

# FRA 222 microcontroller interface

---

WEEK 06 - ONSITE

INPUT CAPTURE

OUTPUT COMPARE

# summary

---

**Input Capture** – หา **edge** ที่กำหนดของสัญญาณภายนอกแล้ว เก็บค่า **counter** ของ **timer** ปัจจุบัน ณ จังหวะเวลาสัญญาณ

- ใช้วัดเวลาระหว่าง **edge – edge** ของสัญญาณได้
- ใช้ในการอ่านเซนเซอร์ หรือ การสื่อสารใดๆ ซึ่งให้ค่าออกมาในลักษณะ ของ ระยะเวลาของสัญญาณ

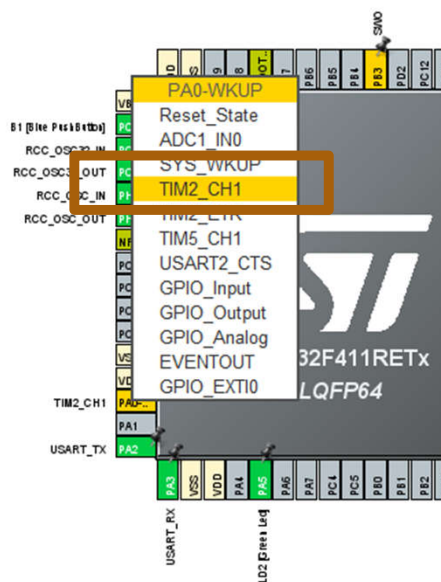
**Output Comparator** – Compare **counter** ของ **timer** กับ ค่าๆ หนึ่งที่กำหนดไว้ แล้ว **output** สัญญาณตามค่าเปรียบเทียบที่ได้

- ใช้ในการสร้างสัญญาณซึ่งกำหนดระยะเวลาเปิดปิดได้ เช่น **PWM**
- ใช้ในการส่งข้อมูลที่ใช้ระยะเวลาของสัญญาณแทนข้อมูลได้

# Example 6

วัดความกว้างของ pulse encoder motor 1 ch แบบ rising edge ออกมาในหน่วย us โดยใช้ timer 2 ,Input capture ch.1

หาค่าเฉลี่ยระยะห่างของเวลา 5 ครั้ง ล้ำสุด ทุกๆ 0.5 วินาที



เปิด input capture

### TIM2 Mode and Configuration

Mode	
Slave Mode	Disable
Trigger Source	Disable
Clock Source	Internal Clock
Channel1	Input Capture direct mode
Channel2	Disable
Channel3	Input Capture direct mode

Prescaler (PSC - 16 bits value)	83
Counter Mode	Up
Counter Period (AutoReload Regi...	4294967295
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
Input Capture Channel 1	
Polarity Selection	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter (4 bits value)	0

## Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

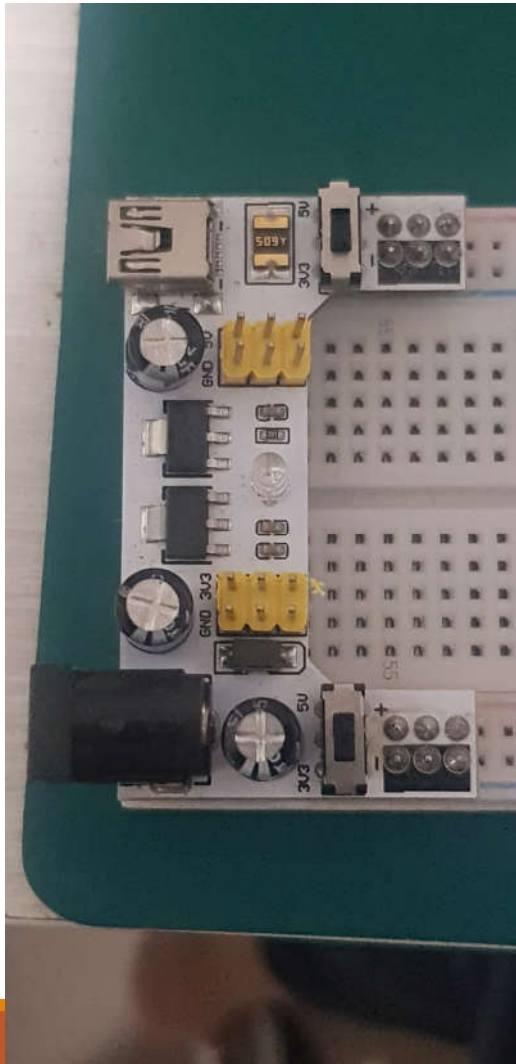
DMA Request	Stream	Direction	Priority
TIM2_CH1	DMA1 Stream 5	Peripheral To Memory	Low

