

FRA222 Microcontroller Interface

02 - GPIO



Spoil Alert!!!

Phase	Week	Date	Main Topic	LAB
Phase 1: Basic Interface and microcontroller programming	1	Wed, 18, Jan	Basic C	
		Thu, 19, Jan	STM32 Environment	
	2	Wed, 25, Jan	GPIO : General-purpose input/output	LAB1 : GPIO safe numpad
		Thu, 26, Jan		
	3	Wed, 01, Feb	ADC : Analog-to-digital converter	กำหนดส่ง LAB 1
		Thu, 02, Feb		
	4	Wed, 08, Feb	blocking and non-blocking operation	LAB2 : ADC Analog with Non Blocking Volt meter
		Thu, 09, Feb	INT : interrupt DMA : Direct Memory Access	
	5	Wed, 15, Feb	Office Hour	กำหนดส่ง LAB 2
		Thu, 16, Feb		
	6	Wed, 22, Feb	Competency 1 : Basic Interface	
		Thu, 23, Feb		

Phase	Week	Date	Main Topic	LAB
Phase 2: Timer and control system in MCU	7	Wed, 01, Mar	TIM : Timer	
		Thu, 02, Mar	Software timer	
	8	Wed, 08, Mar	TIM - IC : Input Capture	Lab 3 TIMER : motor control and speed reader
		Thu, 09, Mar	TIM - OC : Output Compare	
	9	Wed, 15, Mar	TIM - QEI: Quadrature Encoder Interface	กำหนดส่ง LAB3
		Thu, 16, Mar		
	10	Wed, 22, Mar	DSP : Digital Signal Processing Libraly	LAB 4 DSP :Motor Speed control
		Thu, 23, Mar		
	11	Wed, 29, Mar	Office Hour	กำหนดส่ง LAB 4
		Thu, 30, Mar		
	12	Wed, 05, Apr	Competency 2 : Maddess of Timer	
		Thu, 06, Apr		

Phase	Week	Date	Main Topic	LAB	Note
Phase 3 : Communication and Protocol	13	Wed, 12, Apr			สงกรานต์
		Thu, 13, Apr			
	14	Wed, 19, Apr	UART : Universal asynchronous receiver-transmitter		
		Thu, 20, Apr			
	15	Wed, 26, Apr	I2C : Inter-Integrated Circuit	LAB 5 : UART Control I2C Memory	
		Thu, 27, Apr			
	16	Wed, 03, May	SPI : Serial Peripheral Interface	กำหนดส่ง LAB 5	นัดรวมคล
		Thu, 04, May			
	17	Wed, 10, May	Protocol	LAB 6 UART Protocal to SPI	
		Thu, 11, May			
	18	Wed, 17, May	Office Hour	กำหนดส่ง LAB 6	
		Thu, 18, May			
	19	Wed, 24, May	Competency 3 : Just Communicate "SOMETHING"		
		Thu, 25, May			
	20	Wed, 31, May			
		Thu, 01, Jun			

Example 1 ทบทวนความรู้!!!

0. ตั้งค่า stm32cubeMX

1. ทดลองควบคุม LED LD2 บนบอร์ด

- LD2 ต่อกับขา PA5 บนบอร์ดอยู่

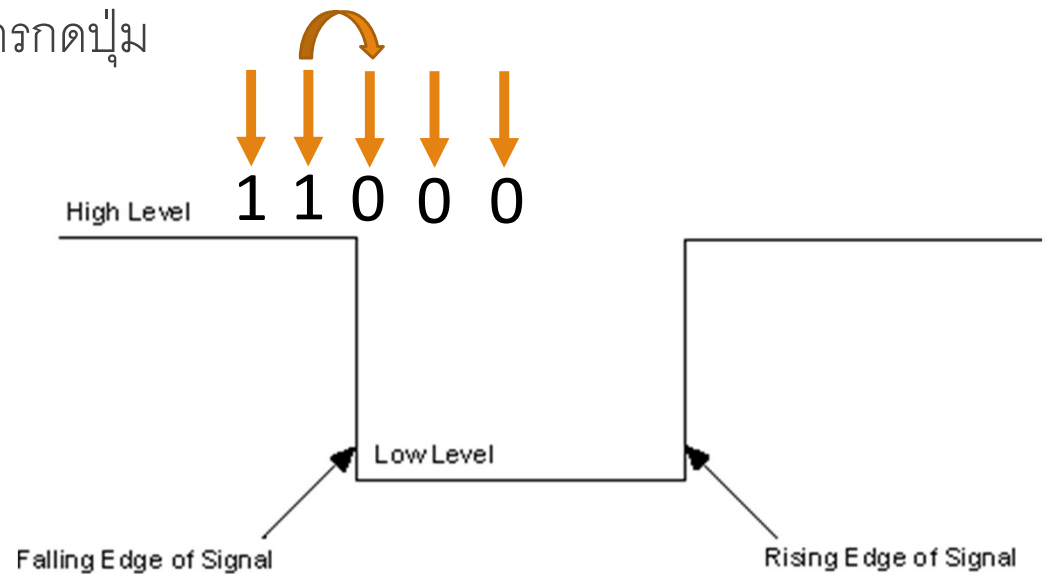
2. ทดลอง ปุ่ม B1

- B1 ต่อกับขา PC13 อยู่

Example 1

2.ทดลอง ปุ่ม B1

- B1 ต่อกับขา PC13 อยู่
- ทดลองควบคุม LED ให้เป็นไปตาม การกดปุ่ม
 - 1. กดติด ปลอยดับ
 - 2. กดติด กดดับ (Toggle)



Example 1

3. ทดลองกะพริบไฟ LED ที่ 1 Hz
- โดยใช้ HAL_Gettick()

