Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

Date: 4/12/2021

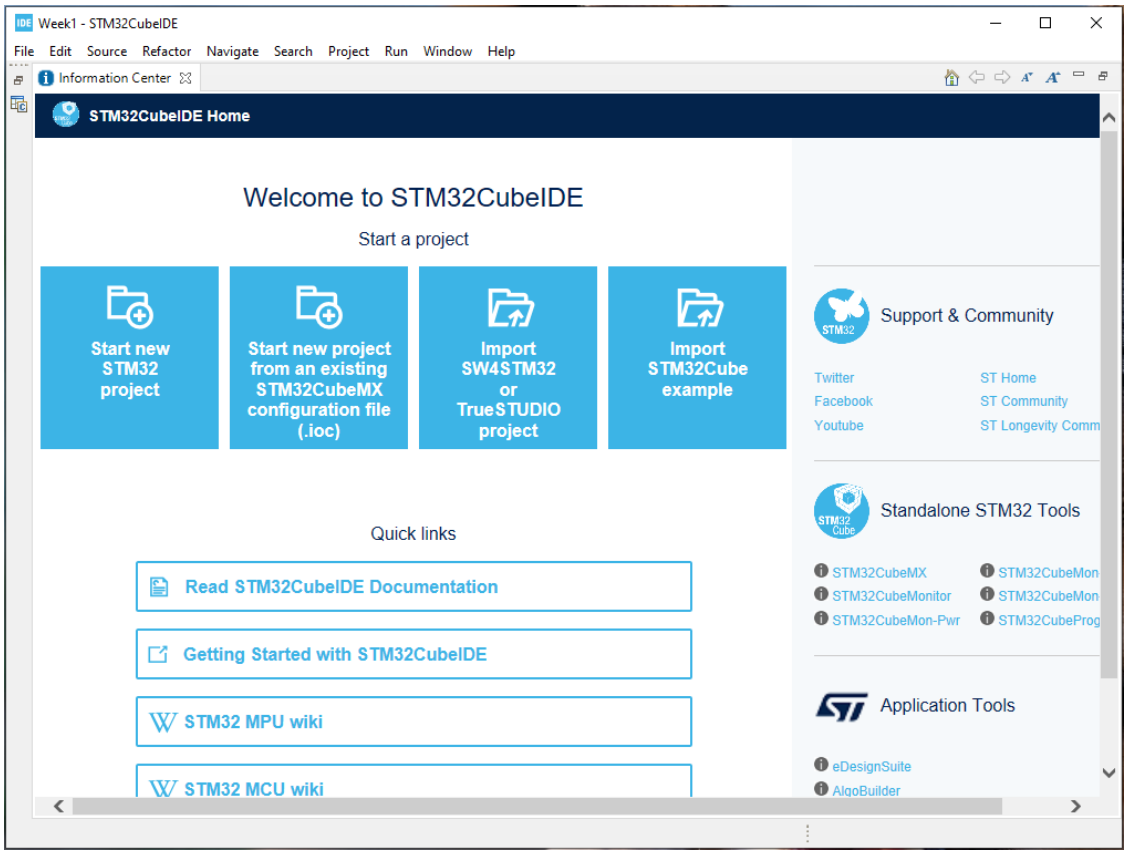
Assignment 1: GPIOHandsOn

The following will document completion of the first assignment for ECE-40293, with the stated goals of:

- A (brief) overview of downloading and installing the STM32CubeIDE
- Use the IDE to generate a project and associated code for the target dev board
- Employ C language programming skills to complete the 3 assigned User Stories, that perform the following:
  - Blinks LED2 three times at startup at a 1 second rate
  - After blinking LED2 from the previous step, blinks the Wifi and BLE LED on/off at a 1 second on/off rate "forever".
  - Pressing the Blue button should generate an interrupt that toggles the LED on/off

1. A overview of downloading and installing the STM32CubeIDE

This process was fairly straightforward: using a Windows 10 PC, download and follow the installer prompts as one would using the standard install wizard format. Upon completion of the install and a PC reboot for good measure, the IDE can be loaded and opens to Welcome dashboard, as seen below.