Add

Delete

### DMA Request Settings

Mode	<input type="text" value="Circular"/>	Increment Address	<input type="checkbox"/>	Peripheral	<input type="checkbox"/>	Memor	<input checked="" type="checkbox"/>
Use Fifo	<input type="checkbox"/>	Threshold	<input type="text"/>	Data Width	<input type="text" value="Word"/>	Word	<input type="text" value="Word"/>
		Burst Size	<input type="text"/>		<input type="text"/>		<input type="text"/>

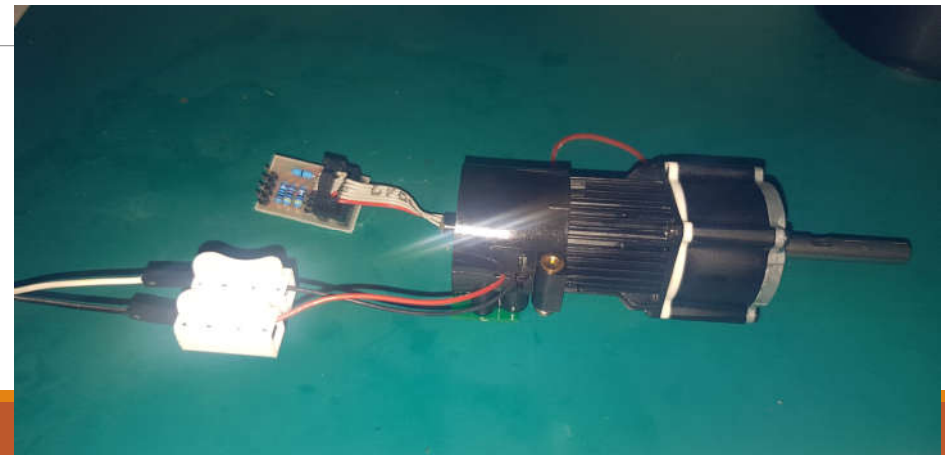
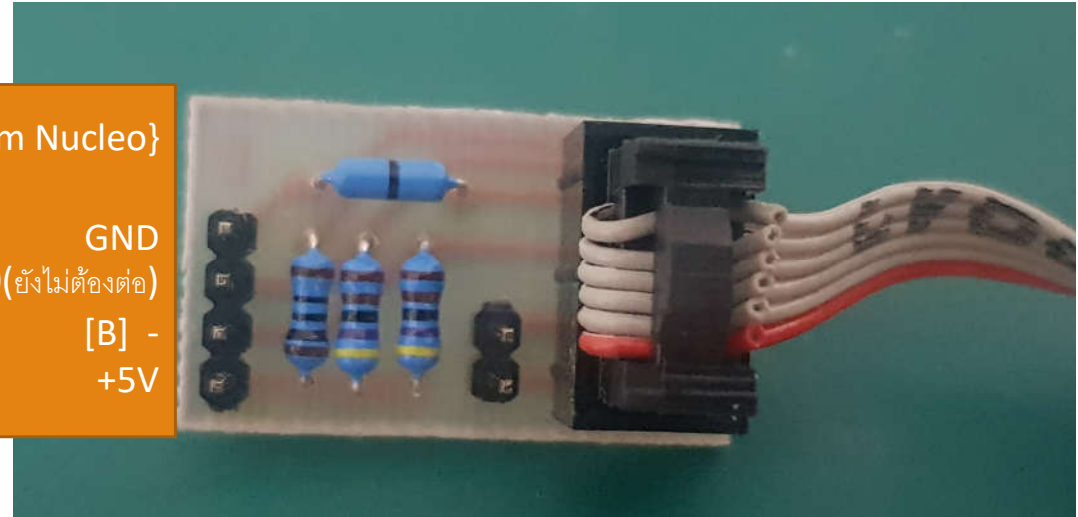