- ~~Mbed studio~~

- ~~us_ticker_read();~~ → ให้เวลานับจากเปิดเครื่องในหน่วย ~~μs~~

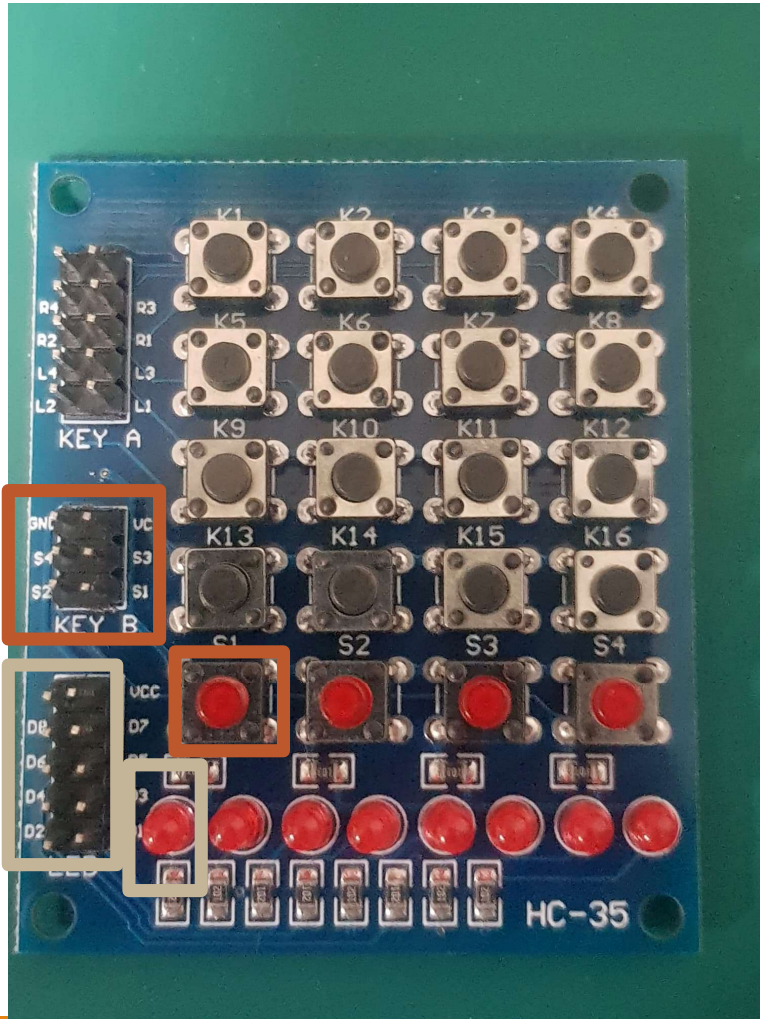
- STM32CubeIDE

- HAL_GetTick(); ให้เวลานับจากเปิดเครื่องในหน่วย *ms*

Example 1

4.ทดลองกะพริบไฟ LED ที่ 1 Hz และ 0.5 Hz

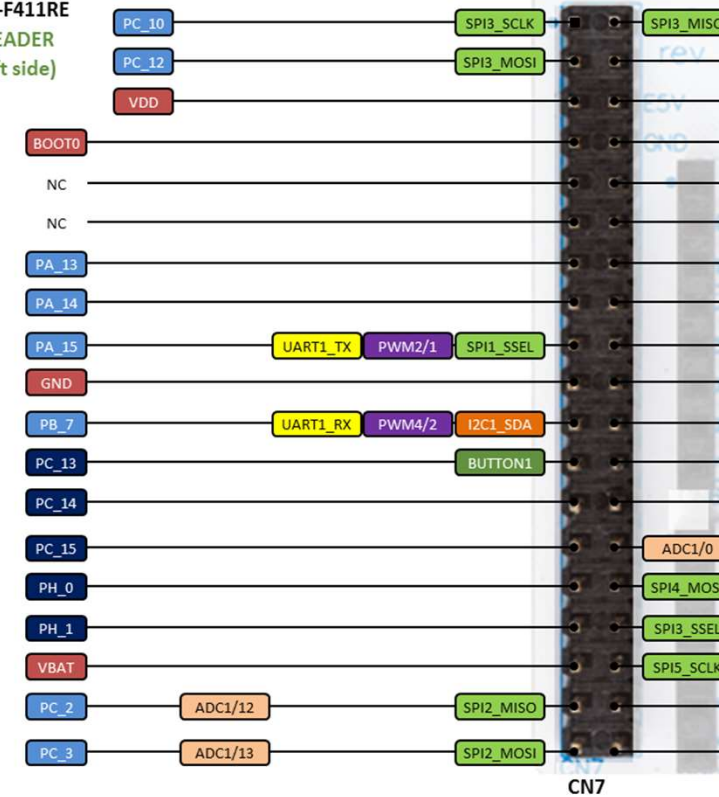
-โดยใช้ B1 กดเพื่อเปลี่ยนความถี่สลับกัน (toggle)



