ECE 6110 : Quiz 1

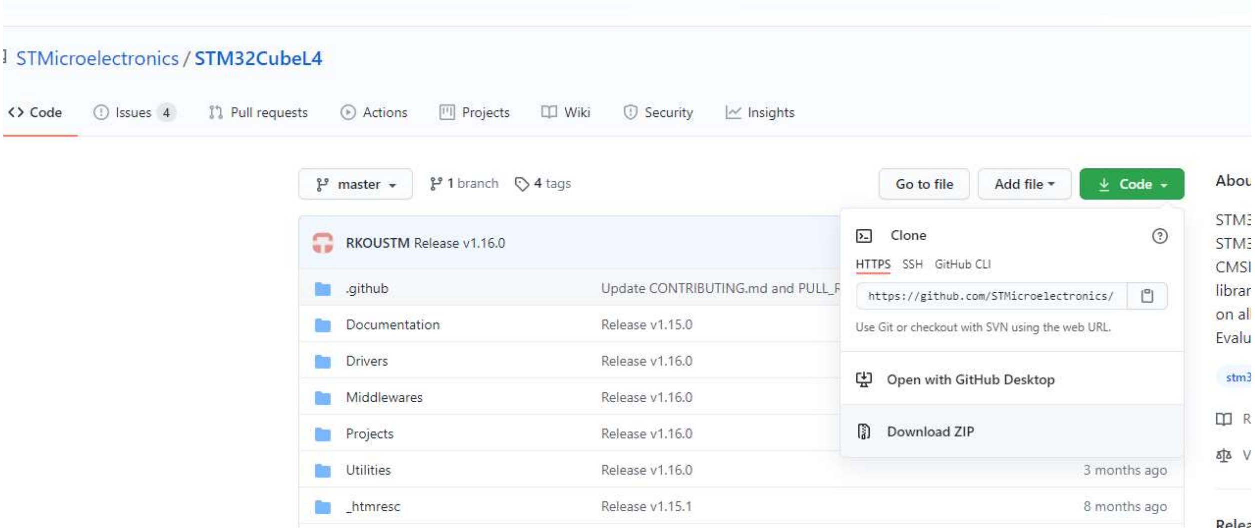
Michael Coultis

Process:

1. Read sensor documentation (no memory map?)
2. ST suggests use of WINDOWS API (hell no)
3. Sensor API suggested by ST.
4. Attempted to use API
    a. Blocked- HAL_master_transmit returning error
    b. Due to size of API substituting for i2c_mem_read/i2c_mem_write not practical
5. Asked peers for help:
    a. Working sensor from ST: https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-mcu-mpu-packages/stm32cubel4.html
    b. Project: https://github.com/STMicroelectronics/STM32CubeL4
    c. Import Method: https://youtu.be/MvOd1h-MGjM
6. SOLUTION: Simplify example code and port to a STMCubeIDE project
    a. *THAT WAY THE MICROCONTROLLER CAN BE RECONFIGUED IN THE CONFIGURATION TOOL IN CUBE IDE!!!*
7. Result: https://www.youtube.com/watch?v=DtzsljcPIQI

Import Method – Default ST Project:

1. Download from GitHub and extract .zip
https://github.com/STMicroelectronics/STM32CubeL4



STMicroelectronics / STM32CubeL4

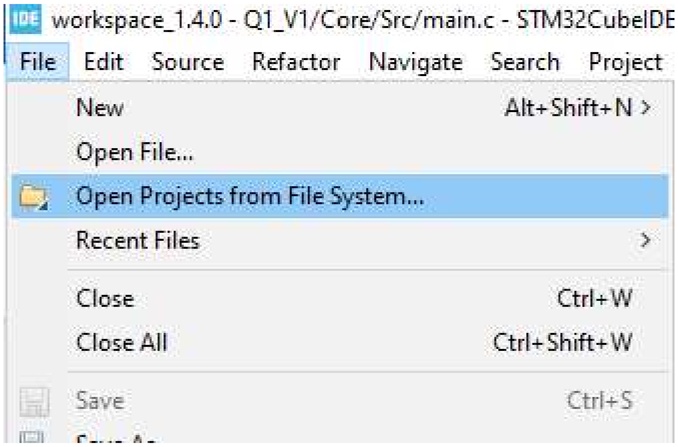
RKOUSTM Release v1.16.0

.github	Update CONTRIBUTING.md and PULL_R	
Documentation	Release v1.15.0	
Drivers	Release v1.16.0	
Middlewares	Release v1.16.0	
Projects	Release v1.16.0	
Utilities	Release v1.16.0	3 months ago
_htmresc	Release v1.15.1	8 months ago