{From Nucleo}

GND

[A] PA0(ยังไม่ต้องต่อ)

[B] - +5V



```
2 /* Private Includes ----- */
3 /* USER CODE BEGIN Includes */
4 #define IC_BUFFER_SIZE 20
5 /* USER CODE END Includes */
```

```
6
7 /* USER CODE BEGIN PV */
8 uint32_t InputCaptureBuffer[IC_BUFFER_SIZE];
9 float averageRisingedgePeriod;
10 /* USER CODE END PV */
```

```
/* USER CODE BEGIN PFP */
float IC_Calc_Period();
/* USER CODE END PFP */
```

```
/* USER CODE BEGIN 2 */

HAL_TIM_Base_Start(&htim2);
HAL_TIM_IC_Start_DMA(&htim2, TIM_CHANNEL_1, InputCaptureBuffer, IC_BUFFER_SIZE);
/* USER CODE END 2 */
```

```

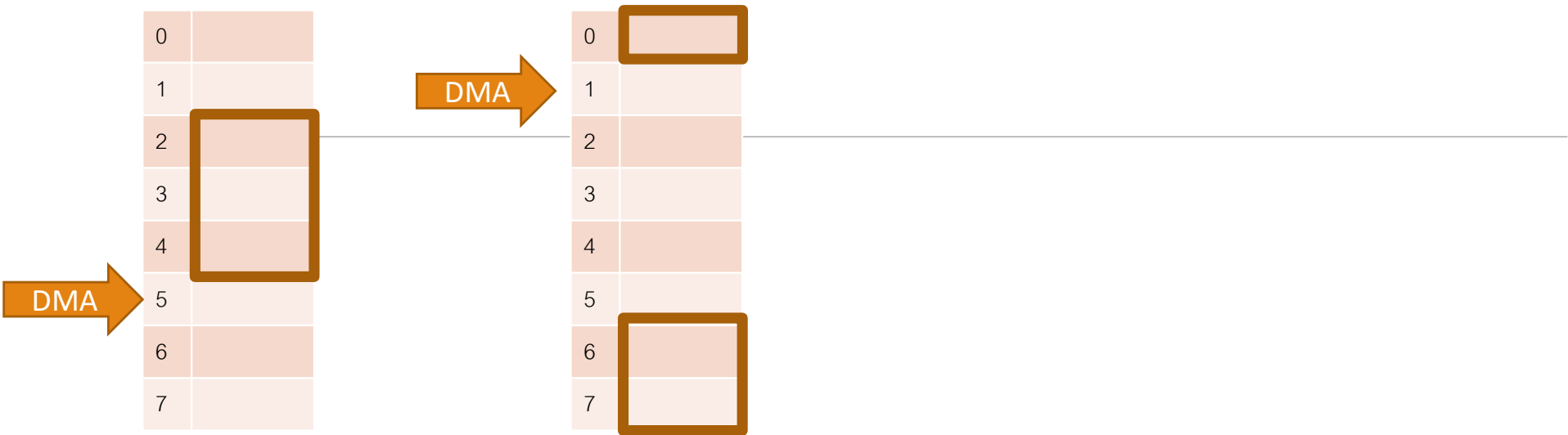
392 /* USER CODE BEGIN 4 */
393 float IC_Calc_Period()
394 {
395     uint32_t currentDMAPointer = IC_BUFFER_SIZE - __HAL_DMA_GET_COUNTER(htim2.hdma[1]);
396     uint32_t lastVaildDMAPointer = (currentDMAPointer-1 + IC_BUFFER_SIZE) % IC_BUFFER_SIZE;
397     uint32_t i = (lastVaildDMAPointer + IC_BUFFER_SIZE - 5) % IC_BUFFER_SIZE;
398
399     int32_t sumdiff = 0;
400     while (i != lastVaildDMAPointer)
401     {
402         uint32_t firstCapture = InputCaptureBuffer[i] ;
403         uint32_t NextCapture = InputCaptureBuffer[(i+1)%IC_BUFFER_SIZE];
404         sumdiff += NextCapture - firstCapture;
405         i = (i+1) % IC_BUFFER_SIZE;
406     }
407     return sumdiff / 5.0;
408 }
409
410 /* USER CODE END 4 */

