

ENGR-2350

Embedded Control

**Timer Capture/Compare
Registers (CCRs)**

ENGR-2350

Embedded Control

Timers: Compare + PWM

Controlling non-Binary Outputs

Devices that are more than just “On” and “Off”

- DC Motors (Speed Control)
- Servo position
- Lighting Brightness
- Audio outputs (crude)
- Communication
- ...

Motors: Speed Control

$$P=VI$$

To control speed, we need a way to control total power going to the motor:

- **Voltage control:** produce an analog voltage to drive motor
- **Current control:** similar to the above, but current
- **Time-based control** of average power consumed

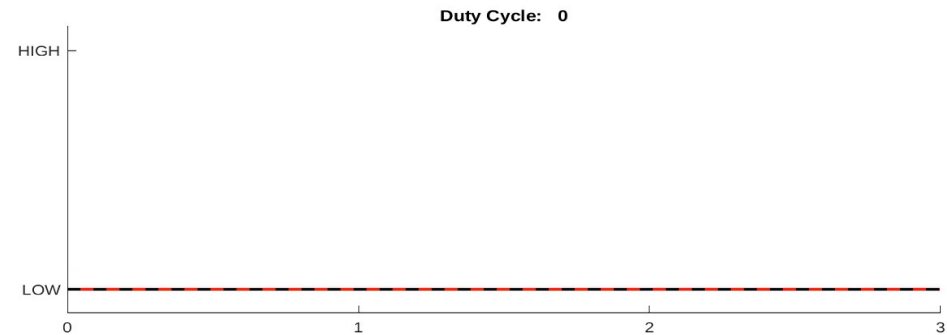
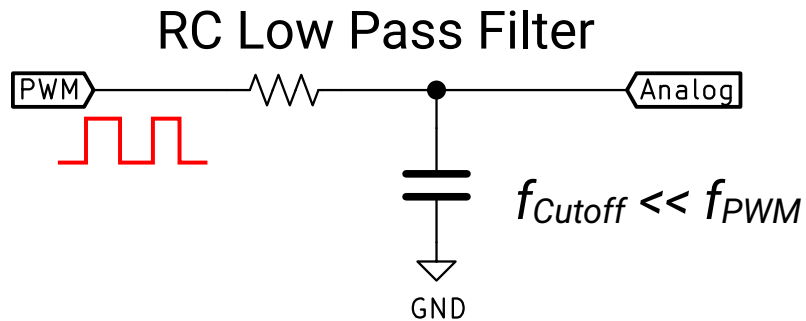
Motors: Speed Control

- Voltage control: DAC (Digital-to-Analog Converter)
 - We have one, but not using it for this.
- Current control: **Current DAC**
 - Not sure if any microcontroller has one...
- Time-based control
 - **Pulse Width Modulation (PWM)**

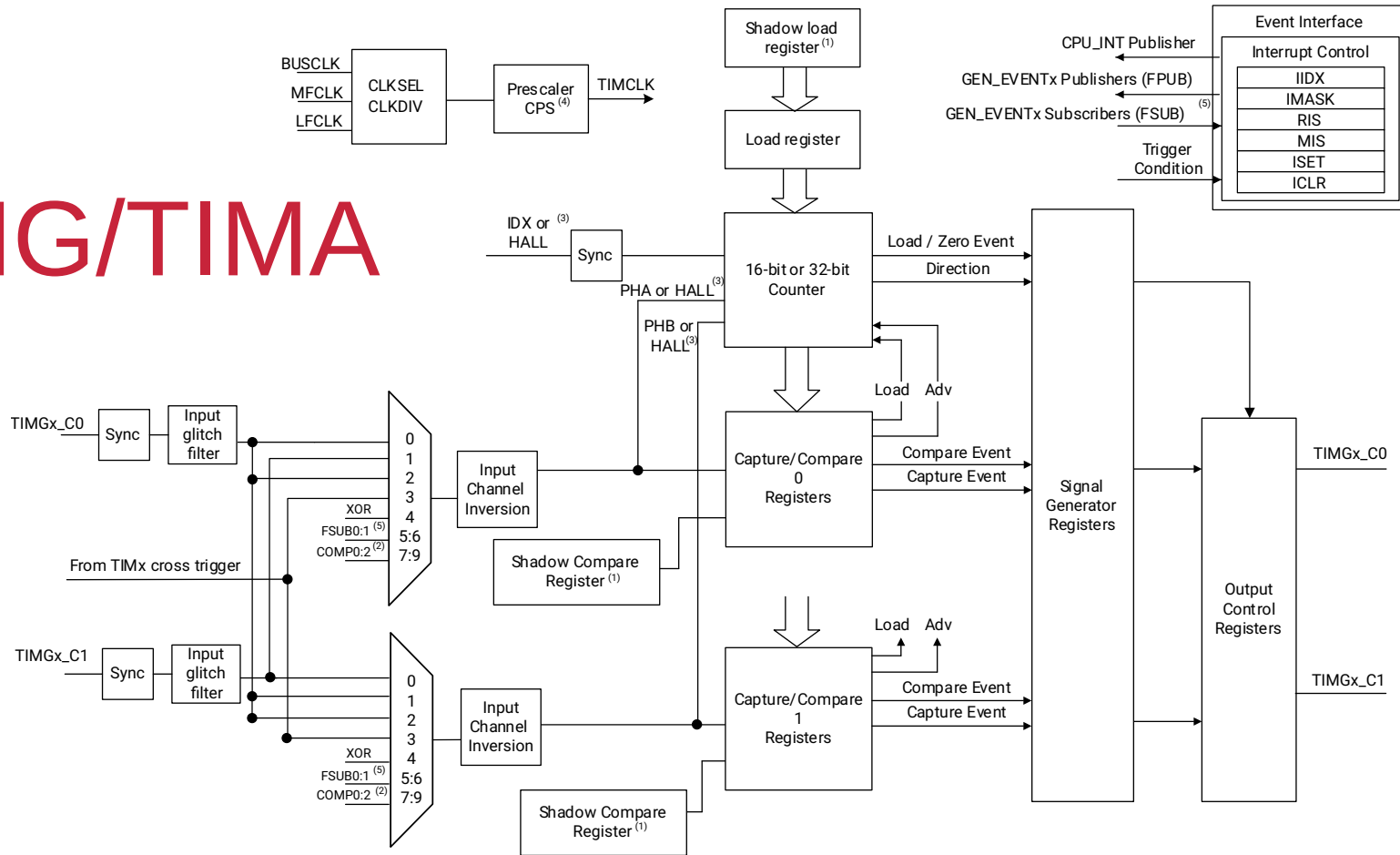
Pulse Width Modulation, PWM

PWM uses a simple GPIO to adjust **Average Power** by controlling on/off time ratio:

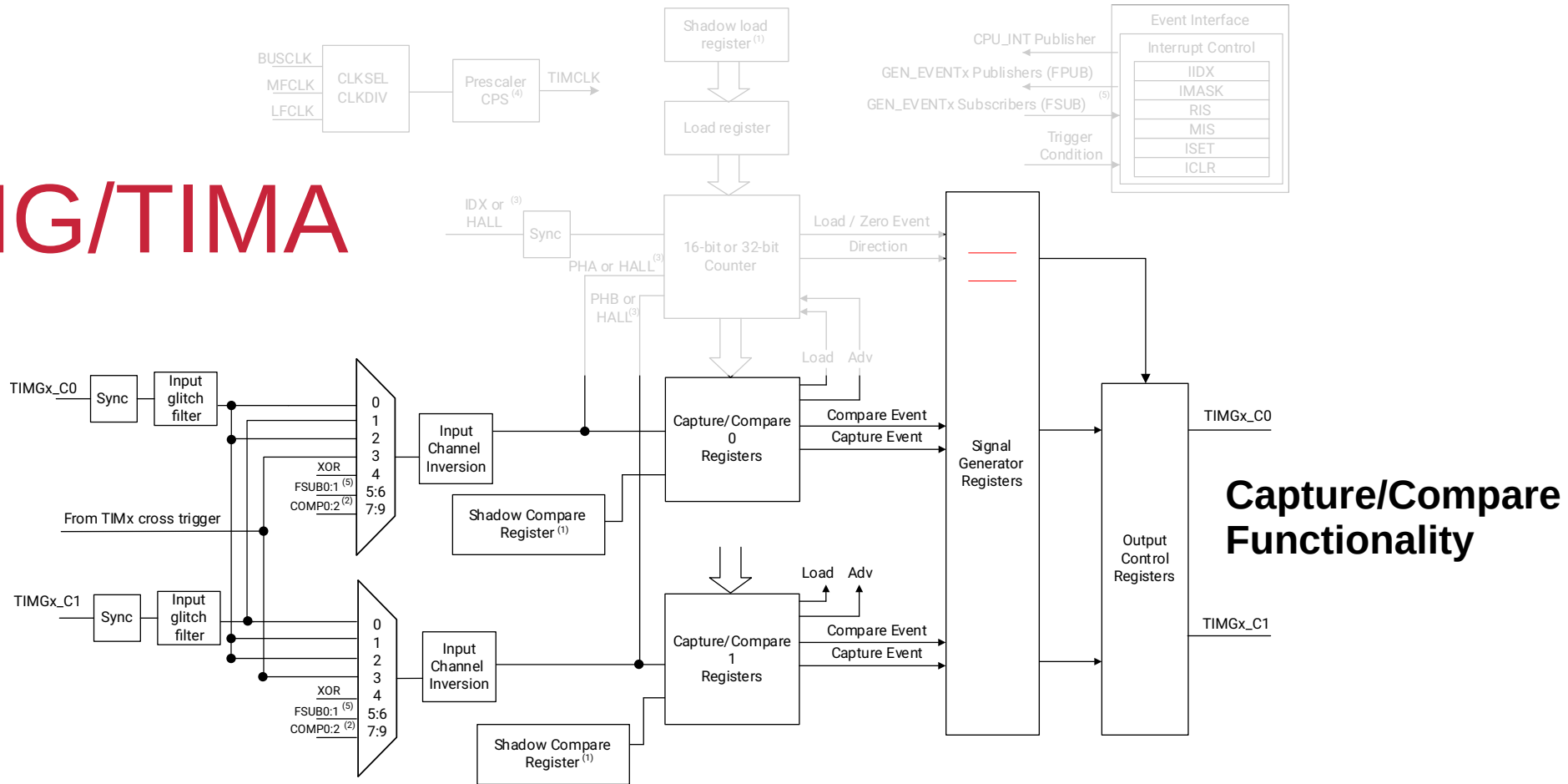
- PWM uses a **fixed frequency**



TIMG/TIMA



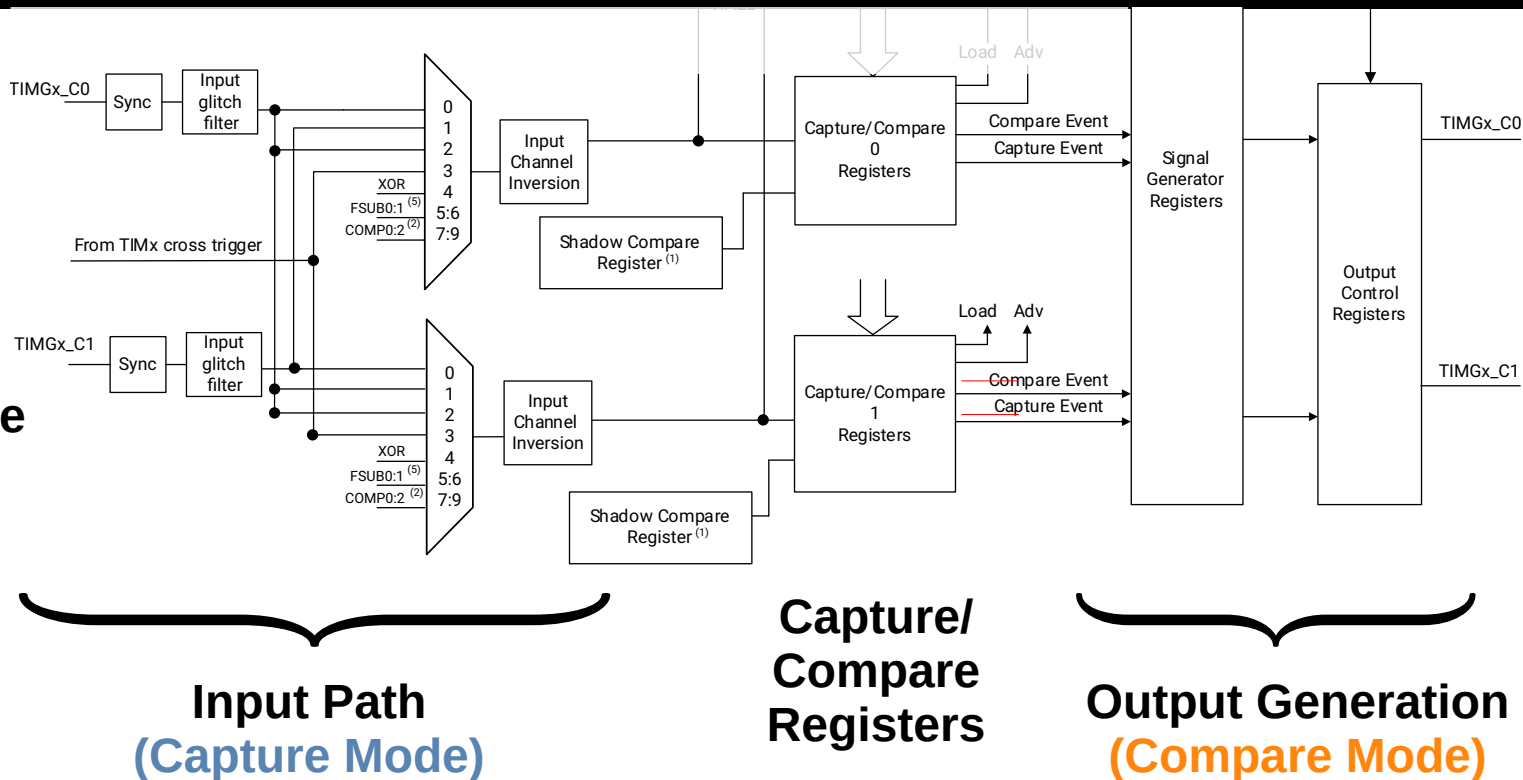
TIMG/TIMA



Capture/Compare Functionality

TIMG/ TIMA

Capture/Compare Functionality

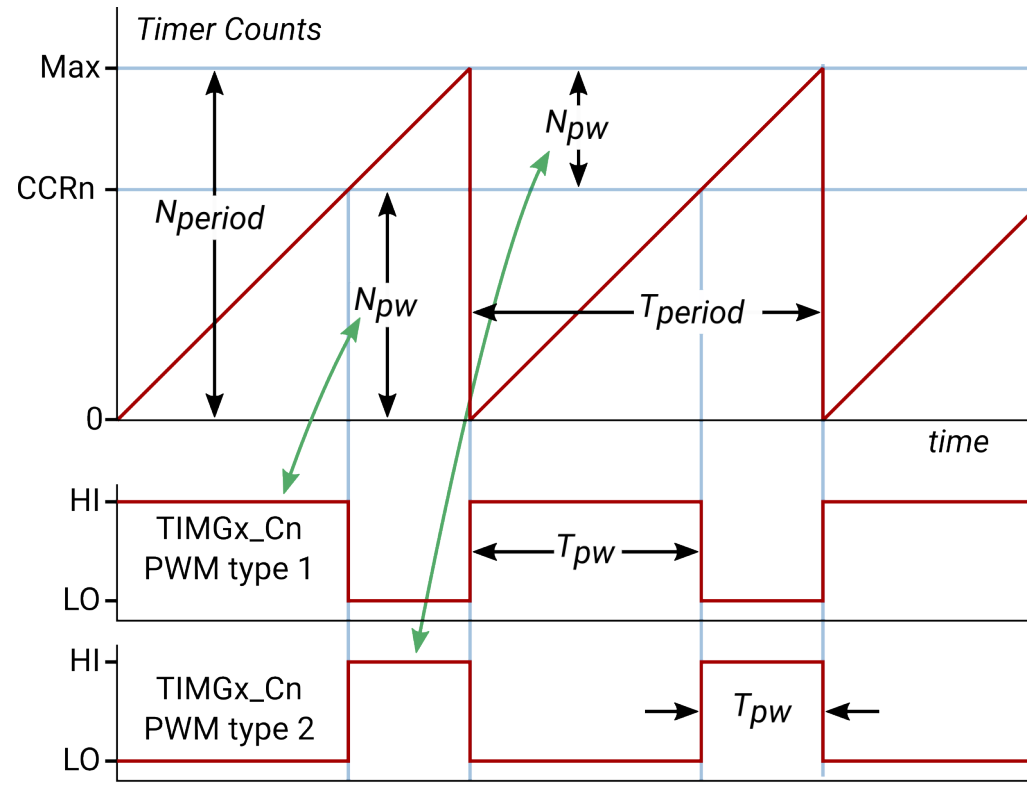


Pulse Width Modulation, PWM

Generating PWM from a Timer using **Compare Mode**

...this part on paper...

Pulse Width Modulation, PWM



