

# Comparator Programming

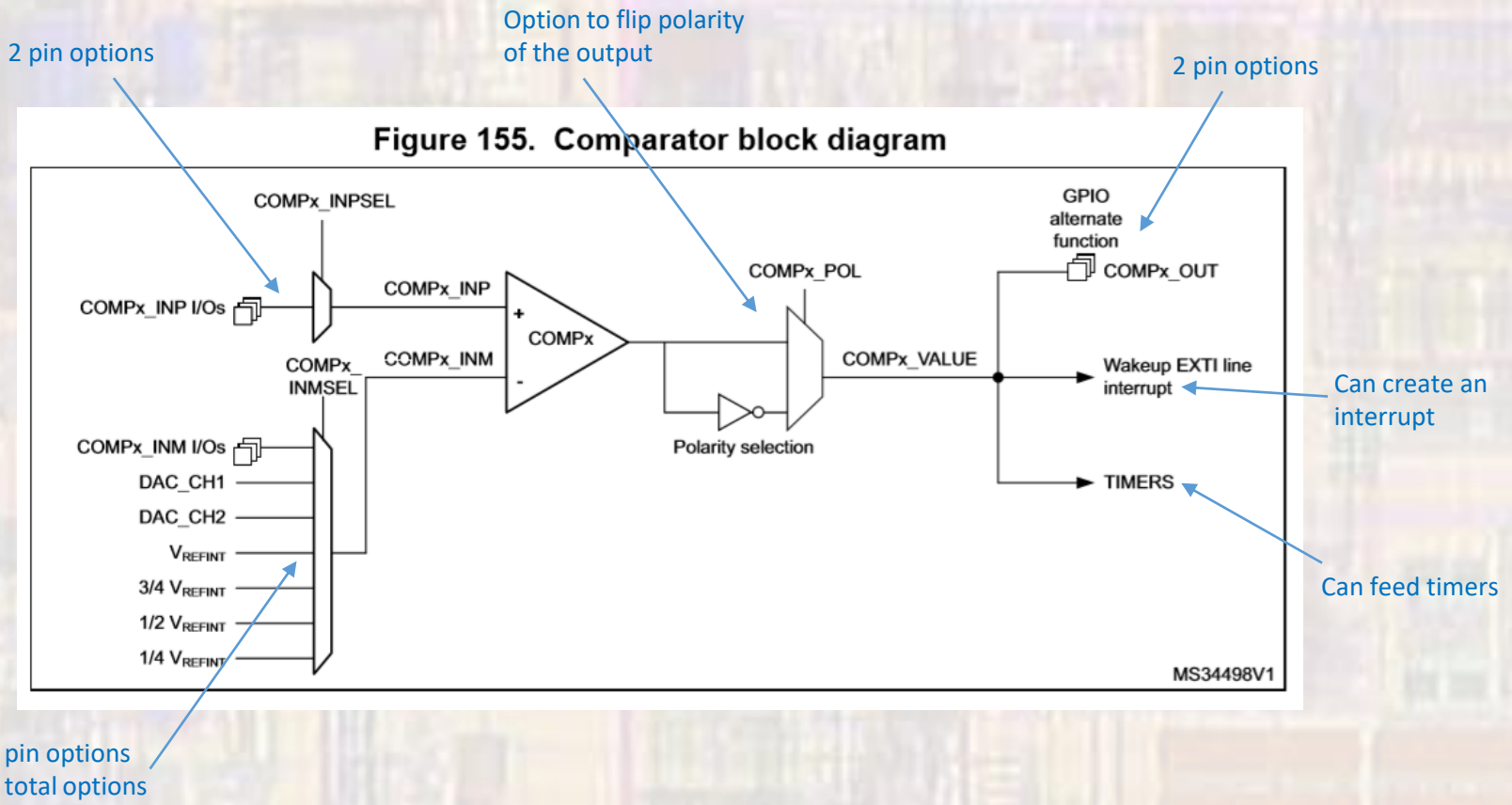
Last updated 6/23/21

# Comparator Programming

- Comparators
  - Nucleo-L476RG has two high speed comparators
    - Mbed **does not** support the comparators
  - We will need to write our own low-level code

# Comparator Programming

- Comparator Configuration



# Comparator Programming

## • Comparator Connections - Inputs

- Nucleo-L476RG uses programmable I/O selection to choose the INP and INM connections
- Multiple steps are required to enable these connections
  - Port(s) must be enabled (clocks enabled)
  - Pins must be selected as Analog Inputs (using the Pin I/O configuration registers)
  - Pins must be selected in the COMPx registers

Table 143. COMP1 input plus assignment

COMP1_INP	COMP1_INPSEL
PC5	0
PB2	1

Table 144. COMP1 input minus assignment

COMP1_INM	COMP1_INMSEL[2:0]
$\frac{1}{4} V_{REFINT}$	000
$\frac{1}{2} V_{REFINT}$	001
$\frac{3}{4} V_{REFINT}$	010
$V_{REFINT}$	011
DAC Channel1	100
DAC Channel2	101
PB1	110
PC4	111

Table 145. COMP2 input plus assignment

COMP2_INP	COMP2_INPSEL
PB4	0
PB6	1

Table 146. COMP2 input minus assignment

COMP2_INM	COMP2_INMSEL[2:0]
$\frac{1}{4} V_{REFINT}$	000
$\frac{1}{2} V_{REFINT}$	001
$\frac{3}{4} V_{REFINT}$	010
$V_{REFINT}$	011
DAC Channel1	100
DAC Channel2	101
PB3	110
PB7	111