GND – GND
VCC – +3V3
S1 – PC10

VCC – +3V3
D1 – PC12

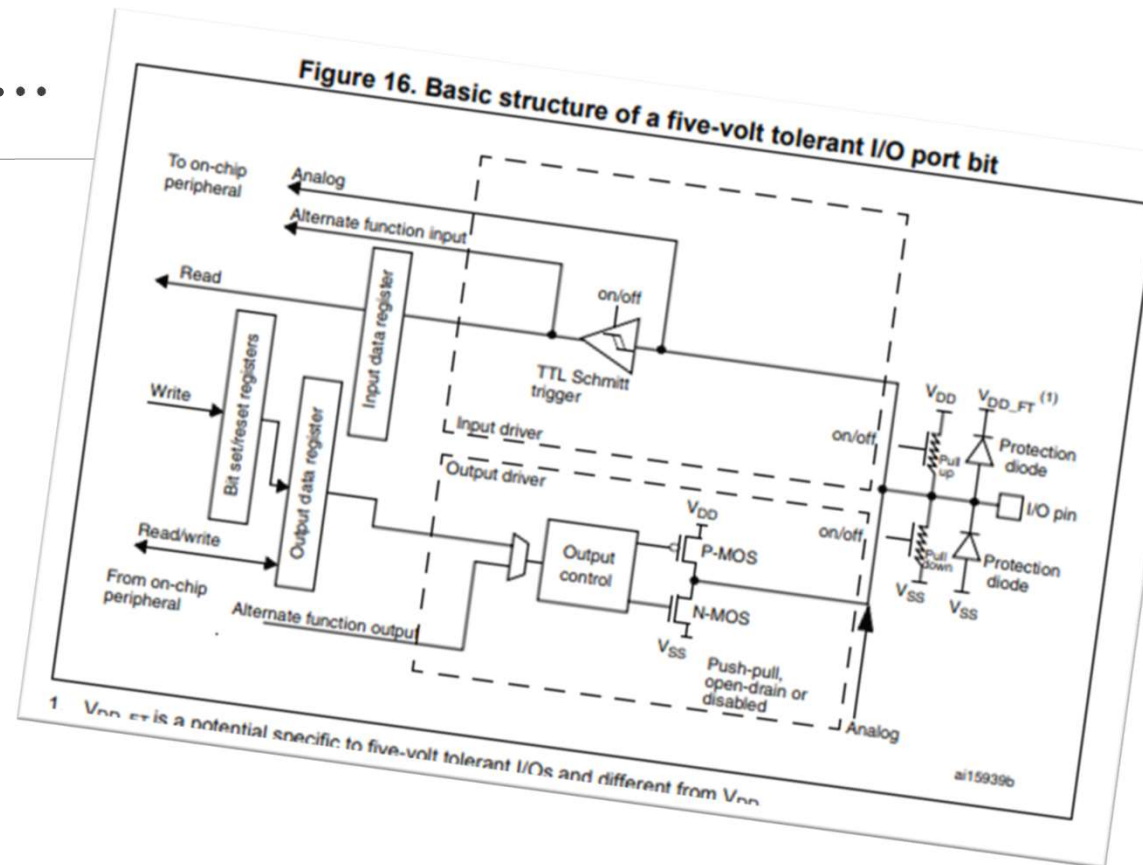

life.augmented
NUCLEO-F411RE
CN7 HEADER
(top left side)





Dig A Little Deeper...

- IO Structure
- Pull up/Pull Down
- Push Pull/Open Drain
- Output Speed



Datasheet Source

<https://www.st.com/resource/en/datasheet/stm32f411re.pdf>

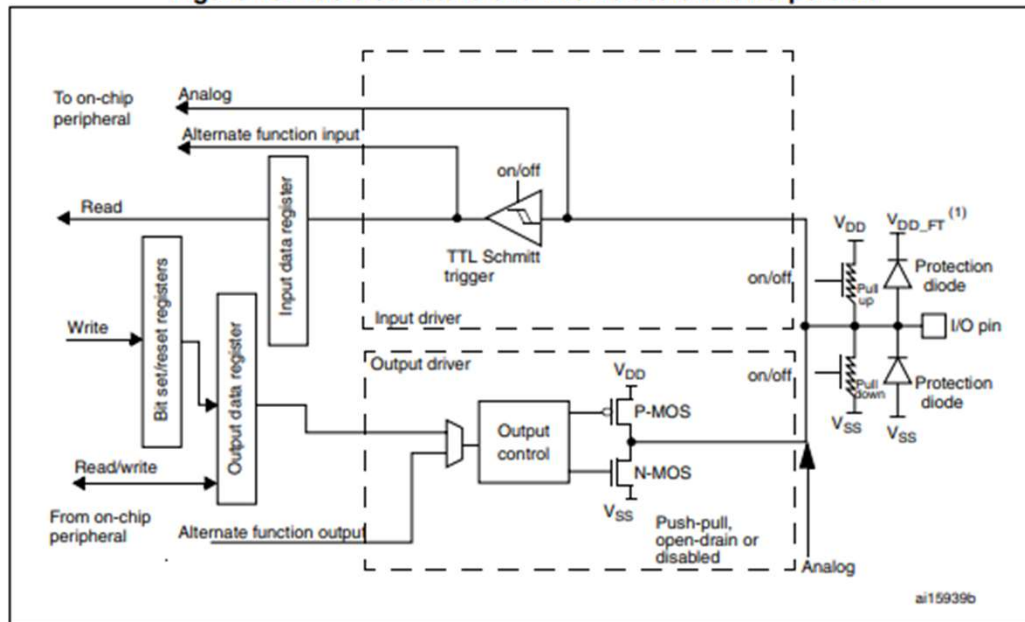
https://www.st.com/resource/en/reference_manual/dm00119316-stm32f411xce-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

<http://ww1.microchip.com/downloads/en/devicedoc/70291f.pdf>

<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>

I/O Structure

Figure 16. Basic structure of a five-volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD} .

Figure 14-2. General Digital I/O⁽¹⁾

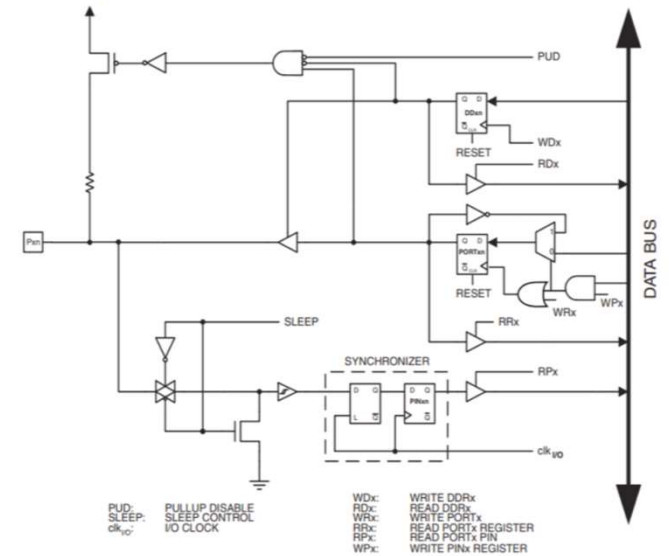
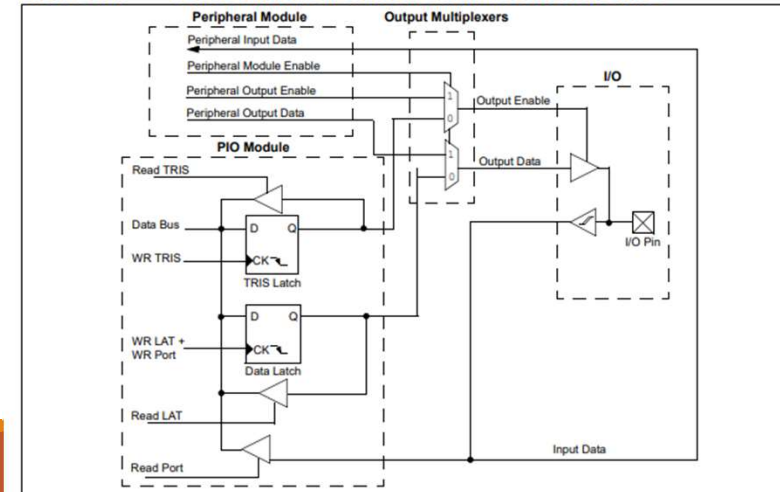
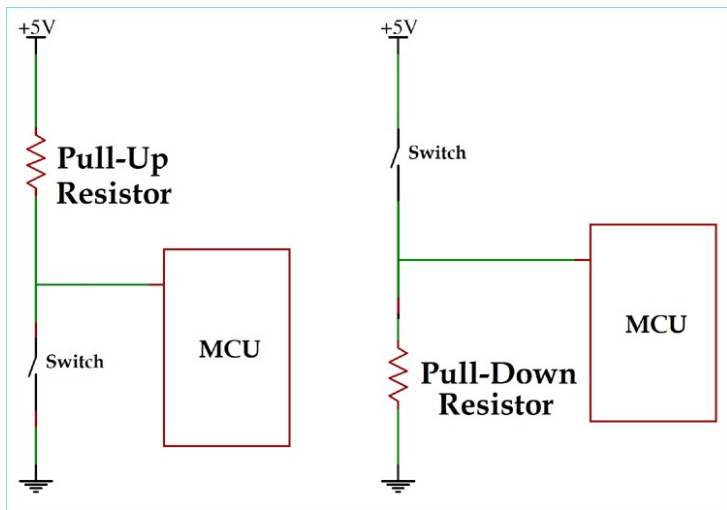


