

UNIVERSITY OF SAINT THOMAS – SAINT PAUL – MN

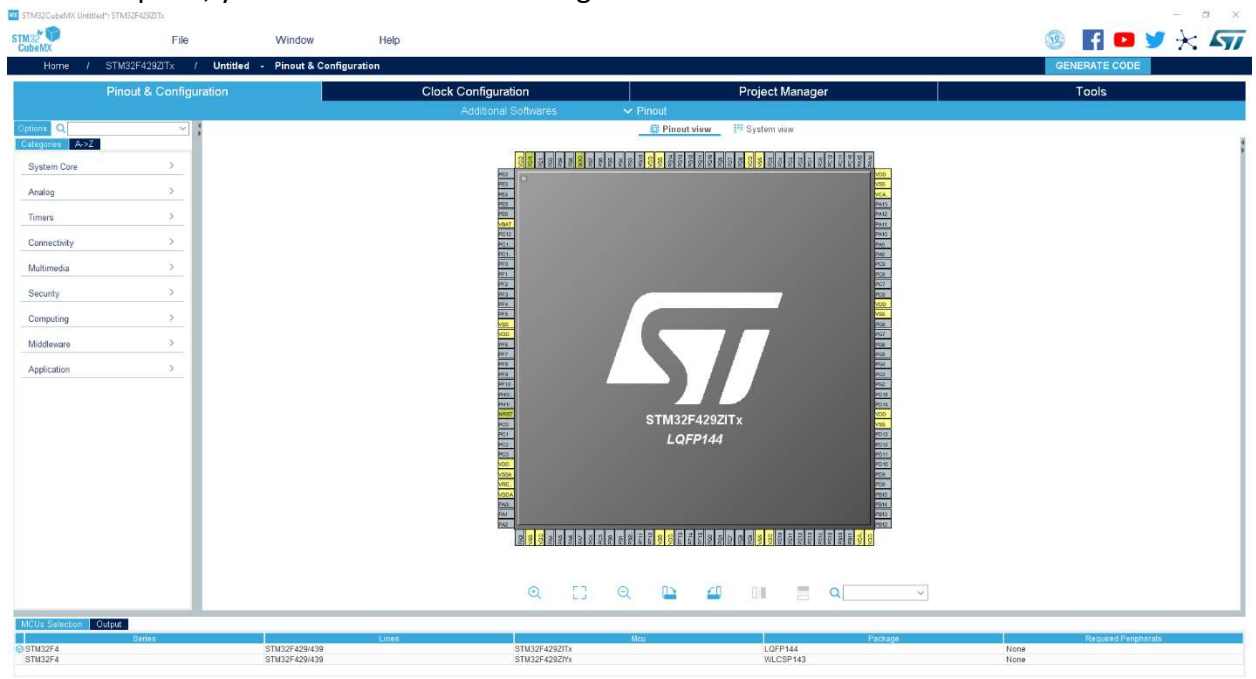
Electrical and Computer Engineering

ENGR 432: REAL TIME SYSTEMS

To help you get started, here are the steps to work with STMCubeMX

CubeMX (HAL) Guide

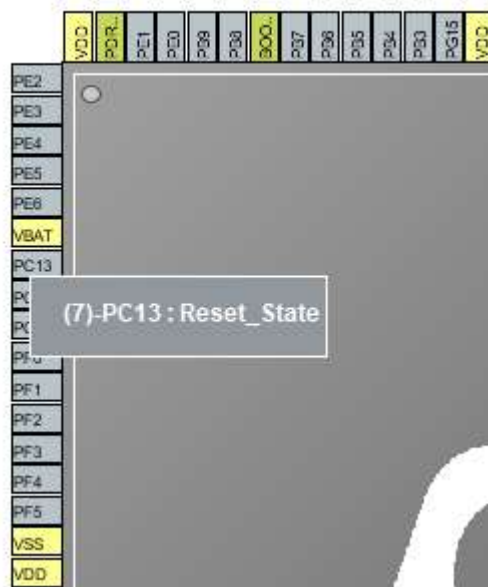
1. Select the “File” option at the top of the window.
 - a. Select New Project and a new GUI will pop-up
 - b. Type into the Search bar on the left “STM32F429ZI”
 - c. Click on the board in the “Part No.” column in the bottom right list
 - d. Click “Start Project” in the top right.
 - e. It may ask if you want to initialize the defaults – choose yes
2. Once complete, you should see the following screen:



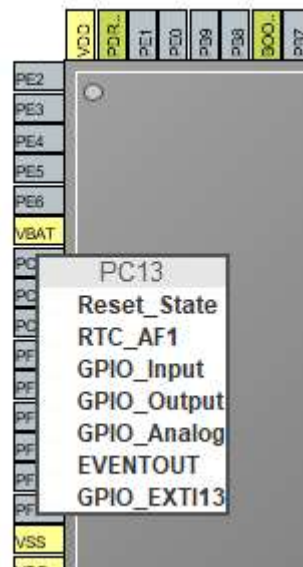
3. In particular we are looking to complete the following tasks:
 - a. Set up PC13 (the user button) as an interrupt
 - b. Toggle the LEDs (PB0, P7, and PB14)
 - c. Set up Timer(s)