```

```

987 /** @defgroup DMA_Handle_index TIM DMA Handle Index
988 * @{
989 */
990 #define TIM_DMA_ID_UPDATE ((uint16_t) 0x0000)
991 #define TIM_DMA_ID_CC1 ((uint16_t) 0x0001)
992 #define TIM_DMA_ID_CC2 ((uint16_t) 0x0002)
993 #define TIM_DMA_ID_CC3 ((uint16_t) 0x0003)
994 #define TIM_DMA_ID_CC4 ((uint16_t) 0x0004)
995 #define TIM_DMA_ID_CC5 ((uint16_t) 0x0005)
996 #define TIM_DMA_ID_CC6 ((uint16_t) 0x0006)

```



```
9  /* USER CODE BEGIN 4 */  
10 float IC_Calc_Period()  
11 {  
12     uint32_t currentDMAPointer = IC_BUFFER_SIZE - __HAL_DMA_GET_COUNTER((htim2.hdma[1]));  
13     uint32_t lastVaildDMAPointer = (currentDMAPointer-1 + IC_BUFFER_SIZE) % IC_BUFFER_SIZE;  
14     uint32_t i = (lastVaildDMAPointer + IC_BUFFER_SIZE - 6) % IC_BUFFER_SIZE;  
15 }
```

DMA เหลืออีกช่อง



```

392 /* USER CODE BEGIN 4 */
393 float IC_Calc_Period()
394 {
395     uint32_t currentDMAPointer = IC_BUFFER_SIZE - __HAL_DMA_GET_COUNTER((htim2.hdma[1]));
396
397     uint32_t lastVaildDMAPointer = (currentDMAPointer-1 + IC_BUFFER_SIZE) % IC_BUFFER_SIZE;
398
399     uint32_t i = (lastVaildDMAPointer + IC_BUFFER_SIZE - 5) % IC_BUFFER_SIZE;
400
401     int32_t sumdiff = 0;
402     while (i != lastVaildDMAPointer)
403     {
404         uint32_t firstCapture = InputCaptureBuffer[i] ;
405
406         uint32_t NextCapture = InputCaptureBuffer[(i+1)%IC_BUFFER_SIZE];
407         sumdiff += NextCapture - firstCapture;
408         i = (i+1) % IC_BUFFER_SIZE;
409     }
410     return sumdiff / 5.0;
411 }
412
413 }
414 /* USER CODE END 4 */