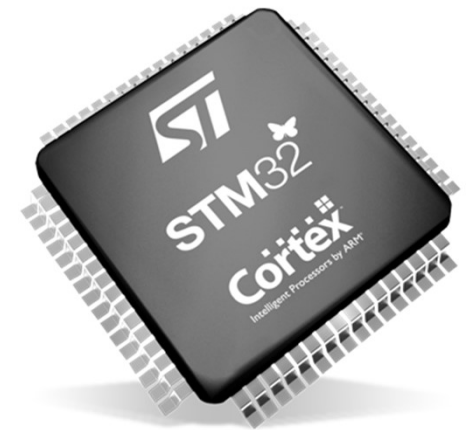
FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



Pull up/Pull Down

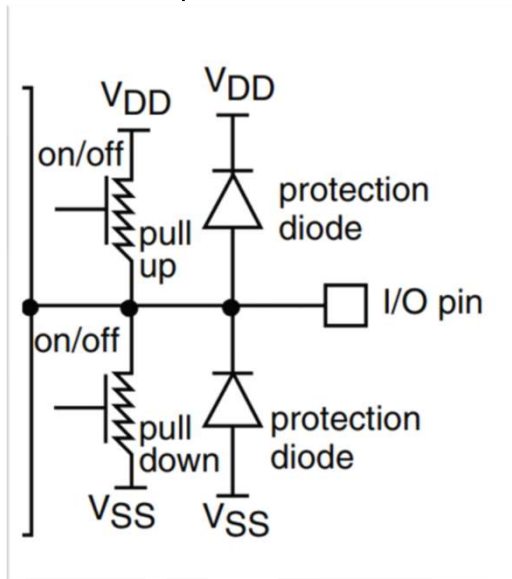


Some MCU Build-in
“Weak” Pull up/Pull Down



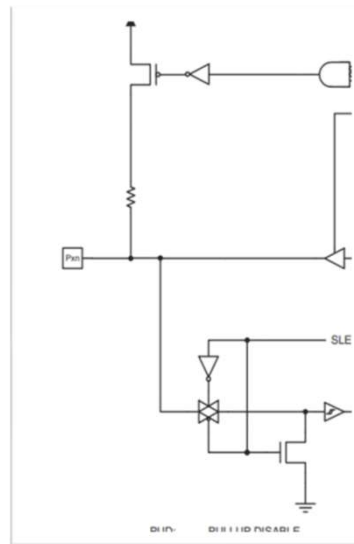
Pull up/Pull Down

Pull up + Pull down

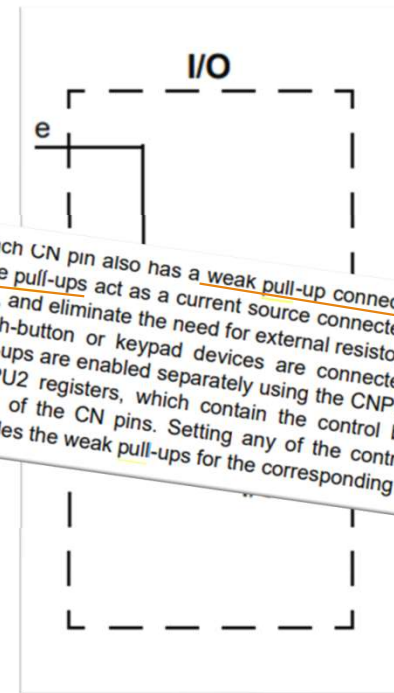


STM32 NUCLEO

Pull up only



AVR Arduino



Each CN pin also has a weak pull-up connected to it. The pull-ups act as a current source connected to the pin, and eliminate the need for external resistors when push-button or keypad devices are connected. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-ups for the corresponding pins.

dsPIC33

STM32
NUCLEO

Table 53. I/O static characteristics (continued)

Symbol	Parameter		Conditions	Min	Typ	Max	Unit
R _{PU}	Weak pull-up equivalent resistor ⁽⁶⁾	All pins except for PA10 (OTG_FS_ID)	V _{IN} = V _{SS}	30	40	50	kΩ
		PA10 (OTG_FS_ID)	-	7	10	14	
R _{PD}	Weak pull-down equivalent resistor ⁽⁷⁾	All pins except for PA10 (OTG_FS_ID)	V _{IN} = V _{DD}	30	40	50	
		PA10 (OTG_FS_ID)	-	7	10	14	
C _{IO} ⁽⁸⁾	I/O pin capacitance		-	-	5	-	pF

“WEAK Pull up”