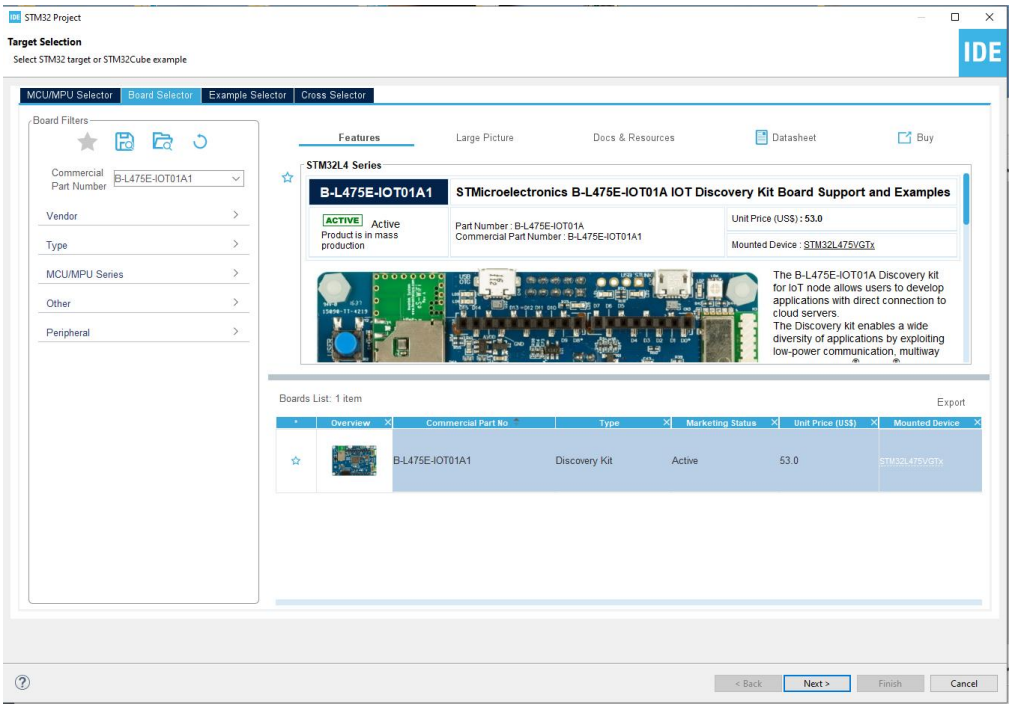


Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

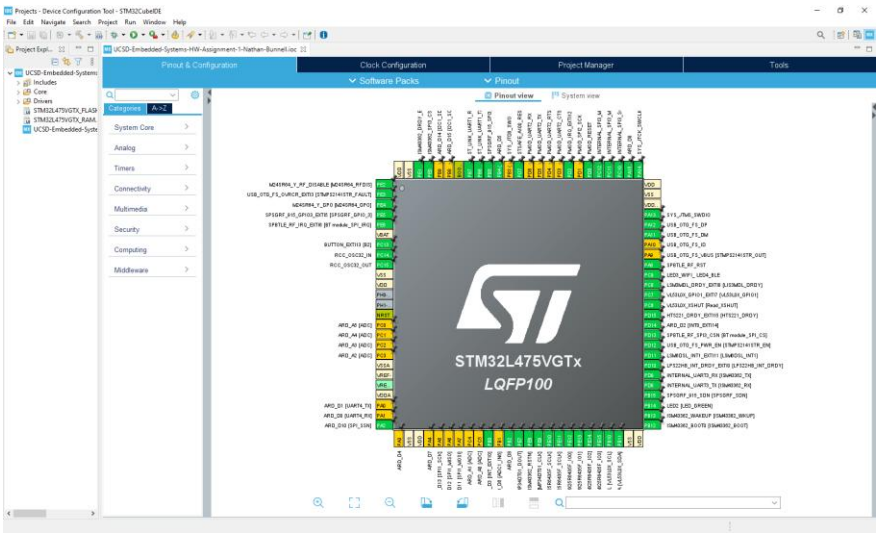
Date: 4/12/2021

2. Use the IDE to generate a project and associated code for the target dev board

Following the prompts in the IDE, select “Start New STM32 project” and wait for the Target Selector interface to appear. Navigate to the Board Selector Tab and enter the model of our target dev board, B-L475E-IOT01A1.



From there, click next and follow prompts to select a project name and location. The project will then proceed to build and launch into the pinout graphical interface with all boilerplate files automatically generated and present in the project organization tree on the left hand of the IDE.



Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

Date: 4/12/2021

3. Complete the 3 assigned User Stories

At this point, we can begin to develop the code necessary to complete the User Stories. The first being:

- Blinks LED2 three times at startup at a 1 second rate

Navigating through the pre-built files to Core/Src/main.c, we can open the main project file in the IDE's text editor. The IDE builds these files with pre-commented sections defining User code locations, to complete this story, we will want to place it in the main() function, after any startup/ init() functions but before entering the primary while(1) loop. To accomplish the 3x toggle, we will use a simple for-loop and leverage the built-in HAL functions, in this case, HAL_GPIO_TogglePin() and HAL_Delay(), to blink the LED2 component on the dev board. See example code implemented below:

```
114
115  /* USER CODE BEGIN 2 */
116
117  /**********************************
118   * User Story 1 implementation begin:
119   *
120   *    Toggle LED2 3x at startup to
121   *    indicate system booting
122   *
123   **********************************/
124
125  for (int i = 0; i < 3; i++)
126  {
127      // Toggle on, delay for short period
128      HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
129      HAL_Delay(500); // 1/2 second, value is arbitrary
130
131      // Toggle off, delay for short period
132      HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
133      HAL_Delay(500); // 1/2 second, value is arbitrary
134  }
135
136  /**********************************
137   * User Story 1 implementation end
138   *
139   **********************************/
140
141  /* USER CODE END 2 */
142
```

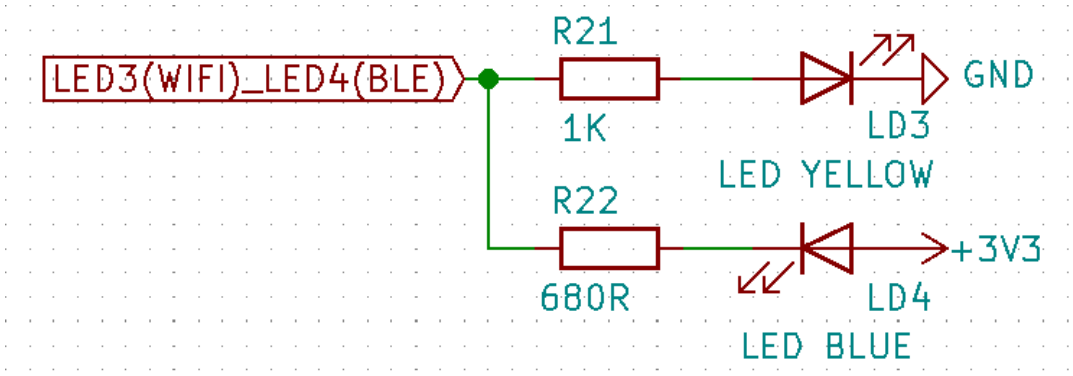
Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

Date: 4/12/2021

The second user story is defined as:

- After blinking LED2 from the previous step, blinks the Wifi and BLE LED on/off at a 1 second on/off rate "forever".

To accomplish this, we will use the same HAL functions and place them in another pre-defined User code space, inside the primary while(1) loop of the main() function, except we will use the pin associated with the LED3 & LED4 components instead of LED2 in this case. Toggling between the two is possible due to the hardware design of the board where, depending on the pin state, current is either flowing from VCC or to GND, with either LED in that path, as seen in this example from the system schematic.




```
45  while (1)
46  {
47    /* USER CODE END WHILE */
48
49    /* USER CODE BEGIN 3 */
50
51    /***********************************
52     * User Story 2 implementation begin:
53     *
54     *  Toggle LED3 (WiFi/BLE) in while(1)
55     *  loop with short delay between each
56     *
57     ***********************************/
58
59    // Toggle, delay for short period
60    HAL_GPIO_TogglePin(LED3_WIFI__LED4_BLE_GPIO_Port, LED3_WIFI__LED4_BLE_Pin);
61    HAL_Delay(500); // 1/2 second, value is arbitrary
62
63    /***********************************
64     * User Story 2 implementation end
65     *
66     ***********************************/
67  }
68  /* USER CODE END 3 */
69 }
70
```