Pulse Width Modulation, PWM

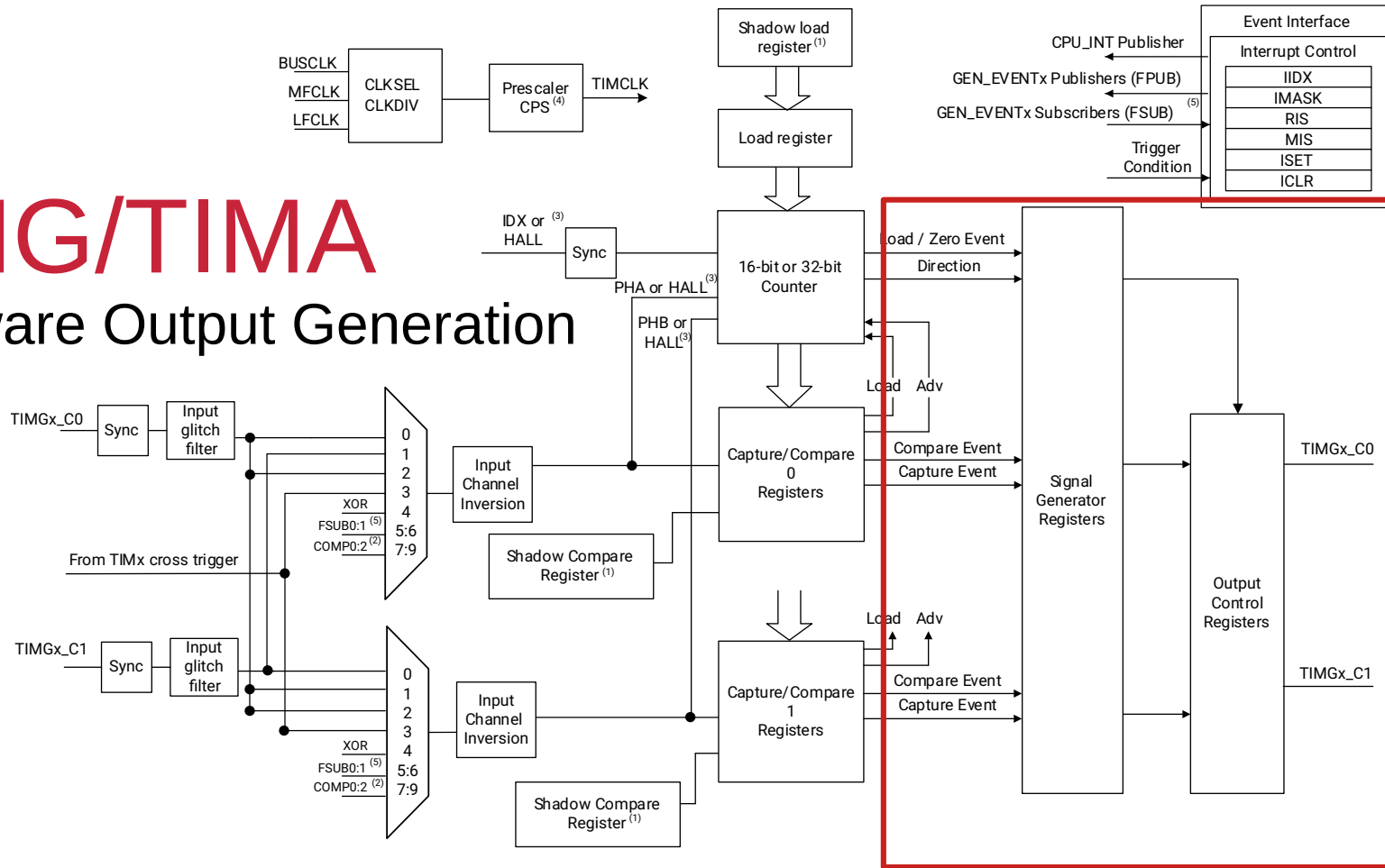
Sample code using CCR0 to produce intermediate interrupt:

```
void TIMG0_IRQHandler(){
    uint32_t active = Timers_getActiveInterrupt(TIMG0);
    if(active == TIMER_INTSRC_ZERO)
        GPIO_clearPins(GPIOB,GPIO_PIN16);
    else if(active == TIMER_INTSRC_CCR0_UP)
        GPIO_setPins(GPIOB,GPIO_PIN16);
}
```

