...:** characters string with specified format	None

3 Conclusion

LCD character display modules are very common in many 8-bit microcontroller based applications. With a very affordable price and a quick implementation time, they offer the easiest way to realize a simple and efficient interface for various applications in industry, consumer, metering, telecommunications and other domains.

This short example includes all the basic functions to drive an HD44780 LCD controller and any other compatible LCD display modules. This brief demo provides STM8SVLDISCOVERY users with many possibilities to quickly create interfaces and menus.

This application note can also be tailored for any microcontroller of the STM8 family with slight firmware modifications.

4 Revision history

Table 3. Document revision history

Date	Revision	Changes
16-Feb-2012	1	Initial release.

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