```

```

/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    static uint32_t timestamp = 0;
    if (HAL_GetTick() >= timestamp)
    {
        timestamp = HAL_GetTick()+500;
        averageRisingedgePeriod = IC_Calc_Period();

    }
}
/* USER CODE END 3 */

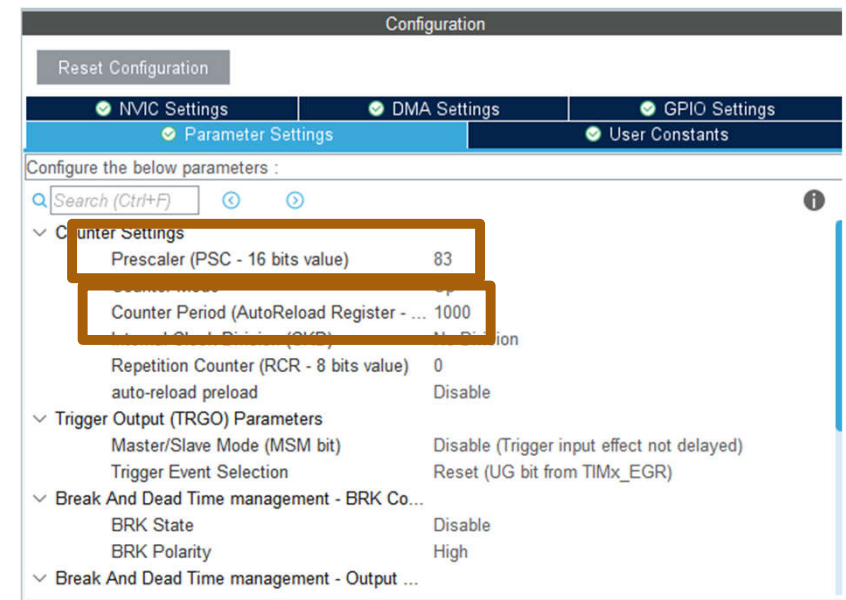
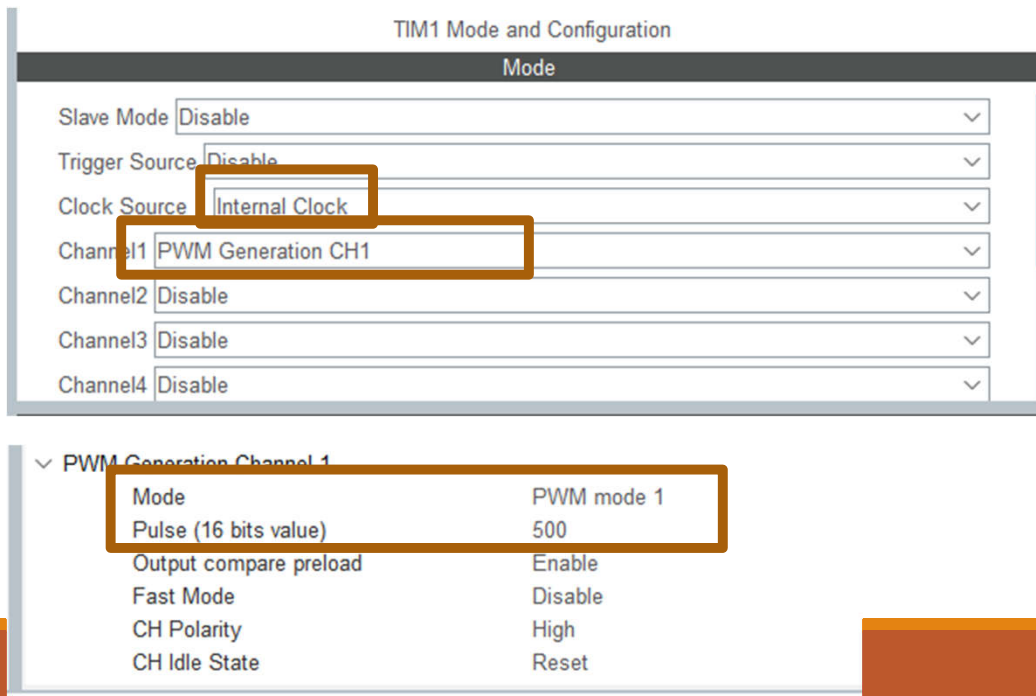
```

---



# EX

- 1.สร้าง สัญญาณ PWM ผ่าน Timer 1 Ch 1 ที่ช่อง PA8โดยสร้างความเร็ว 1kHz duty cycle 50 %
- 2.ปรับสัญญาณDuty cycle ระหว่างการ RUN โดย เปลี่ยนด้วยความเร็ว 2 Hz



```

48 /* USER CODE BEGIN PV */
49 uint32_t InputCaptureBuffer[IC_BUFFER_SIZE];
50 float averageRisingedgePeriod;
51 uint32_t duty = 500;
52 /* USER CODE END PV */

```

```

/* USER CODE BEGIN 2 */

```

```

HAL_TIM_Base_Start(&htim2);
HAL_TIM_IC_Start_DMA(&htim2, TIM_CHANNEL_1, InputCaptureBuffer, IC_BUFFER_SIZE);

```

```

HAL_TIM_Base_Start(&htim1);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
/* USER CODE END 2 */