Better ways to do it...

TIMG/TIMA

Hardware Output Generation



TIMG/TIMA: Signal Generator Regs

Very flexible in what it can do:

- Unique signals generated for each Capture/Compare Register, “CCR”
- Generated signals can be modified on:
 - **Timer Zero Event**
 - **Timer Load Event**
 - **Up-Counting Compare Event**
 - **Down-Counting Compare Event**
- Modifications can be one of: **NONE, SET, CLEAR, TOGGLE**

TIMG/TIMA: Signal Generator Regs

PWM Generation requires full signal period each timer reset

- One **rising edge** per reset (**SET** or **TOGGLE**)
- One **falling edge** per reset (**CLEAR** or **TOGGLE**)

Connect TIMGx_Cy to GPIO

- GPIO Alternate/Peripheral Functions
 - Pins can be typical Inputs/Outputs OR
 - Pins can be connected to modules to route signals out of the microcontroller
 - Examples [C8051F020](#) [STM32F769NI](#) [ATmega328P](#)

MSPM0G3507 GPIO Alternative Functions

Clearer table on website:

GPIO Peripheral Functions

Up to 7 alt. functions per-pin

Table 6-1. Pin Attributes

PINCMx	PIN NAME	SIGNAL NAMES		PIN NUMBER				IO STRUCTURE
		ANALOG	DIGITAL [PIN FUNCTION] (1)	64 LQFP	48 LQFPVQFN	32 VQFN	28 VSSOP	
1	PA0		UART0_TX [2] / I2C0_SDA [3] / TIMA0_C0 [4] / TIMA_FAL1 [5] / TIMG8_C1 [6] / FCC_IN [7] / (Default BSL I2C_SDA)	33	1	1	4	5V Tol. Open-Drain
2	PA1		UART0_RX [2] / I2C0_SCL [3] / TIMA0_C1 [4] / TIMA_FAL2 [5] / TIMG8_IDX [6] / TIMG8_C0 [7] / (Default BSL I2C_SCL)	34	2	2	5	5V Tol. Open-Drain
7	PA2	ROSC	TIMG8_C1 [2] / SPI0_CS0 [3] / TIMG7_C1 [4] / SPI1_CS0 [5]	42	8	6	9	Standard
8	PA3	LFXIN	TIMG8_C0 [2] / SPI0_CS1 [3] / UART2_CTS [4] / TIMA0_C2 [5] / COMP1_OUT [6] / TIMG7_C0 [7] / TIMA0_C1 [8] / I2C1_SDA [9]	43	9	7	10	Standard
9	PA4	LFXOUT	TIMG8_C1 [2] / SPI0_POCI [3] / UART2_RTS [4] / TIMA0_C3 [5] / LFCLK_IN [6] / TIMG7_C1 [7] / TIMA0_C1N [8] / I2C1_SCL [9]	44	10	8	11	Standard
10	PA5	HFXIN	TIMG8_C0 [2] / SPI0_PICO [3] / TIMA_FAL1 [4] / TIMG0_C0 [5] / TIMG6_C0 [6] / FCC_IN [7]	45	11	9	12	Standard
11	PA6	HFXOUT	TIMG8_C1 [2] / SPI0_SCK [3] / TIMA_FAL0 [4] / TIMG0_C1 [5] / HFCLK_IN [6] / TIMG6_C1 [7] / TIMA0_C2N [8]	46	12	10	13	Standard
14	PA7		COMP0_OUT [2] / CLK_OUT [3] / TIMG8_C0 [4] / TIMA0_C2 [5] / TIMG8_IDX [6] / TIMG7_C1 [7] / TIMA0_C1 [8]	49	13	11	-	Standard
19	PA8		UART1_TX [2] / SPI0_CS0 [3] / UART0_RTS [4] / TIMA0_C0 [5] / TIMA1_CON [6]	54	16	12	-	Standard
20	PA9		UART1_RX [2] / SPI0_PICO [3] / UART0_CTS [4] / TIMA0_C1 [5] / RTC_OUT [6] / TIMA0_CON [7] / TIMA1_C1N [8] / CLK_OUT [9]	55	17	13	14	High-Speed
21	PA10		UART0_TX [2] / SPI0_POCI [3] / I2C0_SDA [4] / TIMA1_C0 [5] / TIMG12_C0 [6] / TIMA0_C2 [7] / I2C1_SDA [8] / CLK_OUT [9] / (Default BSL UART_TX)	56	18	14	15	High-Drive
22	PA11		UART0_RX [2] / SPI0_SCK [3] / I2C0_SCL [4] / TIMA1_C1 [5] / COMP0_OUT [6] / TIMA0_C2N [7] / I2C1_SCL [8] / (Default BSL UART_RX)	57	19	15	16	High-Drive
24	PA12		UART3_CTS [2] / SPI0_SCK [3] / TIMG0_C0 [4] / CAN_TX	5	27	16	-	High-Speed

MSPM0G3507 Alternative Functions

- Determine either the Pin need OR Signal Needed
- Activate Alternate function through GPIO configuration
- Enable control of signal within module
 - (if necessary)

[Website: Alternate Functions](#)

PWM: Put it all together

- Select and configure GPIO alternate functions (2 motors)
- Configure Timer in UP (or DOWN) mode with desired PWM period
- Add **COMPARE MODE** CCR*n* support
 - Configure for TIMGx_C*n* to be PWM
- Turn ON Timer
- Change CCR*n* compare value to change PWM Duty Cycle.
- Turn OFF and CLEAR Timer to stop PWM
 - ...or set compare value to invalid number (> .period)