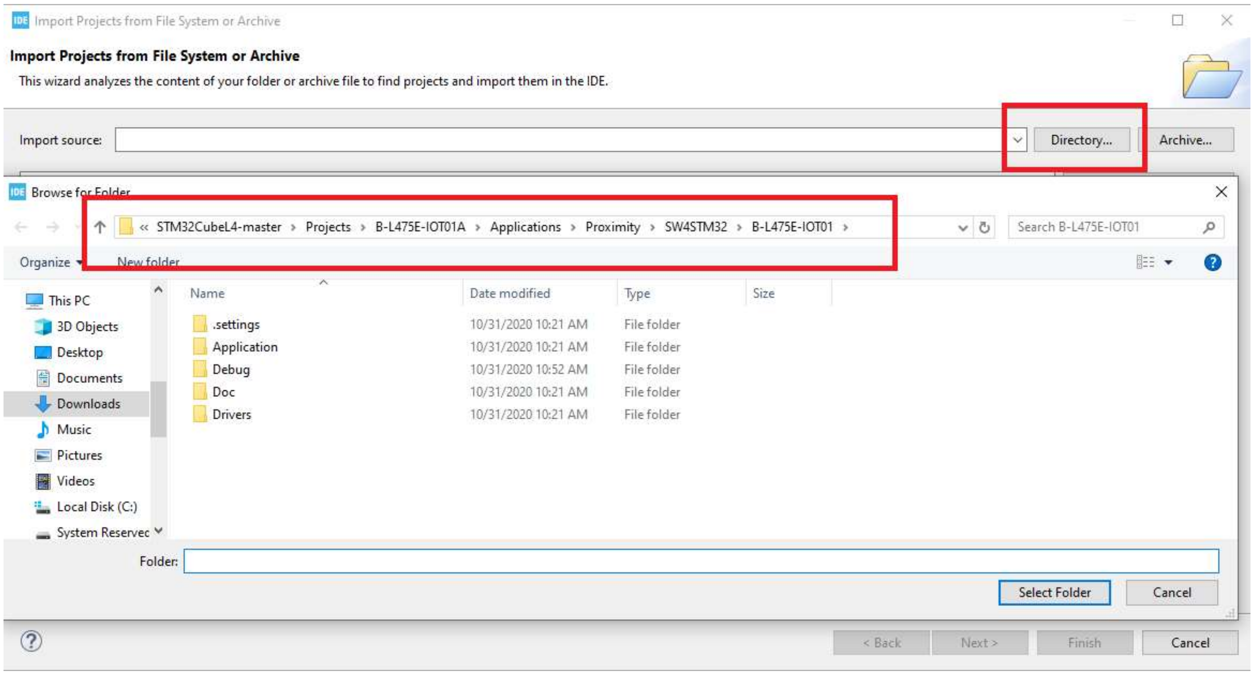
https://github.com/STMicroelectronics/

Use Git or checkout with SVN using the web URL.

2. Open Project:



3. Select Directory:



Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

« STM32CubeL4-master › Projects › B-L475E-IOT01A › Applications › Proximity › SW4STM32 › B-L475E-IOT01 ›

Search B-L475E-IOT01

Name	Date modified	Type	Size
.settings	10/31/2020 10:21 AM	File folder	
Application	10/31/2020 10:21 AM	File folder	
Debug	10/31/2020 10:52 AM	File folder	
Doc	10/31/2020 10:21 AM	File folder	
Drivers	10/31/2020 10:21 AM	File folder	