```

```

/* USER CODE BEGIN 3 */

```

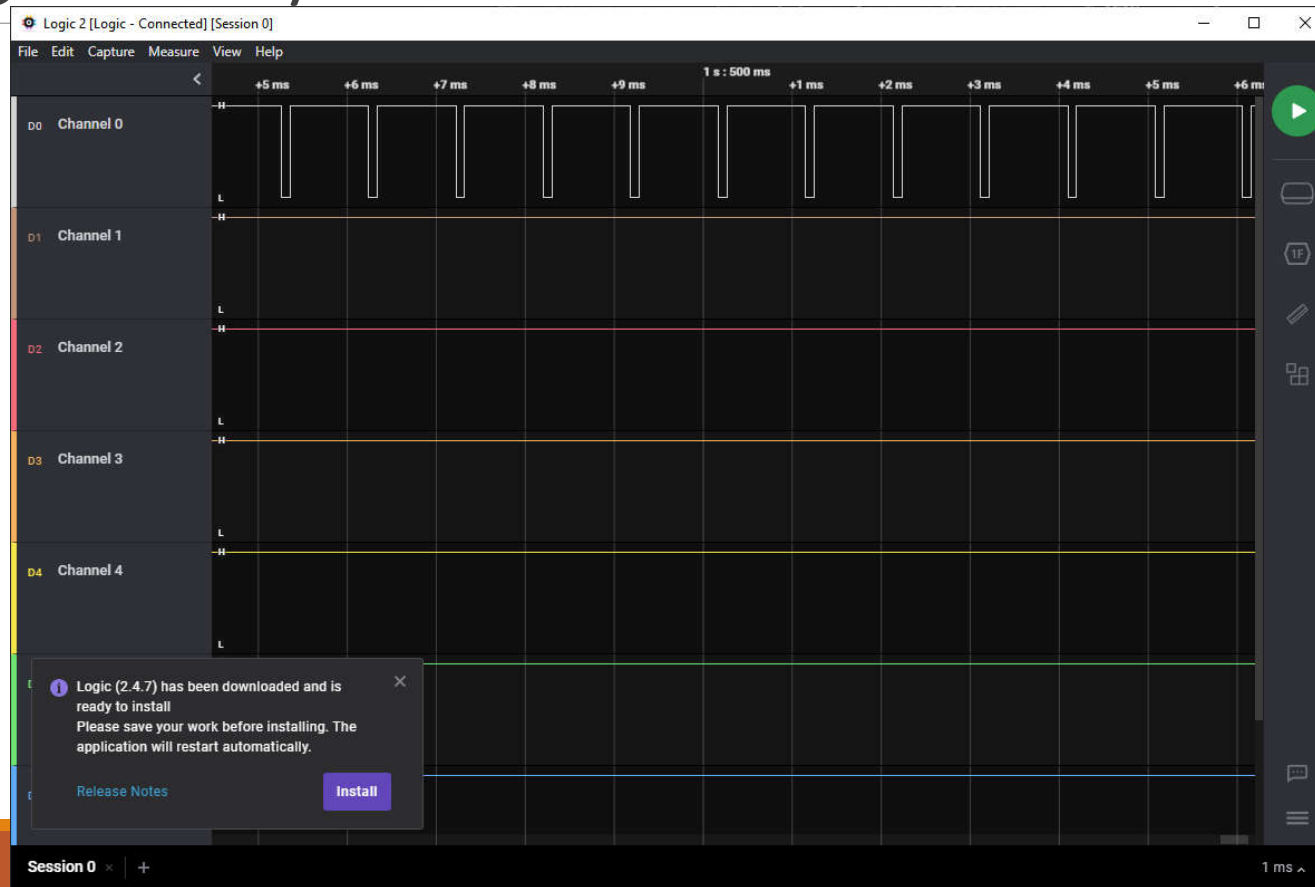
```

static uint32_t timestamp = 0;
if (HAL_GetTick() >= timestamp)
{
    timestamp = HAL_GetTick() + 500;
    averageRisingedgePeriod = IC_Calc_Period();

    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, duty);
}
}
/* USER CODE END 3 */