AVR
Arduino

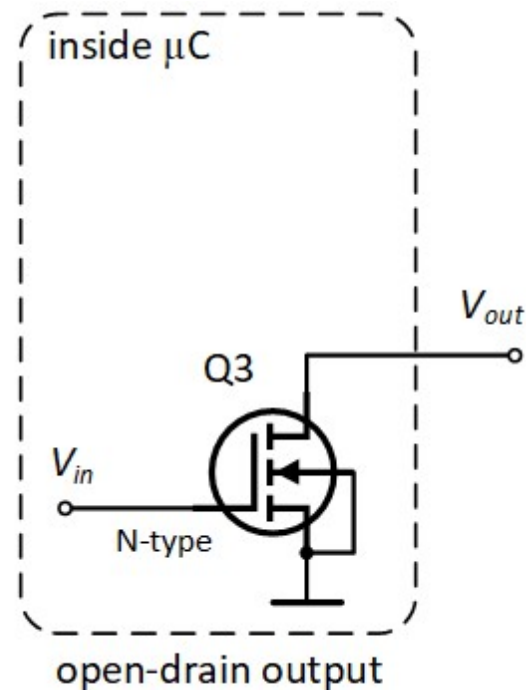
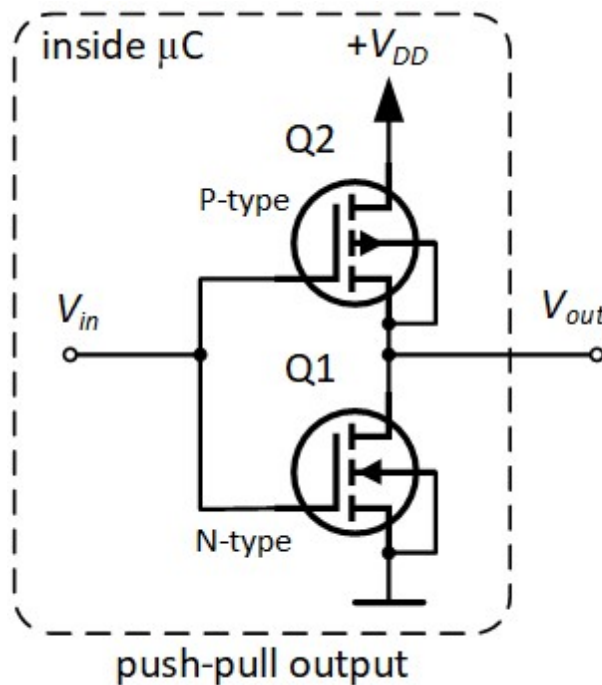
R _{PU}	I/O Pin Pull-up Resistor		20		50	kΩ
-----------------	--------------------------	--	----	--	----	----

dsPIC33

DI30	ICNPU	CNx Pull-up Current	50	250	400	μA	V _{DD} = 3.3V, V _{PIN} = V _{SS}
------	-------	---------------------	----	-----	-----	----	--

Push Pull / (Open Drain / Open Collector)

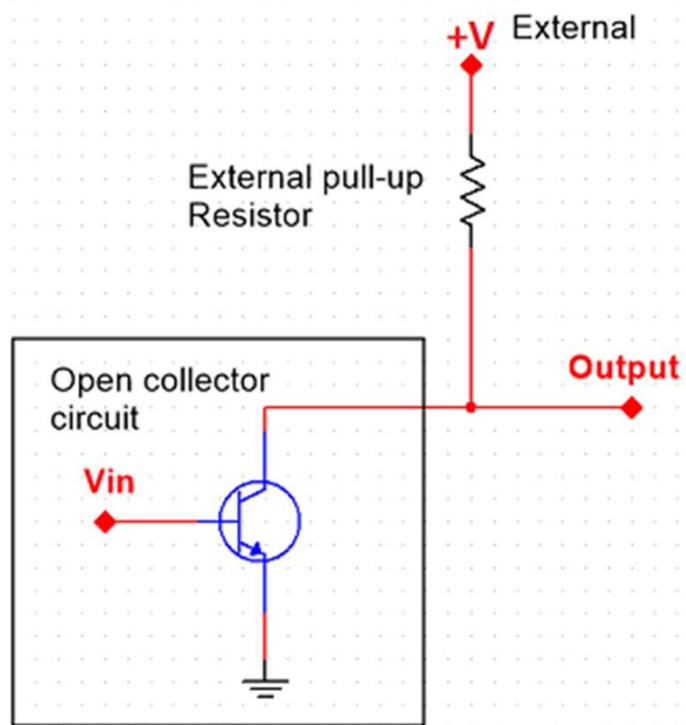
HI
LO



(High impedance)

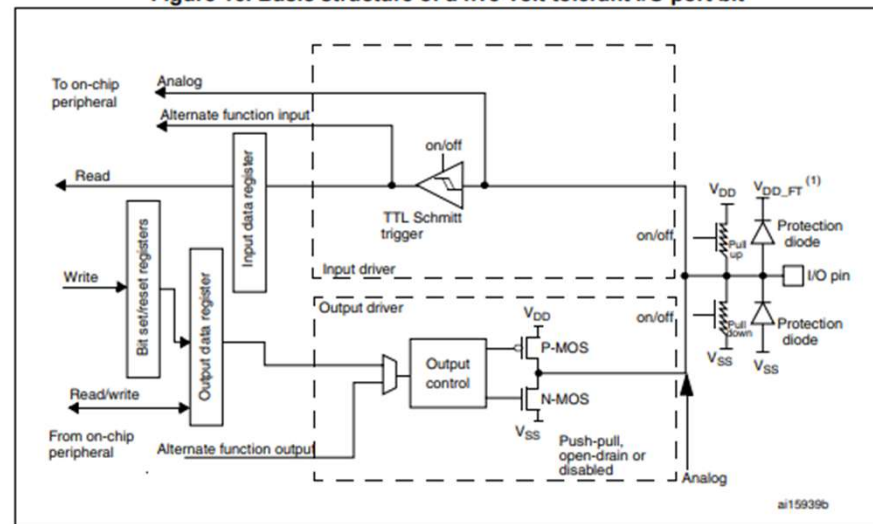
HI-Z
LO

(Open Drain /Open Collector)