ENGR-2350

Embedded Control

Timers: Capture + Encoders

Data Acquisition

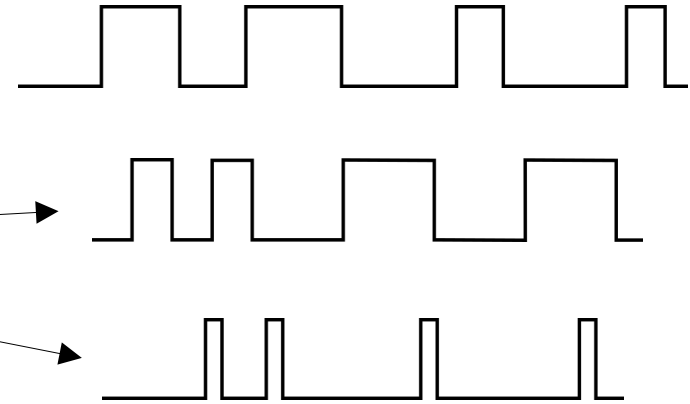
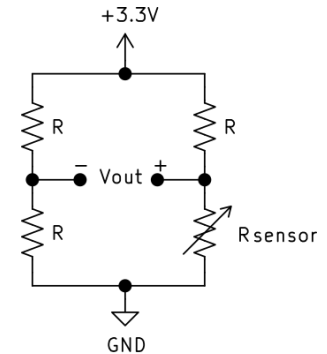
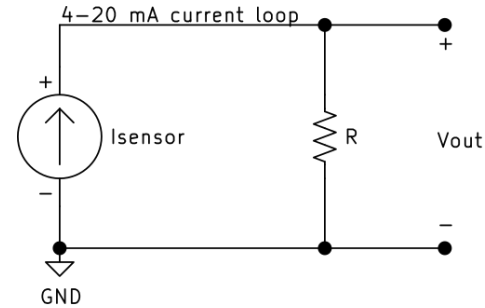
Need more than just plain Digital Inputs to perform most control applications (input → output):

- Temperature INPUT to control Heater
- Distance INPUT to control Drive power
- Speed INPUT to control Cruise control
- Light level INPUT to control screen brightness
- Pressure INPUT to control relief valves
- Tilt INPUT to control game character movement
- Etc.

Data Acquisition

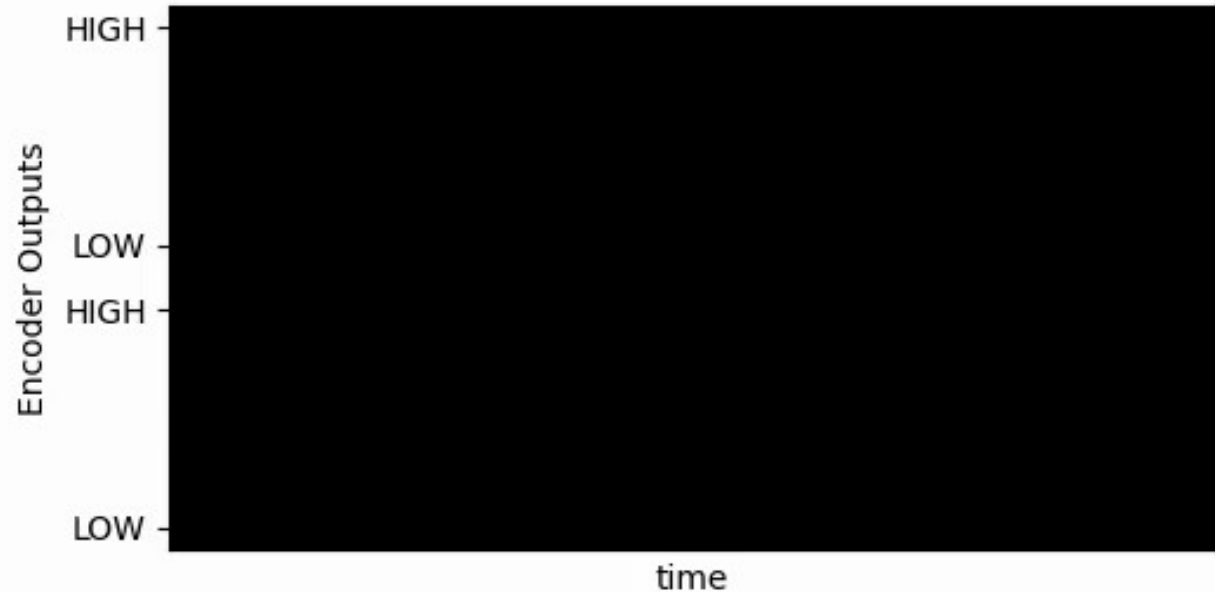
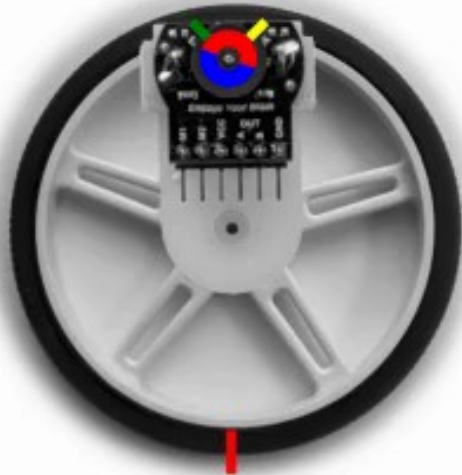
Additional Types of Input Data:

- **Analog Voltages and Currents:**
 - Sensors may output analog voltage directly.
 - Converted 4-20 mA current loops
 - Variable resistance + Bridges
- **Time/Frequency Based Signals:**
 - Pulse width modulated (PWM)
 - Frequency modulated (FM)
 - Delay based signals
- More...



