EYT. Design with STM32-comStick from HITEK

STM32-comStick
STM32-comStick is a HW/SW tool done by HITEX. It connects to PC USB port and through that makes available STM32F107 HW resources. The boards comes with MICRIUM RTOS webserver preinstalled.

Additional informations on: www.hitex.com
The simplest thing to develop a new project with STM32-comStick from HITEX is to start from a working example provided by HITEX and then modify it. In the next slides you will learn how to do it.

**Suggestions:**
Have a look on the following video guide:

*HiTOP Universal user interface for all Hitex test and analysis tools*

It explains in a clear and simple way HITEX IDE main functionalities.
Let’s start with the following assumptions:

Use **STM** libraries **ver.3.1.0** available at this website:

As starting HITEX example we will use the following:
**SYSTICK LED blinking application with GNU**
available at this website:

Our working directory will be:
.../Prova1

Working directory structure will be:
...ESEMPI-SW/STM32-Examples/HITEX_comStick
**STEP n.1**

Copy the directories from this path:

```
C:\ESEMPI-SW\STM32-Examples\HITEX_comStick\SYSTICK LED blinking application with GNU
```

To our working space: 

*Prova1*

See picture beside for additional explanations
Design with STM32-comStick from HITEX

**STEP n.2**

Run **HiTOP53-STM32-comStick**

The page below should be displayed.

From this page click on: **Open an existing project**

**NOTE:**

STM32-comStick must be connected to the PC
STEP n.2/Cont.

From the actual menu move into the following directory:
 …\Prova1\AP-ARM-0126\HiTOP\STM32-ComStick

Select file *project.htp* and then click on **Open**.

1) On the page that appears select **STM32-comStick** and press **NEXT**.
2) A new page will appear where you click on **press CONNECT**.
3) Again new page will appear where you click on **I WANT TO CONTINUE EVALUATION**.
STEP n.2/Cont.

Now you should see the page below, **click on OK**.
STEP n.2/Cont.

If you did a good job you should see a page like this one
STEP n.3

To verify the application run it by clicking on this icon

If all OK, executing the program you should see the first orange led on the left blinking (see below, red arrow).

Now we will modify the program in order to:

- *Use STM library only*
- *Get all LED blinking*
STEP n.3/Cont.

Before modification you have to stop program execution clicking on this icon

To enter in editor do a right Click inside the page containing C code and select

Switch to Edit Mode.
STEP n.4

HITEX application basically cover the following functions:
- MCU Configuration
- GPIO Configuration
- CLOCK (RCC) Configuration
- SYSTICK configuration
- INTERRUPT (NVIC) configuration

LED blinks when SYSTICK expires, this is managed via interrupt.

Interested files are:
- main.c: Setup routines and main loops
- tm32f10x_it.c: Interrupt vectors
- stm32f10x_lib.c: Library definition module
- stm32f10x_gpio.c: Library for GPIO module
- stm32f10x_rcc.c: Library for RCC module
- stm32f10x_systick.c: Library for SYSTICK module
- stm32f10x_nvic.c: Library for NVIC module
Here are some other important files for MCU configuration:

- `main.h`: General inclusions
- `stm32f10x_conf.h`: Library configuration file
- `stm32f10x_lib.h`: Library inclusions file
- `stm32f10x_type.h`: Definitions and types
- `stm32f10x_it.h`: Interrupt vector pre-declarations

Special note to file `stm32f10x_conf.h`. It provides peripheral enable/disable and set the system clock used to get USART desired baud rate.

Here are additional info on the files described above:

*Application Example AE-CORTEX-0102.pdf*

STEP n.4/Cont.

To modify HITEX program we should know LED connections. To do this open STM32 data sheet included in STM32-comStick in this directory: 
C:\Program Files\Hitex\HiTOP53-STM32-comStick\STM32-ComStickView\Doc
File name: 
stm32-io-board-ds.pdf
LED connection is done as follow:

Set the desired port to 'high' in order to light up the corresponding LED.
Open file **main.c**
look at the function:
```c
void ToggleLED( void )
```
And delete its content as shown in **step 1** and **step 2**.
the new function content for:

```c
void ToggleLED(void)
```

Is the following.

```c
static u8 n=0;
n++; if (n==1) {
    // LED ON
    GPIO_SetBits(GPIOE, GPIO_Pin_15);
    GPIO_SetBits(GPIOB, (GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_5 | GPIO_Pin_9));
} if (n==2) {
    // LED OFF
    GPIO_ResetBits(GPIOE, GPIO_Pin_15);
    GPIO_ResetBits(GPIOB, (GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_5 | GPIO_Pin_9));
    n=0;
}
```
STEP n.4/Cont.

search for function `void IO_init ( void )` and add the following lines:

```c
GPIO_InitStructure.GPIO_Pin = (GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_5 | GPIO_Pin_9);
GPIO_Init(GPIOB, &GPIO_InitStructure);
```
STEP n.4/Cont.

To compile the program click on this icon. If everything is OK you should see the following on the screen. Click on OK and you will execute the program. All LED should blink.
BreakPoint

To set a B.P. move on the interested code line and left click with the mouse. The red mark will indicate Breakpoint is set.
Watch Variables

To put variables in the Watch Window just select them, right click mouse and chose Add Watch. Variable will appear in the watch window.
Reset works on the MCU initialization routine
Execute Script, click on low arrow, this makes MCU reset.
Design with STM32-comStick from HITEX

**STOP** stop program execution

**RUN** run program from actual Program Counter content

**Set/Remove BrakPoint** on the line where cursor is.

**Show PC** move cursor to line code pointed by Program Counter

**StepInto**

**StepOwer**
ST-MCU
http://www.st.com/mcu/index.html

STM32
http://www.st.com/mcu/inchtml-pages-stm32.html

Documents and files for family STM32
http://www.st.com/mcu/familiesdocs-110.html

STM32 for motor control

MCU Training & Seminars

Product Brochures & Selectors
http://www.st.com/stonline/products/promlit/p_microcontrollers.htm

Example for STM32
http://emcu.altervista.org/