+ การจ่ายไฟ Logic ที่มากกว่า VCC

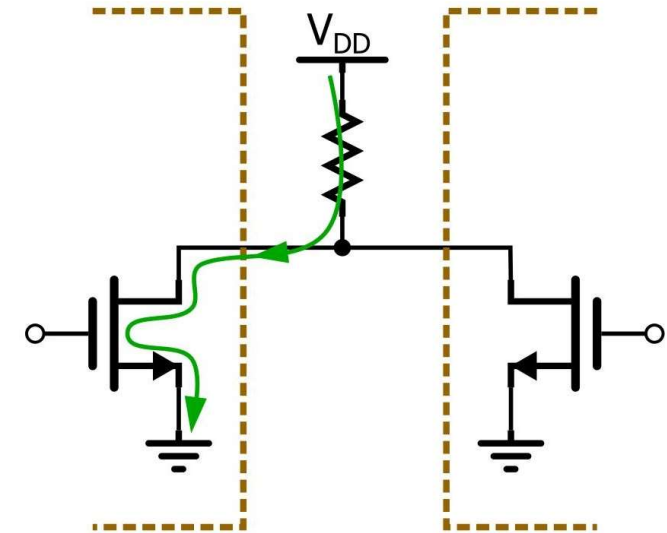
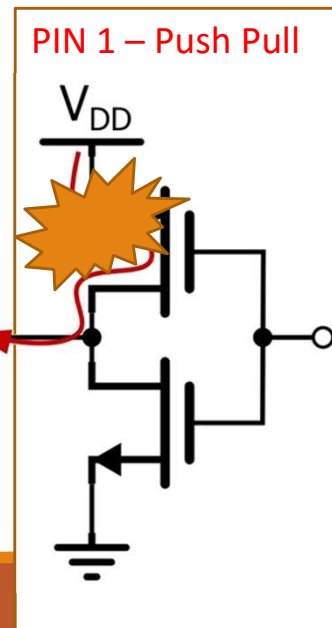
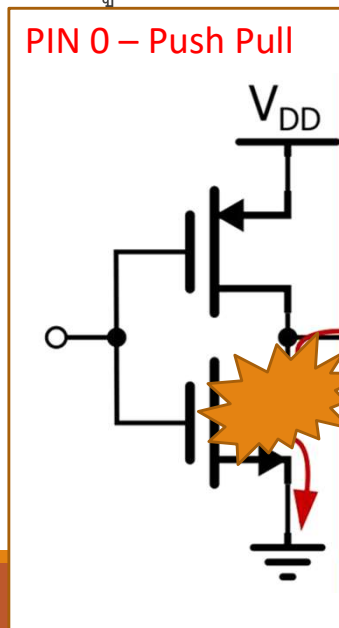
Figure 16. Basic structure of a five-volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD} .

(Open Drain / Open Collector)

- + การต่อ Multiple Output (เป็น AND Gate อย่างง่าย)
- Speed / Matching External Resistor
- + ถูกใช้ใน Protocol บางชนิด เช่น I2C , 1 wire



Output SPEED ???

Slew Rate

Lower Slew Rate

=

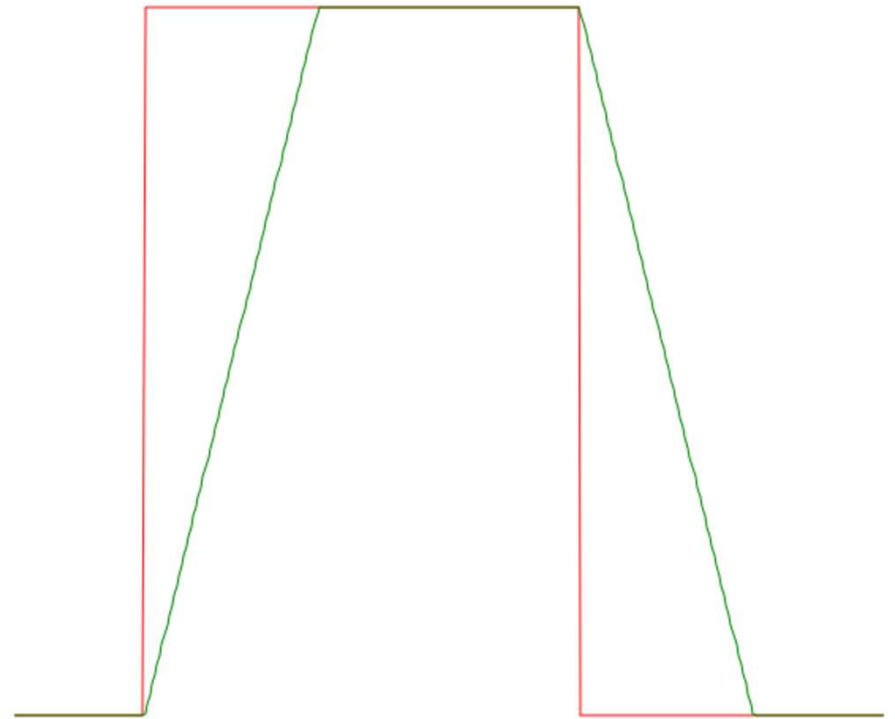
Lower EMI Emission

=

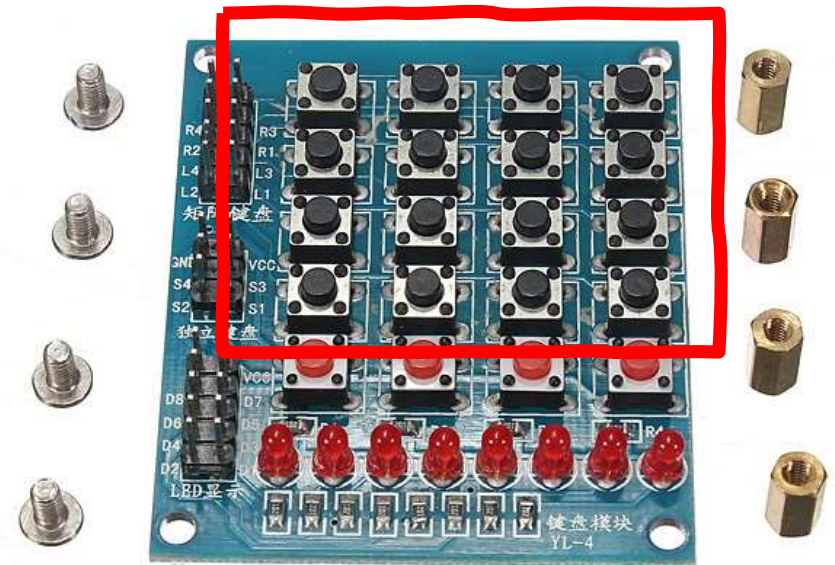
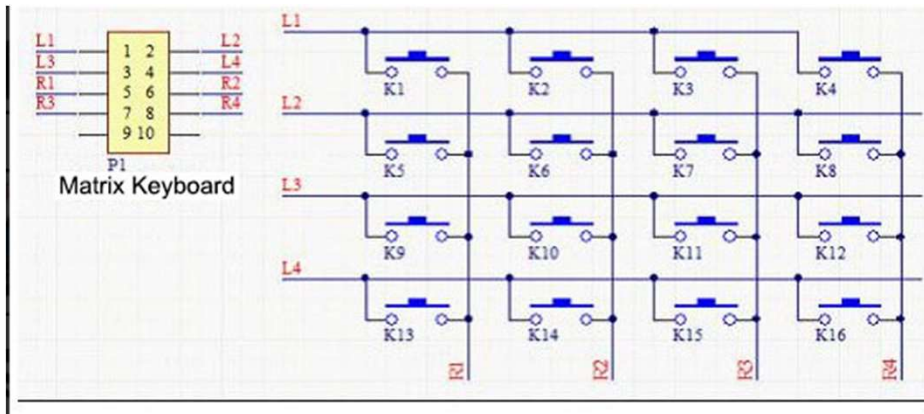
Lower Electronic Noise in System