4. Let's start with the Pinouts

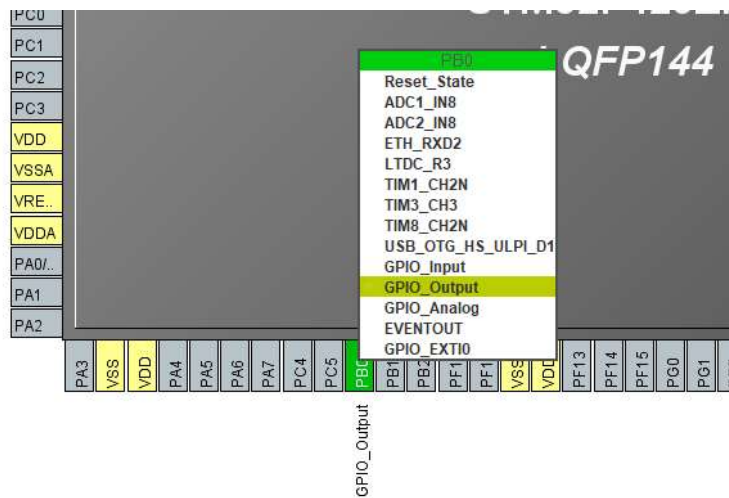
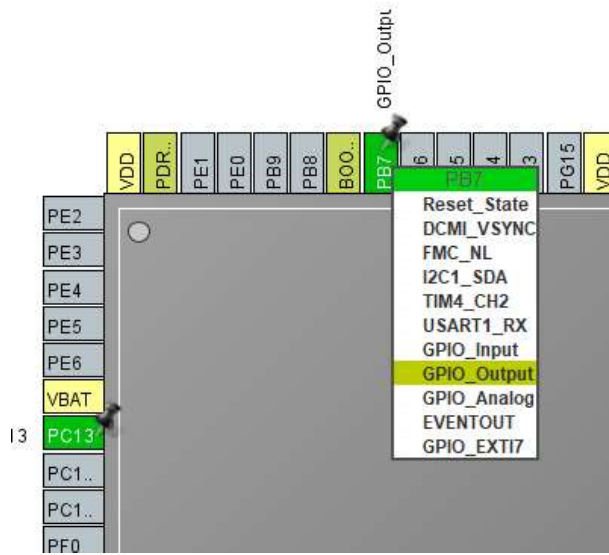
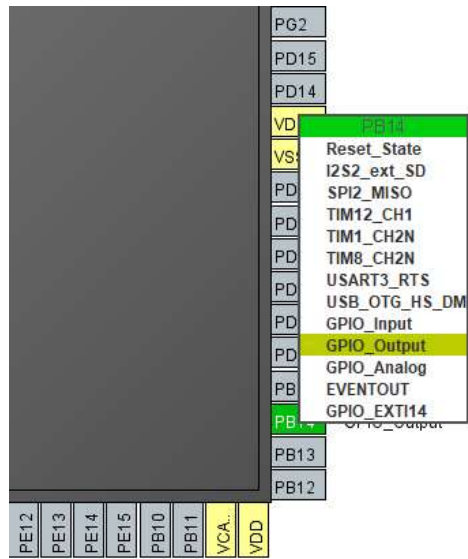
- a. If you zoom in on the STM32F429ZITx, you can see tiny little buttons, each



- b. In particular, the 8th one down on the left side is PC13, which by default is in its "reset_state". We want to change that to GPIO_EXTI13 (External Interrupt).



- c. Similarly, we want to locate the PB0, PB7, and PB14 ones, and set them up as well. These ones should be setup as GPIO_Output.



5. Now that we've told the board which pins we want to use (and for what), we can configure the clocks/timers.
 - a. On the left hand side, there is a Timers tab. Open it up and hover over a clock, it will tell you the uses. In this case, we want to setup one as a "general purpose timer". So, I selected Timer2, and changed the clock source to Internal Clock.

TIM2 Mode and Configuration

Mode	
Slave Mode	Disable
Trigger Source	Disable
Clock Source	Internal Clock
Channel1	Disable
Channel2	Internal Clock
Channel3	Disable
Channel4	Disable
Combined Channels	Disable
<input type="checkbox"/> Use ETR as Clearing Source <input type="checkbox"/> XOR activation <input checked="" type="checkbox"/> One Pulse Mode	

- b. A new section became available, where we are able to select ARR and PSC values for the clock, to ensure it is a 1 second clock. After testing, I determined that an ARR of 21000 and PSC of 999 worked to get a near perfect 1 second timer, but your results may vary.