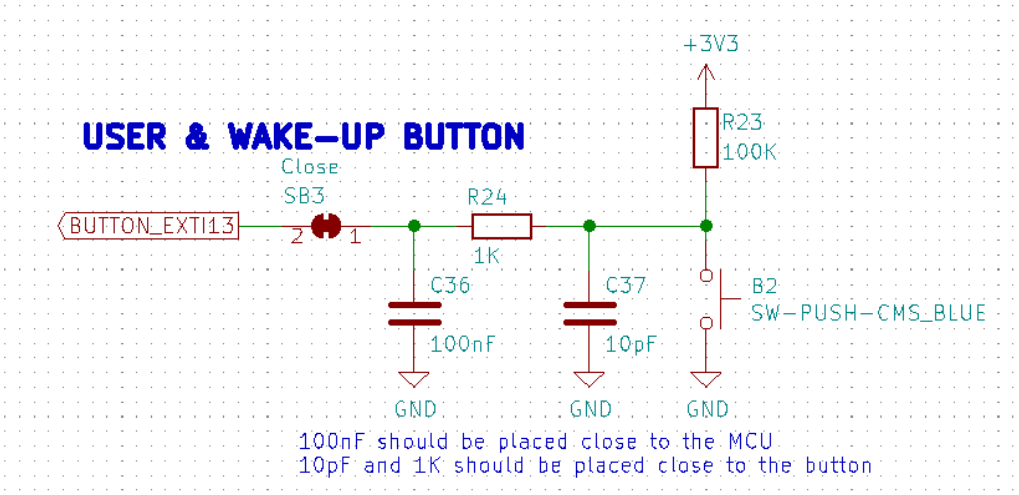
Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

Date: 4/12/2021

The third user story is defined as:

- Pressing the Blue button should generate an interrupt that toggles the LED on/off

To accomplish this, we will build a very simple callback function outside of main() that defines behavior if an external interrupt is received on GPIO pin 13, which ties to the blue user button on the dev board, as can be seen in the example schematic below.



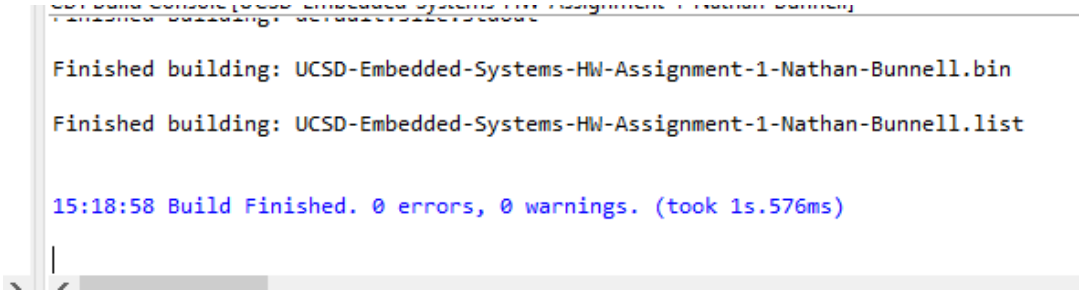
Again, the HAL_GPIO_TogglePin() function is used with LED2 defined as the GPIO port and pin to flip. With this complete, we can toggle LED2 at any point the system is running, including during the initial 3x blink.

```
171 /**********************************
172  * User Story 3 implementation begin:
173  *
174  *  Toggle LED2 on interrupt received
175  *  by pin 13, blue push button
176  *
177  **********************************/
178
179 /**
180   * @brief Define GPIO EXTI callback function
181   * @retval none
182   */
183 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
184 {
185     if (GPIO_Pin == GPIO_PIN_13)
186     {
187         HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
188     }
189 }
190
191 /**********************************
192  * User Story 3 implementation end
193  *
194  **********************************/
195
```

Nathan Bunnell
Embedded Controller Programming with C
ECE-40293
Student ID: U08895857

Date: 4/12/2021

Now, with all three stories defined in code, we can build and flash our project file to the target and ensure that there are no compilation errors or issues with running the code. Of note: my typical development workflow is to host all source files on a network share on a Linux server and use VSCode as an editor/ SSH client to manage any builds. In setting this project up from my Windows environment to what would be the normal location within my network, I received build errors from the compiler saying that it couldn't find the linker script files, even though they were visible in the project and editor. I was able to work around this by moving the project to a location on my C:\ drive but still want to look into it further.



With the build complete, the program also flashed without issue and the hardware reacted as expected from the story definitions.