=

Lower Maximum Frequency



Matrix Button

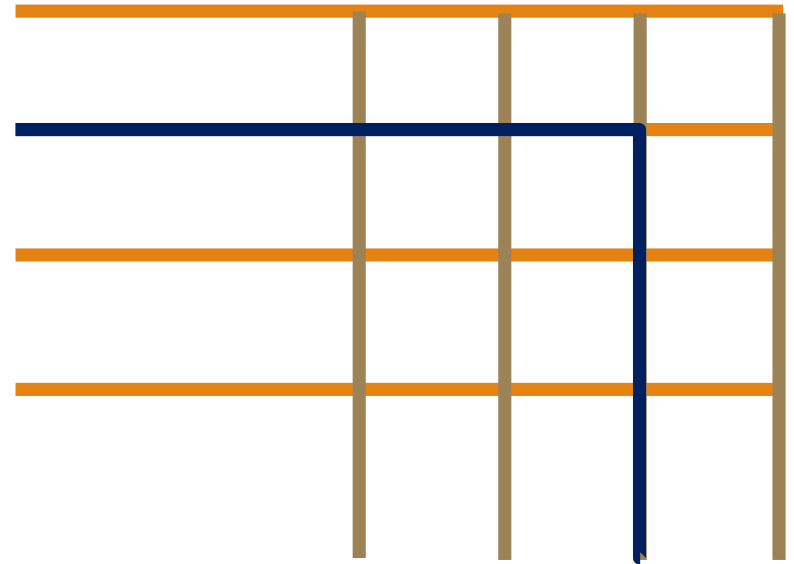
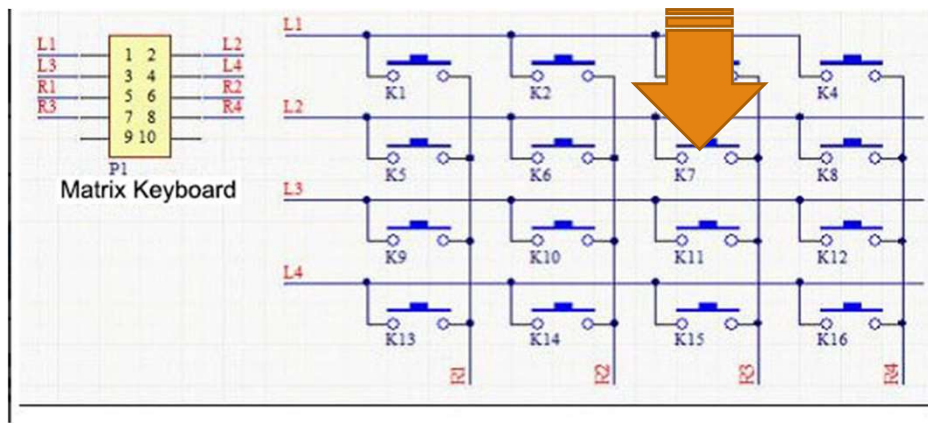


N x M Button with N+M Pin

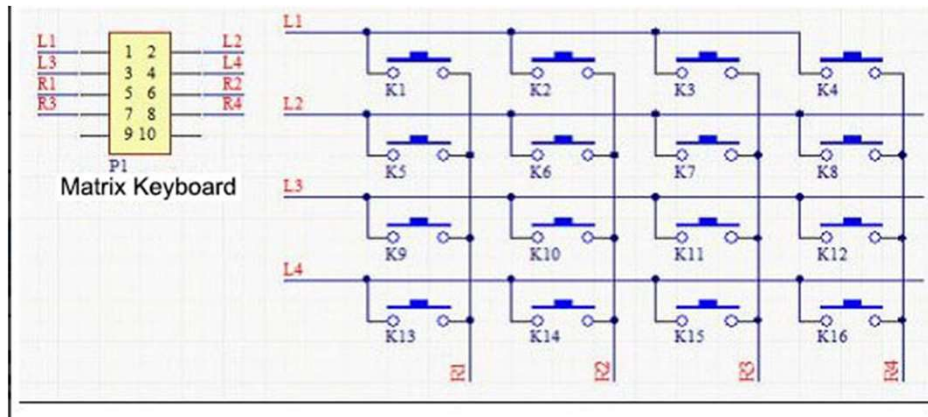
Lag input and use some MPU resource/time

Use in user interface

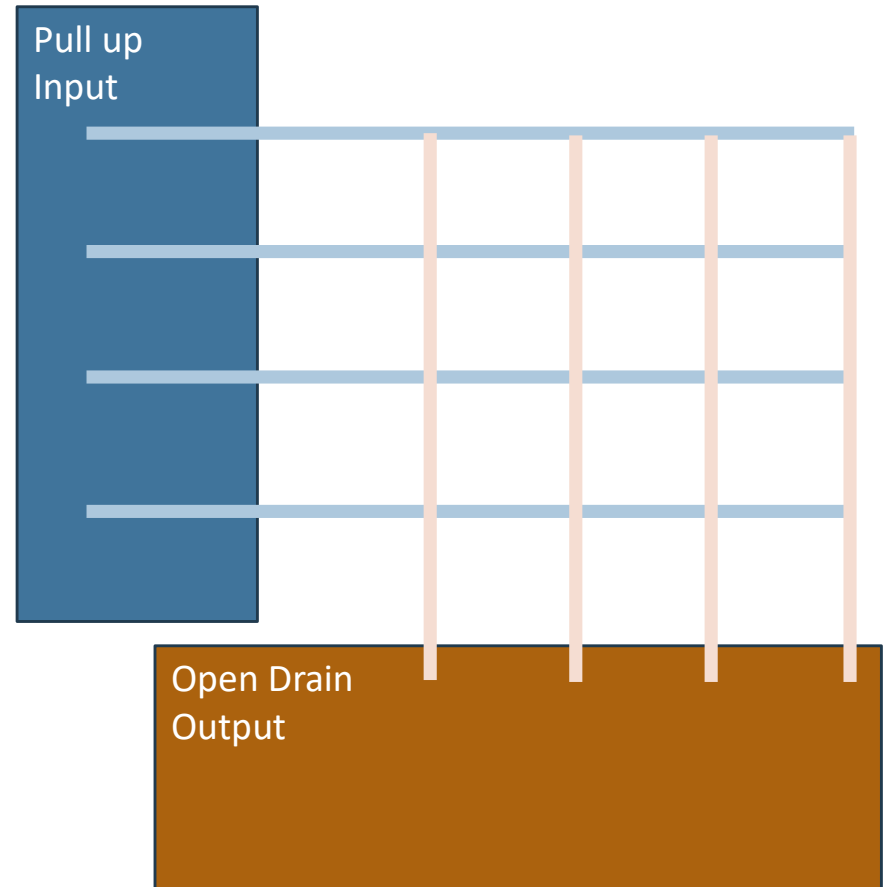
DO NOT USE in safety signal like emergency switch