# Comparator Programming

## • Comparator Connections - Output

- Nucleo-L476RG uses programmable I/O selection for the OUT connection
- Multiple steps are required to enable these connections
  - Port must be enabled (clocks enabled)
  - Pins must be configured to allow Alternate Function
  - The COMPx output must be selected as the alternate function

Table 18. Alternate function AF8 to AF15<sup>(1)</sup> (continued)

Port	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15	
	UART4, UART5, LPUART1	CAN1, TSC	OTG_FS, QUADSPI	LCD	SDMMC1_COMP1, COMP2_FMC, SWPMI1	SAI1, SAI2	TIM2, TIM15, TIM16, TIM17, LPTIM2	EVENTOUT	
PB0	-	-	QUADSPI_BK1_IO1	LCD_SEG5	COMP1_OUT	-	-	EVENTOUT	
PB1	-	-	QUADSPI_BK1_IO0	LCD_SEG6	-	-	LPTIM2_IN1	EVENTOUT	
PB2	-	-	-	-	-	-	-	EVENTOUT	
PB3	-	-	-	LCD_SEG7	-	SAI1_SCK_B	-	EVENTOUT	
PB4	UART5_RTS_DE	TSC_G2_IO1	-	LCD_SEG8	-	SAI1_MCLK_B	TIM17_BKIN	EVENTOUT	
PB5	UART5_CTS	TSC_G2_IO2	-	LCD_SEG9	COMP2_OUT	SAI1_SD_B	TIM16_BKIN	EVENTOUT	
PB6	-	TSC_G2_IO3	-	-	TIM8_BKIN2_COMP2	SAI1_FS_B	TIM16_CH1N	EVENTOUT	
PB7	UART4_CTS	TSC_G2_IO4	-	LCD_SEG21	FMC_NL	TIM8_BKIN_COMP1	TIM17_CH1N	EVENTOUT	
Port B	PB8	-	CAN1_RX	-	LCD_SEG16	SDMMC1_D4	SAI1_MCLK_A	TIM16_CH1	EVENTOUT
	PB9	-	CAN1_TX	-	LCD_COM3	SDMMC1_D5	SAI1_FS_A	TIM17_CH1	EVENTOUT
	PB10	LPUART1_RX	-	QUADSPI_CLK	LCD_SEG10	COMP1_OUT	SAI1_SCK_A	-	EVENTOUT
	PB11	LPUART1_TX	-	QUADSPI_NCS	LCD_SEG11	COMP2_OUT	-	-	EVENTOUT
	PB12	LPUART1_RTS_DE	TSC_G1_IO1	-	LCD_SEG12	SWPMI1_IO	SAI2_FS_A	TIM15_BKIN	EVENTOUT
	PB13	LPUART1_CTS	TSC_G1_IO2	-	LCD_SEG13	SWPMI1_TX	SAI2_SCK_A	TIM15_CH1N	EVENTOUT
	PB14	-	TSC_G1_IO3	-	LCD_SEG14	SWPMI1_RX	SAI2_MCLK_A	TIM15_CH1	EVENTOUT
	PB15	-	TSC_G1_IO4	-	LCD_SEG15	SWPMI1_SUSPEND	SAI2_SD_A	TIM15_CH2	EVENTOUT



# Comparator Programming

- Simple example
- Comparator setup

```
////////////////////////////////////
//
// comp_class_ex_1 project
//
// created 6/4/21 by tj
// rev 0
//
////////////////////////////////////
//
// Comparator example file for class
//
// shows basic comparator operation
// also shows direct register access
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

int main(void){
    setbuf(stdout, NULL); // disable buffering

    // splash
    printf("comp_class_ex_1 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
        MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // Using PC5 as the + input, PC4 as the - input, and PB0 as the output

    // Must enable PortC to access the comparator inputs
    // RCC_AHB2ENR bit 2 for port C
    RCC->AHB2ENR |= 0x04;
    // Set inP(PC4) and inM(PC5) to analog in mode
    GPIOC->MODER |= 0x00000F00;
```

```
    // Must enable PortB AND enable it's alternate function to access the
    // comparator output
    // RCC_AHB2ENR bit 1 for port B
    RCC->AHB2ENR |= 0x02;
    // Set PB0 to alternate function
    // 10 to bits 1-0
    GPIOB->MODER = (GPIOB->MODER | 0x00000002) & ~0x00000001;
    // Select alternate function 12 - comparator output
    // 1100 to bits 3-0
    // NOTE: documentation calls this AFRL
    GPIOB->AFR[0] = 0x0000000C;

    // Enable comparator peripheral clock
    // APB2 (high speed APB)
    // RCC_APB2ENR, 1 to bit 0 to enable
    RCC->APB2ENR |= 0x00000001;

    // Setup COMP1 values
    // INP - PC5, 0 to bit 7
    // INM - PC4, 111 to bits 6-4
    // On, 1 to bit 0
    // all others 0
    COMP1->CSR = 0x00000071;

    while(1){
        // Read from the comparator output value
        // This is what is sent to the output pin
        // output register, bit 30
        printf("%x\n", (COMP1->CSR & 0x40000000) && 1);

        wait_us(10000);
    } // end while

    return 0;
} // end main
```



