Code Description of the STM32F0 LSPCB1x PCB

Introduction

The STM32F0 LSPCB1x pcb integrates with the stm32 microcontroller to perfom the necessary operations using the data retrieved from the sensor in the PCB. In this code reference document the programming language c is used, however the PCB isn't only limited to c, c++ can also be used. Following are the fucntions used in the program, with details on how each plays into the whole fundumental program.

System Clock Configuration:

```
void SystemClock_Config(void);
```

This function initializes the Reset and Clock Control(RCC) registers of the STM32 microcontroller. The function initializes the RCC oscillators in accordance with the user specified parameters. Shown below is code snippet, showing the latter.

```
void SystemClock_Config(void)
        {…
        RCC_OscInitTypeDef RCC_OscInitStruct = {0};
        RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
        …}
```

The function also goes on to handle initialization of the CPU, AHB and APB buses' clocks, to which a user may modify to work best for their application.

GPIO Pins Configuration:

```
static void MX_GPIO_Init(void);
```

This function initialized the STM32 GPIO pins, to which the user may add on other pins should there be a need of some GPIO pins. The code snippet below shows how to initialize and use a GPIO pin by toggling the STM32 blue LED, but before that RCC clocks have to be enabled for the appropriate GPIOs.

```
static void MX_GPIO_Init(void)
        {…
        //enabling a clock
        __HAL_RCC_GPIOC_CLK_ENABLE();
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_8, GPIO_PIN_RESET
        …}
```

USART2 Initialization:

```
static void MX_USART2_UART_Init(void);
```

This function handles the initialization of the USB to serial converter. A FT232RL 3.3V / 5.5V FTDI USB is integrated with the PCB for this purpose. The function sets multiple parameters for the communication link, from Baud rate to Word Length and other parameters involved. Below is code snippet showing how some of the parameters are set in the function:

```
static void MX_USART2_UART_Init(void)
      {…
      huart2.Instance = USART2;
      huart2.Init.BaudRate = 9600;
      huart2.Init.WordLength = UART_WORDLENGTH_8B;
      …}
```

I2C1 Initialization:

```
static void MX_I2C1_Init(void);
```

The PCB uses the I2C communication protocol, where data is transmitted bit by bit along the SDA wire on a shared clock signal. This function initializes the I2C protocol by setting the appropriate parameters demonstrated below.

```
static void MX_I2C1_Init(void)
      {…
      hi2c1.Instance = I2C1;
      hi2c1.Init.Timing = 0x2000090E;
      hi2c1.Init.OwnAddress1 = 0;
      …}
```

Error Detection and Handler

```
void Error_Handler(void);
```

This function is the failsafe of the program for the anticipated errors that might occur, it is the function that executes when an error is detected in the program.

```
void Error_Handler(void)
      {…
      //user may add code here.
      __disable_irq();


      …}
```

This function can be called under multiple functions if an error is detected, below is a demonstration where the function is called to perform the user desired instruction when an error is detected under USART2 initialization

```
static void MX_USART2_UART_Init(void)
      {…
      if (HAL_UART_Init(&huart2) != HAL_OK)
            {
                  Error_Handler();
            }
      …}
```

The Main Function

```
int main(void);
```

The main function in the main.c file is the execution function of all the used functions and files for the program. From the initializations of the other functions, to the data processing where data is retrieved from the sensor, written to the EEPROM and sent to the program via the USART serial com.

All the initializing functions are called in main before any other processing is done so that all the necessary GPIOs and clocks are set. The user may extend under the main function to add other functions they may wish to add. The main function is created as follows:

```
int main(void)
{
  //All the initializing functions are called
SystemClock_Config();
MX_GPIO_Init(void);
```

Code Reference

..

```
Data processing between the computer, EEPROM and the PCB sensors.
```

//And the user may add their own code below.