☒ Parameter Settings
 ☒ User Constants
 ☒ NVIC Settings
 ☒ DMA Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings
 Prescaler (PSC - 16 bits value) : 999
 Counter Mode : Up
 Counter Period (AutoReload Register - 32 bits value) : 21000
 Internal Clock Division (CKD) : No Division

Trigger Output (TRGO) Parameters
 Master/Slave Mode (MSM bit) : Disable (Trigger input effect not delayed)
 Trigger Event Selection : Reset (UG bit from TIMx_EGR)

- c. Under the NVIC settings, we want to enable the TIM2 Global Interrupt checkbox.

Reset Configuration

☒ Parameter Settings
 ☒ User Constants
 ☒ NVIC Settings
 ☒ DMA Settings

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
TIM2 global interrupt	<input checked="" type="checkbox"/>	0	0

- d. Now go back to the left panel, and close up Timers, and select System Core.
- e. Click on the GPIO options, and you should see our 4 items listed

Configuration							
<input type="checkbox"/> Group By Peripherals							
<input checked="" type="checkbox"/> GPIO <input checked="" type="checkbox"/> NVIC							
Pin Name	Signal on Pin	GPIO output le...	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PB0	n/a	Low	Output Push ...	No pull-up an...	Low		<input type="checkbox"/>
PB7	n/a	Low	Output Push ...	No pull-up an...	Low		<input type="checkbox"/>
PB14	n/a	Low	Output Push ...	No pull-up an...	Low		<input type="checkbox"/>
PC13	n/a	n/a	External Interr...	No pull-up an...	n/a		<input type="checkbox"/>

- f. Select the NVIC tab, and check the option to enable EXTI line[15:10] interrupts

Configuration			
<input type="checkbox"/> Group By Peripherals			
<input checked="" type="checkbox"/> GPIO <input checked="" type="checkbox"/> NVIC			
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	0	0

6. Go over to the Project Manager tab
 - a. In the Project tab on the left
 - i. Give the file a name, and select the location
 - ii. Change the Toolchain / IDE to MDK-ARM V5

	Pinout & Configuration	Clock Configuration
Project	Project Settings Project Name: HALTest Project Location: C:\Users\ [redacted] Desktop Application Structure: Basic <input type="checkbox"/> Do not generate the main() Toolchain Folder Location: C:\Users\ [redacted] Desktop\HALTest\	
Code Generator	Toolchain / IDE: MDK-ARM V5 <input type="checkbox"/> Generate Under Root	
Advanced Settings	Linker Settings Minimum Heap Size: 0x200 Minimum Stack Size: 0x400	
	Mcu and Firmware Package Mcu Reference: STM32F429ZITx Firmware Package Name and Version: STM32Cube_FW_F4_V1.23.0 <input checked="" type="checkbox"/> Use latest available version <input checked="" type="checkbox"/> Use Default Firmware Location: C:\Users\ [redacted] STM32Cube\Repository\STM32Cube_FW_F4_V1.23.0 <input type="button" value="Browse"/>	

- b. Click the Code Generator option on the left

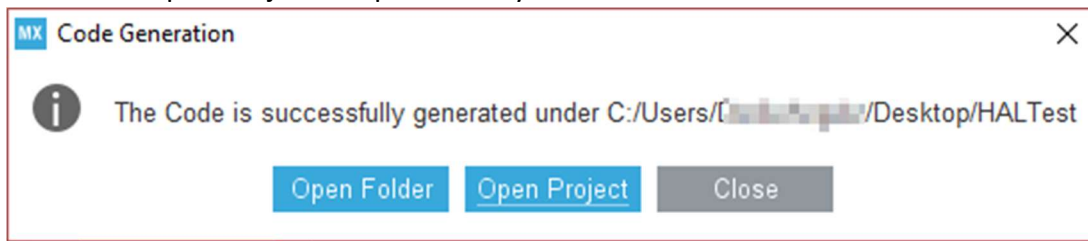
- i. Select Copy only the necessary library files

	Pinout & Configuration	Clock Configuration
Project	STM32Cube Firmware Library Package <input type="radio"/> Copy all used libraries into the project folder <input checked="" type="radio"/> Copy only the necessary library files <input type="radio"/> Add necessary library files as reference in the toolchain project configuration file	
Code Generator	Generated files <input type="checkbox"/> Generate peripheral initialization as a pair of '.c/.h' files per peripheral <input type="checkbox"/> Backup previously generated files when re-generating <input checked="" type="checkbox"/> Keep User Code when re-generating <input checked="" type="checkbox"/> Delete previously generated files when not re-generated	
Advanced Settings	HAL Settings <input type="checkbox"/> Set all free pins as analog (to optimize the power consumption) <input type="checkbox"/> Enable Full Assert	
	Template Settings Select a template to generate customized code Settings...	

7. Hit GENERATE CODE in the top right

GENERATE CODE

8. Choose to Open Project to open directly in Keil



9. From there, the code should be all set up and initialized for use. Obviously, there are some additional functions so accomplish, such as the logic regarding the LEDs, and configuring what actually occurs inside the interrupts.
 - a. A quick way to confirm that the initialization worked is to the following code in main.c file to toggle the LEDs every second, using the built in HAL_delay() function:

```
while(1) {
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_7);
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_14);
  HAL_Delay(1000);
}
```