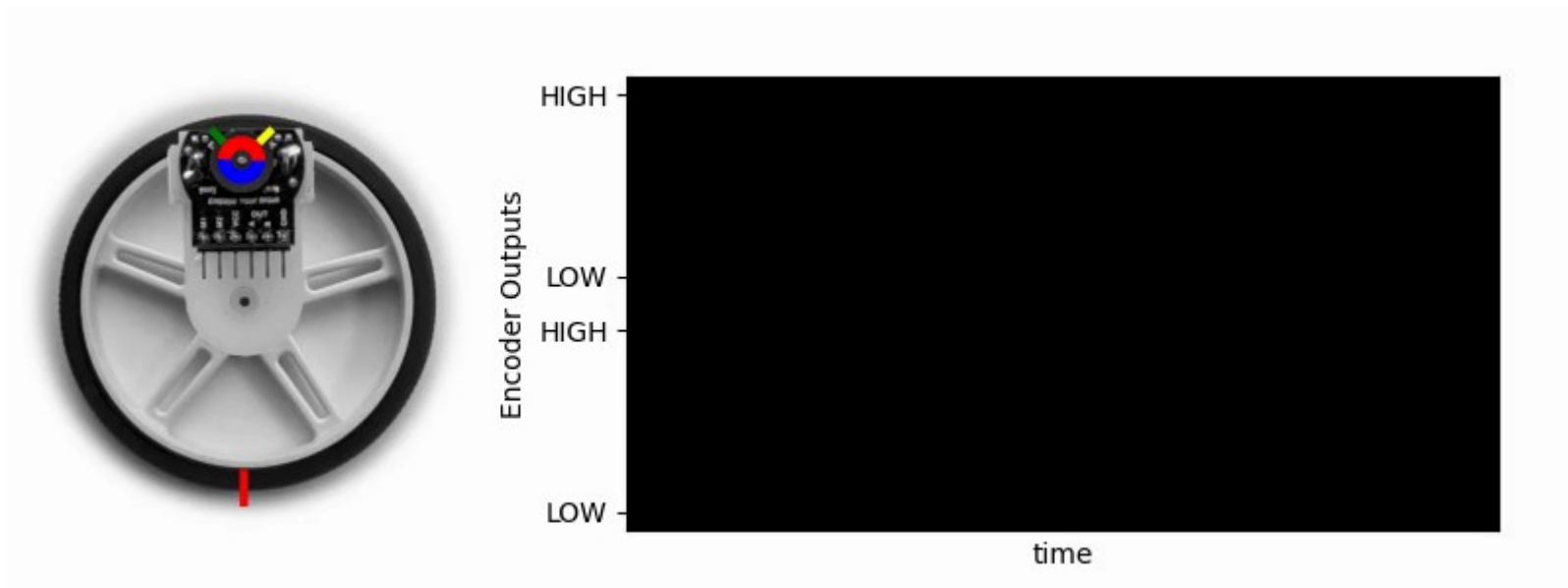
Frequency Modulated Example

Encoders: Shaft / Motor / Position / Etc.



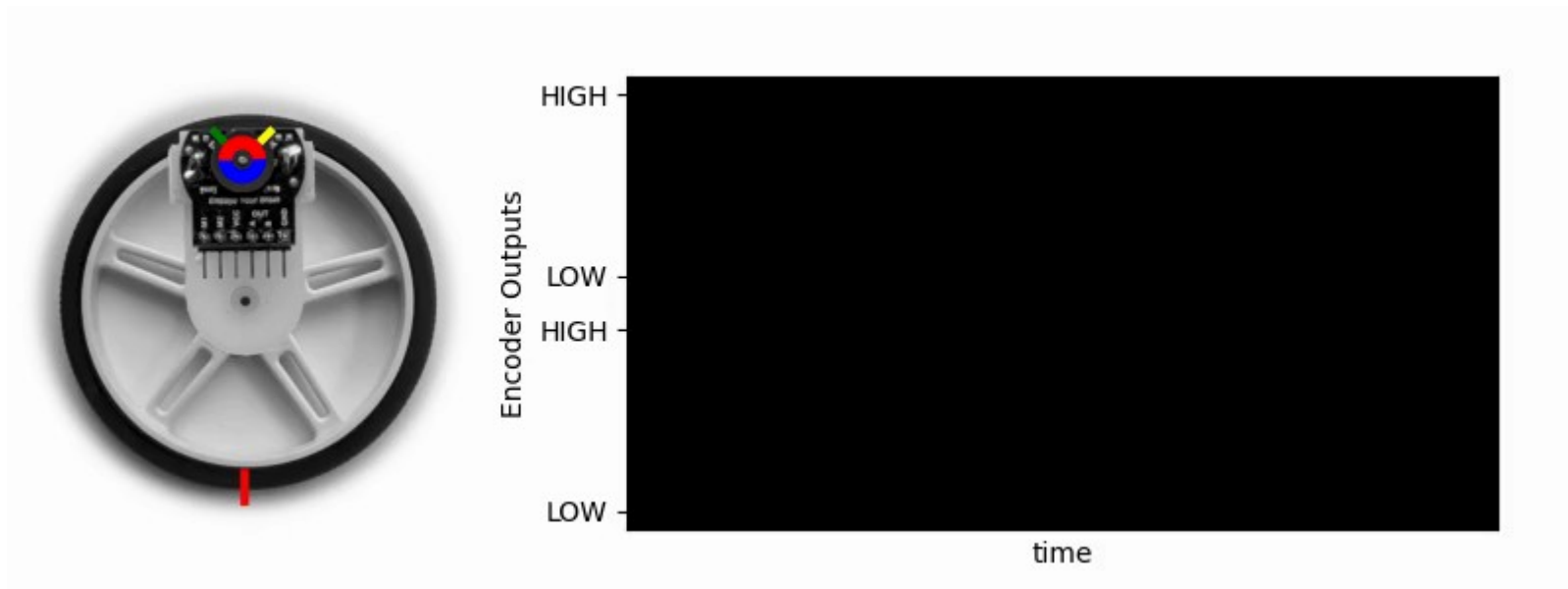
Frequency Modulated Example

Encoders: Detecting Forwards Rotation, two encoders



Frequency Modulated Example

Encoders: Detecting Reverse Rotation, two encoders



Data Acquisition: Methods

Analog Voltages and Currents:

- Next lecture...

Time/Frequency Based Signals:

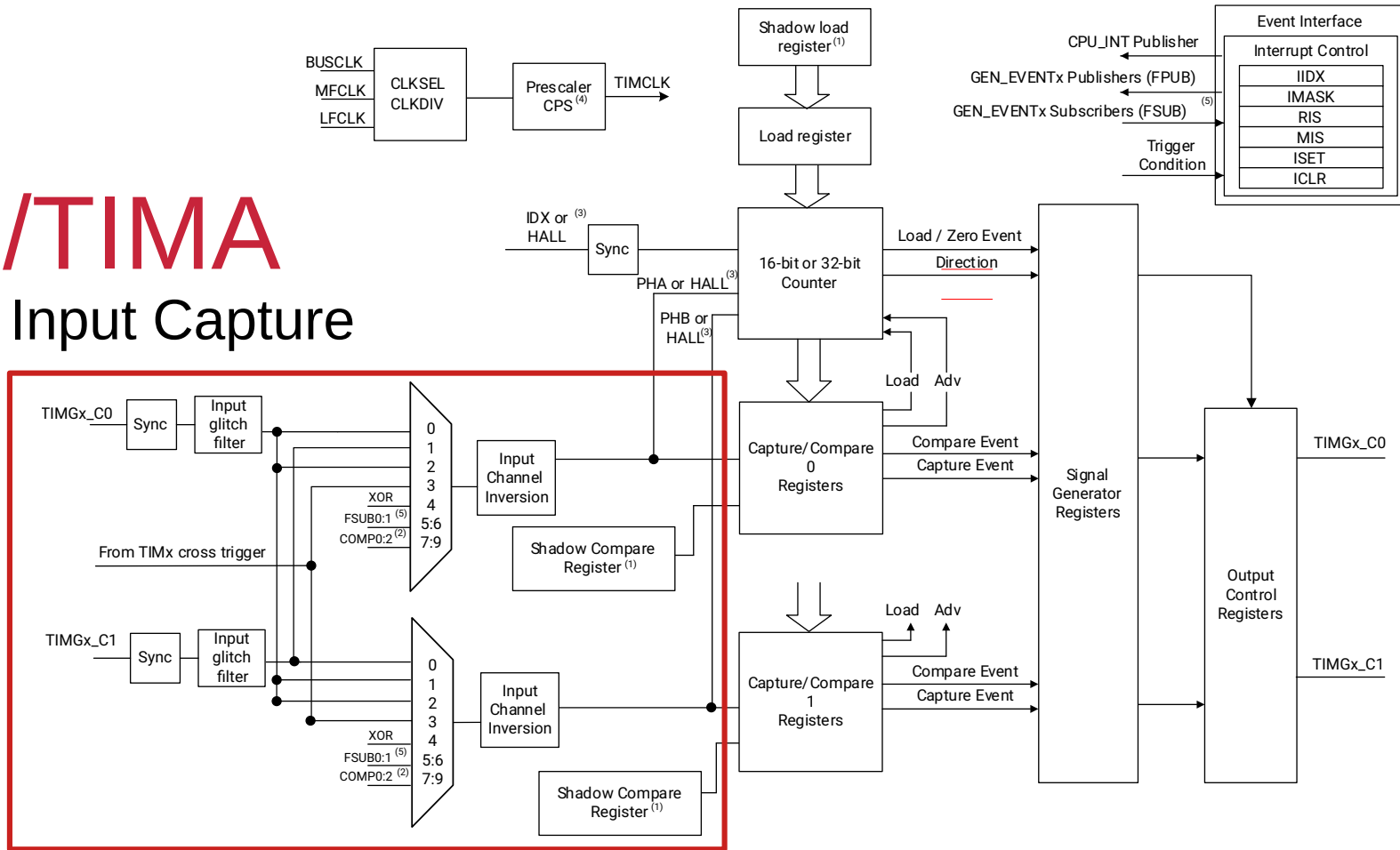
– Timer **Capture** Modes:

- Measure time between signal edges
- Capable of measuring frequency, delay, duty cycle, etc.
- Loss of information due to **Time Resolution**

Timer Capture Modes

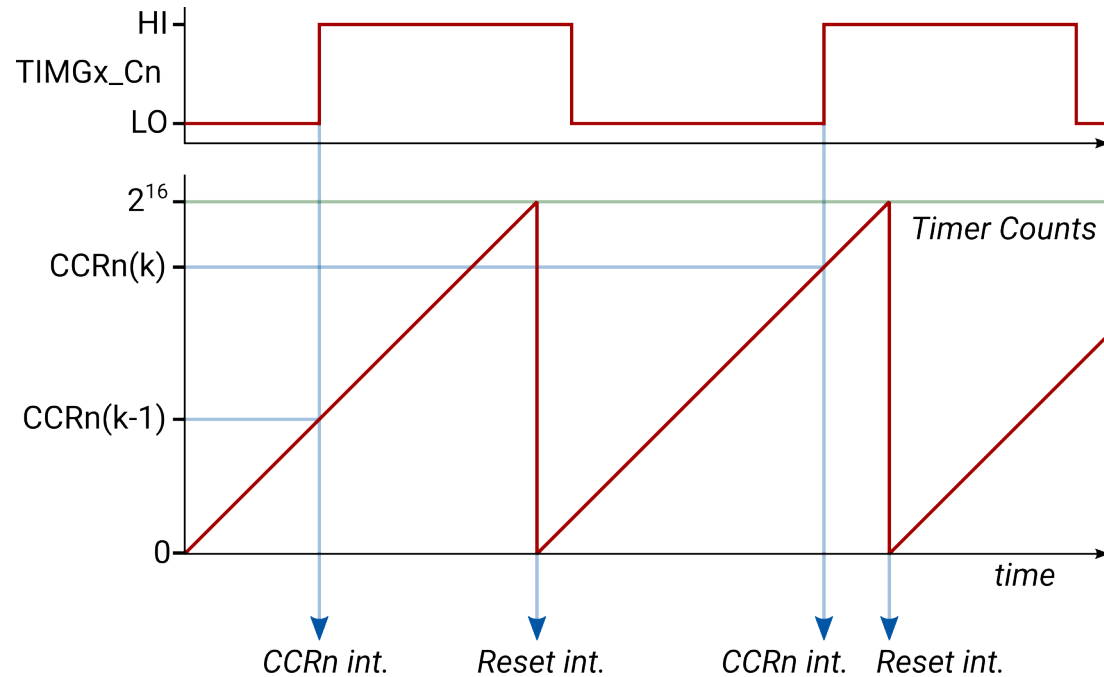
TIMG/TIMA

Hardware Input Capture



Timer CCR in Capture Mode

- Timer counts as normal
- **CCRn** monitors an external signal **TIMGx_Cn** for edges
- Edge detected:
 - **Capture Event**
 - Save curr. Timer count (in CCRn register)
 - Trigger Interrupt
 - Return to monitoring



TIMG/TIMA in *Up Mode*

Timer CCR in Capture Mode

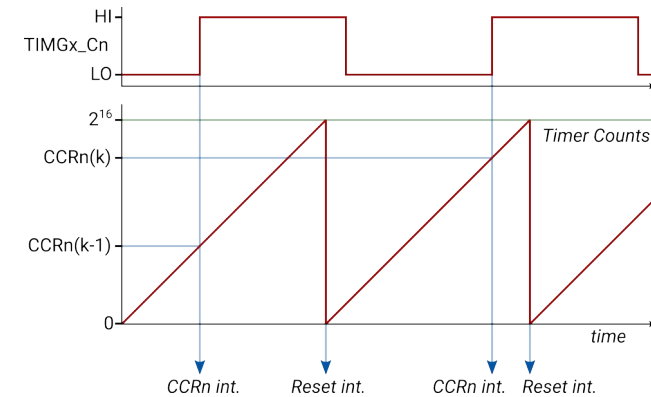
How to use Capture Events for Motor / Wheel Encoder

- **Measure Distance Traveled:** Each event signals a 1/360 rotation of the wheel:
 - *Sum capture events, directly convert to distance.*