```

# logic analyzer



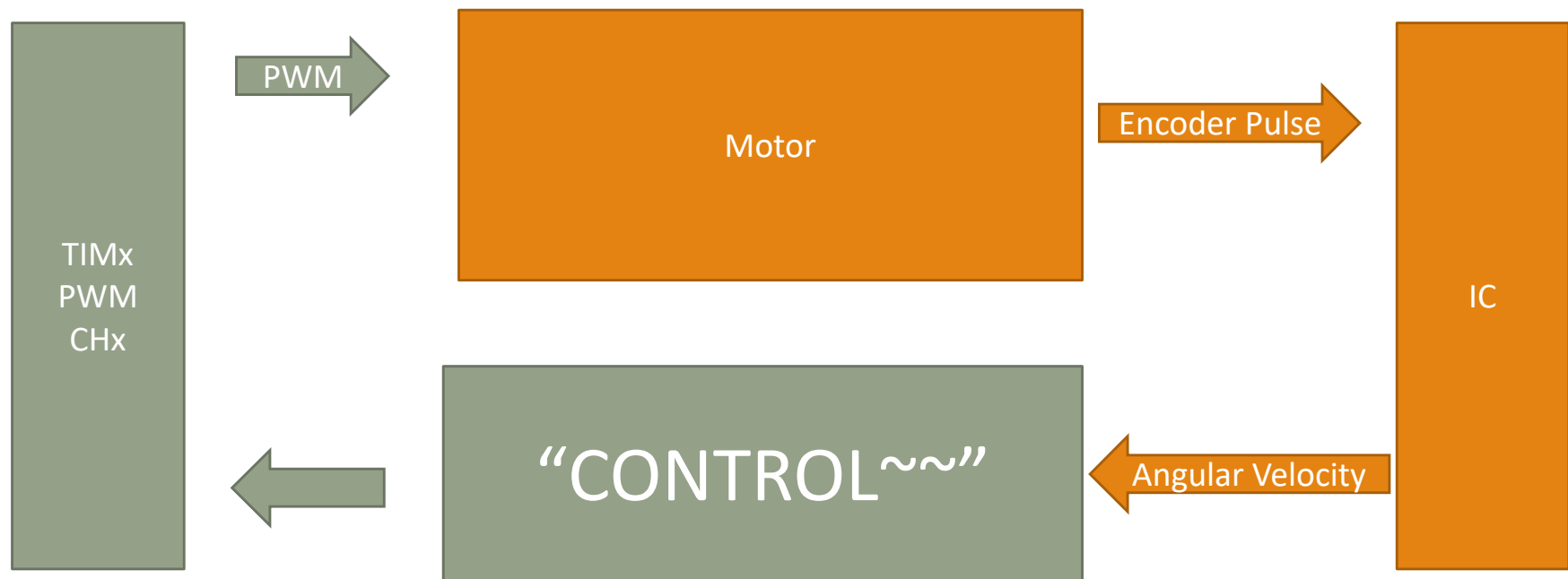
# LAB 3

---

1. สร้างระบบ ปรับความเร็วมอเตอร์ โดยสามารถกำหนดค่า **PWM Duty cycle** ได้ ระหว่าง 0-100 % โดยมีความละเอียดในการปรับ 1%
  - ปรับค่า ระหว่าง **0 – 100** ในตัวแปร ชื่อ **MotorSetDuty** ใน live expression (ในที่นี้ กำหนดให้มอเตอร์หมุนได้ทางเดียว) (ความถี่ PWM เท่าไรก็ได้)
2. สร้างระบบ อ่านความเร็วมอเตอร์ โดยใช้ **input capture** ในการอ่านความเร็วปัจจุบันของมอเตอร์ ในหน่วย **RPM**
  - มอเตอร์ มีอัตราทดเฟือง ที่ **1 : 64** และ ใบพัด encoder 12 ใบ อ่านค่าจากตัวแปรที่ชื่อ **MotorReadRPM** ใน Live Expression
3. สร้างระบบควบคุมความเร็วอย่างง่าย เพื่อควบคุมความเร็วของมอเตอร์ให้อยู่ในความเร็วที่ต้องการ (ใช้ระบบควบคุมแบบไหนก็ได้) โดยระบุความเร็วในหน่วย **RPM** ในตัวแปรชื่อ **MotorSetRPM** โดย ระบบควบคุมดังกล่าวสามารถเปิด ปิดการทำงานได้โดยการใช้ ตัวแปร **MotorControlEnable** หาก **set** เป็น 1 ก็จะทำงาน หาก **set** เป็น 0 ระบบนี้จะไม่ทำงานและมอเตอร์จะหมุนตามค่าที่กำหนดในข้อ 1 (มอเตอร์ไม่มีโหลดขณะ ใช้งาน)

## คำถาม

- 1. นื่องๆ แปลงหน่วย จาก **duty cycle** ไปใช้ใน **output compare** อย่างไร
- 2. นื่องๆ แปลงหน่วย จาก **input capture** ไปเป็น **RPM** อย่างไร
- 3. นื่องควบคุมความเร็วมอเตอร์ให้เป็นไปตามที่ต้องการได้อย่างไร



## Motor Drive

### Features introduction :

- Size: 18\* 21mm
- Input voltage: 2.7 V – 10.8 V
- Single H- bridge output current: 1.5A , can drive 2 DC geared motors.
- Built-in overcurrent protection, short circuit protection, undervoltage lockout and overheat protection. With low-power sleep mode.
- Uses: It can drive DC motors below 1.5A and 4- wire stepper motors.