4. Build
5. Run

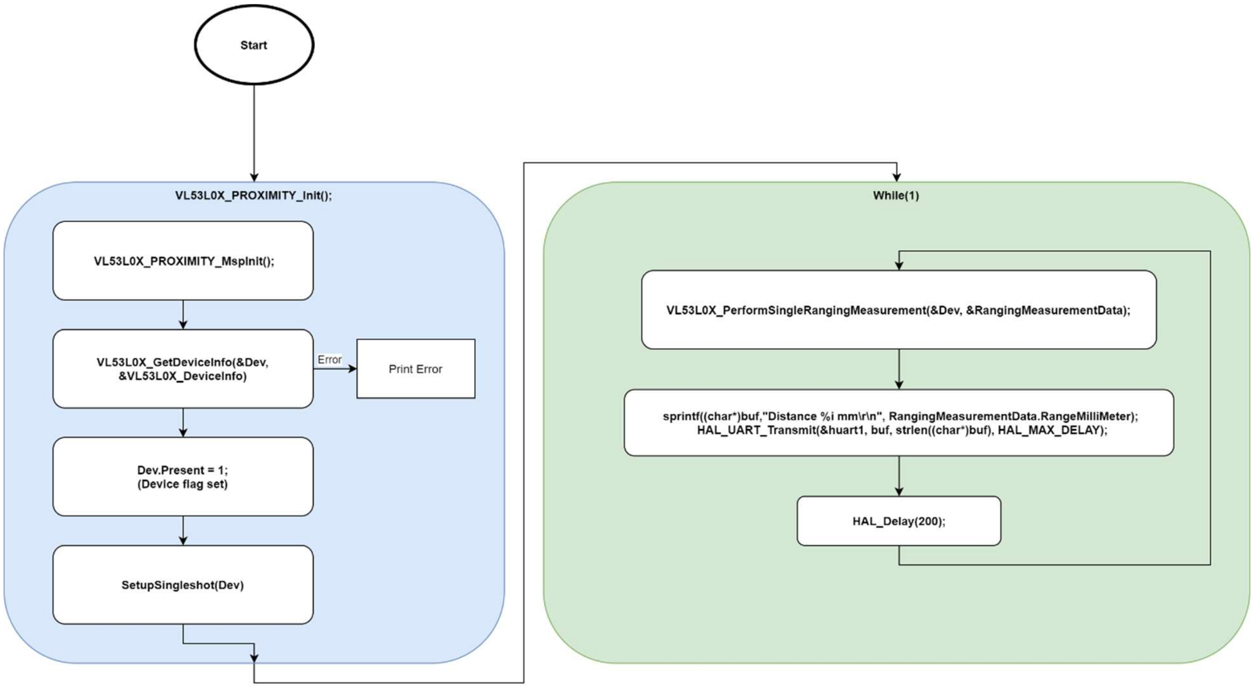
Simplified Example – STCubeIDE Generated project:

1. Generate B-L475E-IOT1 board project.
2. Dragged all relevant libraries to the Project Explorer file tree.
3. I eliminated the “main.h” used in the example.

    a. Copied all headers to my main.c
    b. Copied all Defines to my main.c

4. Eliminated VL53L0X_PROXIMITY_GetDistance(void)
    a. Not an API function.
    b. Doesn’t help me learn how this project works.
5. Kept VL53L0X_PROXIMITY_Init(void)
    a. Simple implementation.
    b. Actually checks to see if the sensor is good to go.
6. Reconfigure main () to poll the sensor repeatedly and output result as serial

Project Diagram:


Start
VL53L0X_PROXIMITY_Init();
VL53L0X_PROXIMITY_MspInit();
VL53L0X_GetDeviceInfo(&Dev, &VL53L0X_DeviceInfo)
Error
Print Error
Dev.Present = 1; (Device flag set)
SetupSingleshot(Dev)
While(1)
VL53L0X_PerformSingleRangingMeasurement(&Dev, &RangingMeasurementData);
sprintf((char*)buf,"Distance %i mm\r\n", RangingMeasurementData.RangeMilliMeter); HAL_UART_Transmit(&huart1, buf, strlen((char*)buf), HAL_MAX_DELAY);
HAL_Delay(200);

Resulting main():

```
int main(void)
{
  /* USER CODE BEGIN 1 */
	uint8_t buf[50]; //Local UART storage buffer
	VL53L0X_RangingMeasurementData_t RangingMeasurementData; // A struct for
returning time of flight sensor data.
  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DFSDM1_Init();
  MX_I2C2_Init();
  MX_QUADSPI_Init();
  MX_SPI3_Init();
  MX_USART1_UART_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  /* USER CODE BEGIN 2 */
  //Fire up the time of flight sensor.
  VL53L0X_PROXIMITY_Init();

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
         //Get sensor data
         VL53L0X_PerformSingleRangingMeasurement(&Dev, &RangingMeasurementData);
         //Output distance in mm to serial port
         sprintf((char*)buf,"Distance %i mm\r\n",
RangingMeasurementData.RangeMilliMeter);
         HAL_UART_Transmit(&huart1, buf, strlen((char*)buf), HAL_MAX_DELAY);
         HAL_Delay(200);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```