## **GPIO\_INPUT**

Main.c

|  |
| --- |
|  |

BML\_GPIO.c

(Double click to open)

|  |
| --- |
|  |

BML\_GPIO.h

(Double click to open)

|  |
| --- |
|  |

***GPIO Functions:-***

***Serial Printing Functions***

* uart\_print\_config()
* print(char \*msg, ...)
* gpio\_altfn(GPIO\_TypeDef \*port, uint32\_t pinNumber,uint32\_t alternate\_function)

***How to use Functions:-***

* Call the uart\_print\_config() function in main() to configure the UART for serial printing at 9600 baud rate.
* Then, simply use the print function to print any string.
* There is some problem in print function in printing numbers, so using different code my senior gave me. It is in header file call ‘APP\_DEBUG’. Use it print variables.

***How the function works?***

1. **UART PRINT CONFIG Function -**

* It enables the UART clock first which is in APB1ENR register for UART2.
* Then we have to configure the gpio to RX and TX of UART. Here, we are using a different mode which we did not used till now.
* In the mode parameter, we are slecting the ALT\_MODE or alternate mode.
* The alternate mode definations are in the header file and looking at the Pin number we can configure them according to us.
* Next we jump to UART registers, UART2 is connected to the USB port.
* First we have to select the baud rate using this formula

**clkPer / (baudR x 16) = 48MHz / (9600 x 16) = 312.5 = 1388**

* Our UART clock is at 8Mhz so we are using value 0x341 we got from formula. You can change according to your clock in the uart\_print\_config function
* Then, in the UART control register 1, the TX, RX interrupt are enables and UART is enabled using USART\_CR1\_UE.
* As our UART is configured, we will move to the print function. In the print function no need to understand the code till vsprint as it is just converting the message to string with %d,%c etc in consideration.
* What we are doing to send the data using UART is filling the UART->TDM(Transmission Data register) with the data we want to send.
* After sending it, we wait till the transmission interrupt bit is not set means, transmission is still not complete. We are checking it but ANDing Interrupt and Status register with the TXIE bit.

1. **GPIO\_altfn function –**

* To configure the gpio to output mode, first configure the Pin to the Alternate mode using gpio\_config() function.
* Next, call the gpio\_altfn(), it takes 3 parameters and the 3rd parameter is what Alternate function you want?
* Alternate function are inside the datasheet. On one Pin there are upto 7 different alternate functions.
* The alternate function is selected by setting the AFR register of GPIO. To configure each pin, 4 pair of bits have to be configured. From ‘0’ to ‘7’ will configure the AF0 to AF7.
* There are 2 registers High and Low to select the AFSEL pin.
* As we need to shift 4 bits to configure the pins, the pinNumber is multiplied by 4 and then shifted.
* The definations are in header file for different alternate funtions.