### Pin description:

- ANI1: AO1 logic input control port, level 0-5V.
- AIN2: AO2 logic input control port, level 0-5V.
- BNI1: The logic input control port of BO1, level 0-5V.
- BIN2: Logic input control port of BO2, level 0-5V.
- AO1 and AO2 are 1-channel H-bridge output ports, which are connected to the two feet of a DC motor.
- BO1 and BO2 are 2-way H-bridge output ports, which are connected to the two feet of another direct motor.
- GND: ground.
- The VM : the motor supply pin and the chip, the voltage range of 2.7 V - 10.8 V .
- STBY : Ground or floating chip does not work, no output, connect to 5 V to work ; level 0 -5V .
- NC : empty foot

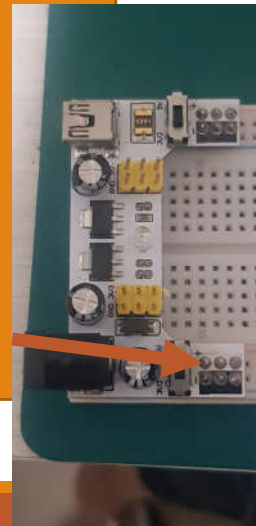
LOGIC-LEVEL INPUTS				
V <sub>IL</sub>	Input low voltage	nSLEEP	0.5	V
		All other pins	0.7	
V <sub>IH</sub>	Input high voltage	nSLEEP	2.5	V
		All other pins	2	



AIN1 – PWM1  
 AIN2 – GND  
 BIN1 – GND  
 BIN2 – GND  
 AO1 – motor  
 AO2 – motor

GND – GND  
 STBY – 3.3V

VM – 3.3V External





### H- bridge logic drive :

xIN1	x IN2	x O1	xO2	Features
0	0	Z	Z	Inertia running / fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake / slow speed damping

### PWM control motor speed :

xIN1	x IN2	Features
PWM	0	Forward P WM , fast attenuation
1	PWM	Forward P WM , slow decay
0	PWM	Reverse P WM , block speed attenuation
PWM	1	Reverse P WM , slow decay

# LAB 3

---

## DUE DATE

- 22 / 23 Mar 2023

การต่อคิวส่งงาน — ตรวจ online เท่านั้น

- ยกเลิกระบบต่อคิวเก่าใน Discord
- ยกเลิกการตรวจในคาบ office hour — นอกกรอบ ในกรณีติดธุระในวันส่งให้ติดต่อมาพิจารณาเป็นรายครั้งไป
- จะมีแบบ google form ให้ลง โดยลงเมื่อทำเสร็จแล้วเท่านั้น ภายในฟอร์มจะสามารถเลือกช่วงเวลาตรวจได้เสร็จก่อน มีสิทธิเลือกก่อน
  - ตรวจเช็คเวลาตาม GitHub ถ้า ไม่มี commit ที่สามารถ ใช้งานได้จริง ตามเวลาที่ระบุใน Google form TA สามารถไม่ตรวจ หรือ ลดคะแนนตามความเหมาะสม
  - สามารถแก้ไขได้หลังลงเวลาไปแล้ว แต่อย่างไรก็ตาม เป็นการแก้ไขเพื่อปรับปรุงเล็กน้อยเท่านั้น ดังนั้นก่อนส่งโปรดตรวจทานให้เรียบร้อย
- หากมีข้อสงสัย สอบถามได้ใน Discord ห้องถามตอบ
- การตรวจ จะเป็นไปตามตารางเวลาที่ประกาศก่อนถึงเวลา โดยเวลาอาจจะคลาดเคลื่อนได้ +/- 10 นาที เพื่อให้ พี่ๆ TA ได้เช็คความเข้าใจของเพื่อนน้อง ๆ ได้ถูกต้อง มาเตรียมตัวก่อนเวลาประมาณ 15 นาที