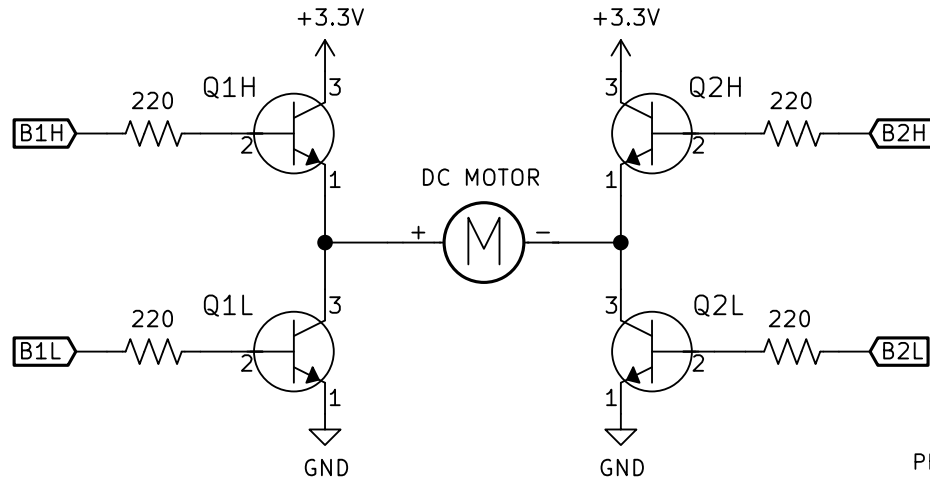
- **Measure Wheel Speed:** Keep track of time between events, convert to rpm, etc.

$$\Delta t(k) = \frac{(CCRn(k) - CCRn(k-1)) + 2^{16} N_{resets}(k)}{f_{TIMCLK}} \quad \omega = \frac{1}{360} \frac{1}{\Delta t} [\text{rpm}] = \frac{2\pi}{360} \frac{1}{\Delta t} \left[\frac{\text{rad}}{\text{s}} \right]$$

- **Measure Car Speed:** Convert wheel speed to car speed knowing wheel radius $v = \omega r$



Activity 9: H-Bridge



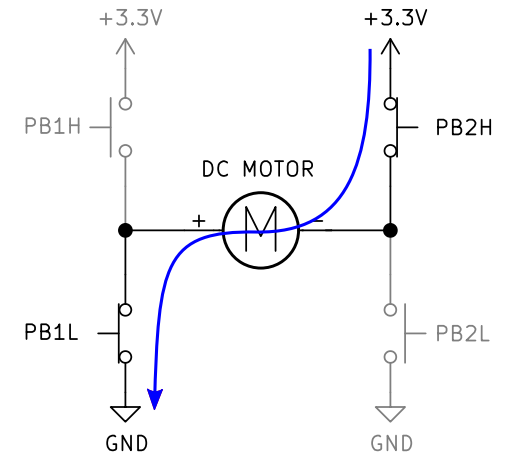
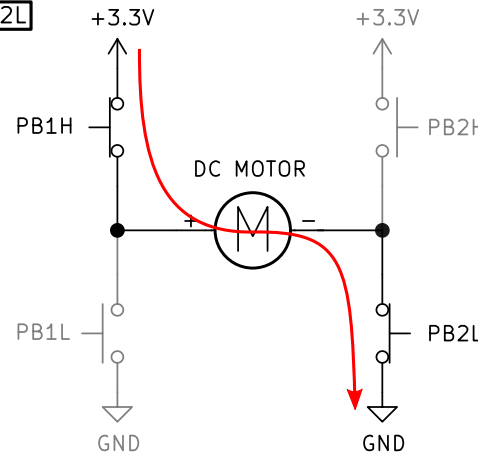
No direct speed control!

H-Bridge provides 3-states:

On (forward current)

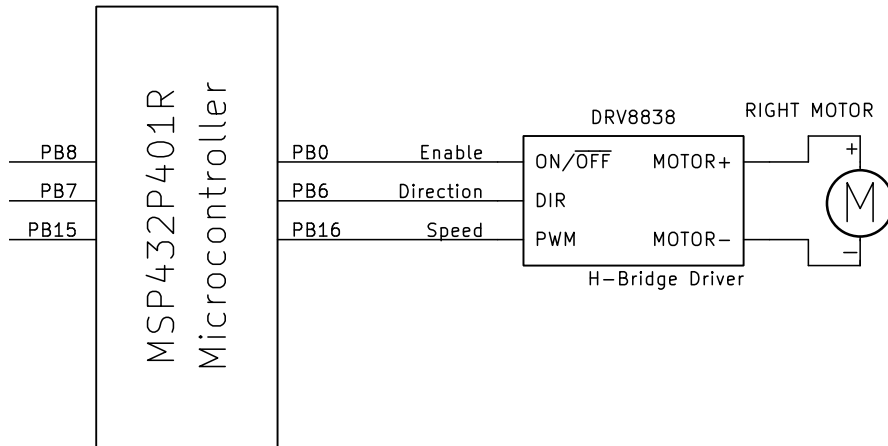
On (reverse current)

Off (no current)



Motors: H-Bridge RSLK Support

- The RSLK has H-Bridge chips to support the motors



DRV8837, DRV8838
SLVSB44F – JUNE 2012 – REVISED APRIL 2021

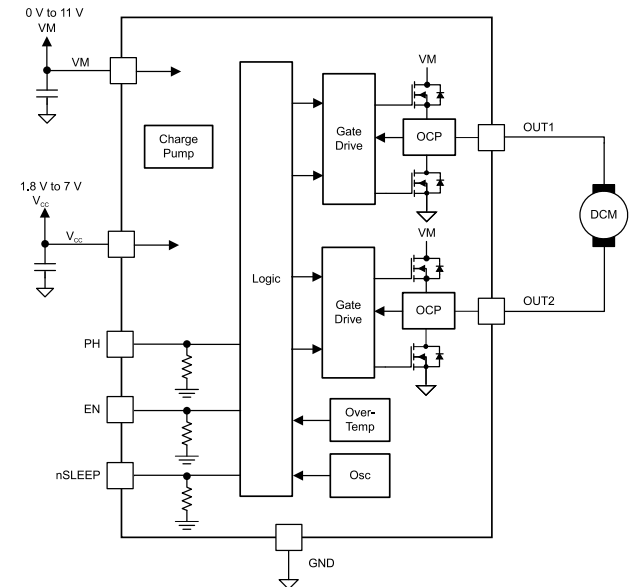


Figure 7-2. DRV8838 Functional Block Diagram

Activity 10/11: RSLK Motors/Encoders

- Introduces:
 - **A10**: PWM to control wheel speed and
 - **A11**: Encoders to measure wheel speed and drive distance.
 - **Don't let cars drive off benches!!**
 - **Leave on foam blocks when not driving/testing/etc.**