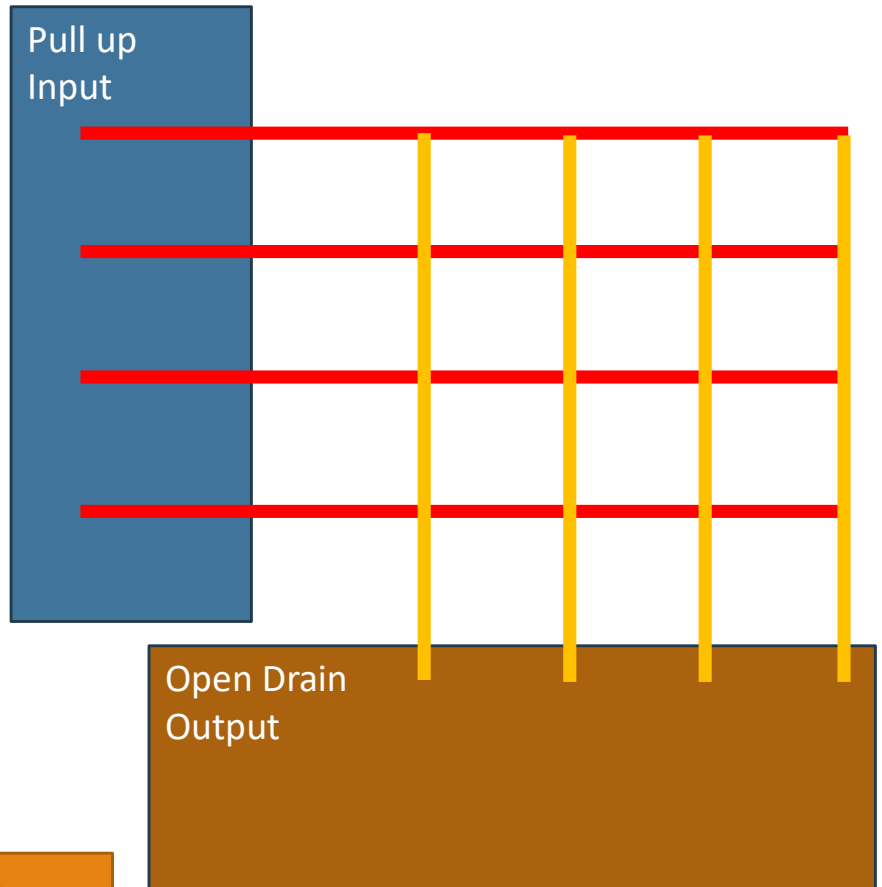
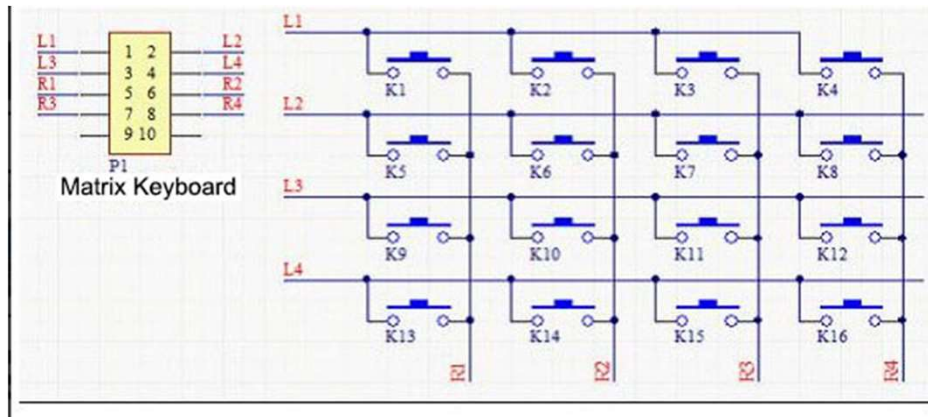


Press K7 Connect L2 And R3 Together



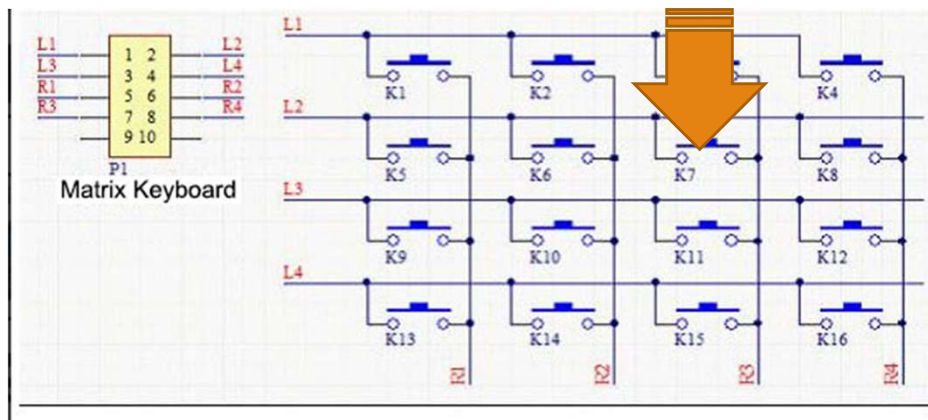
- Hi (Because Weak Pull up)
- Hi – Z (Floating)
- LO



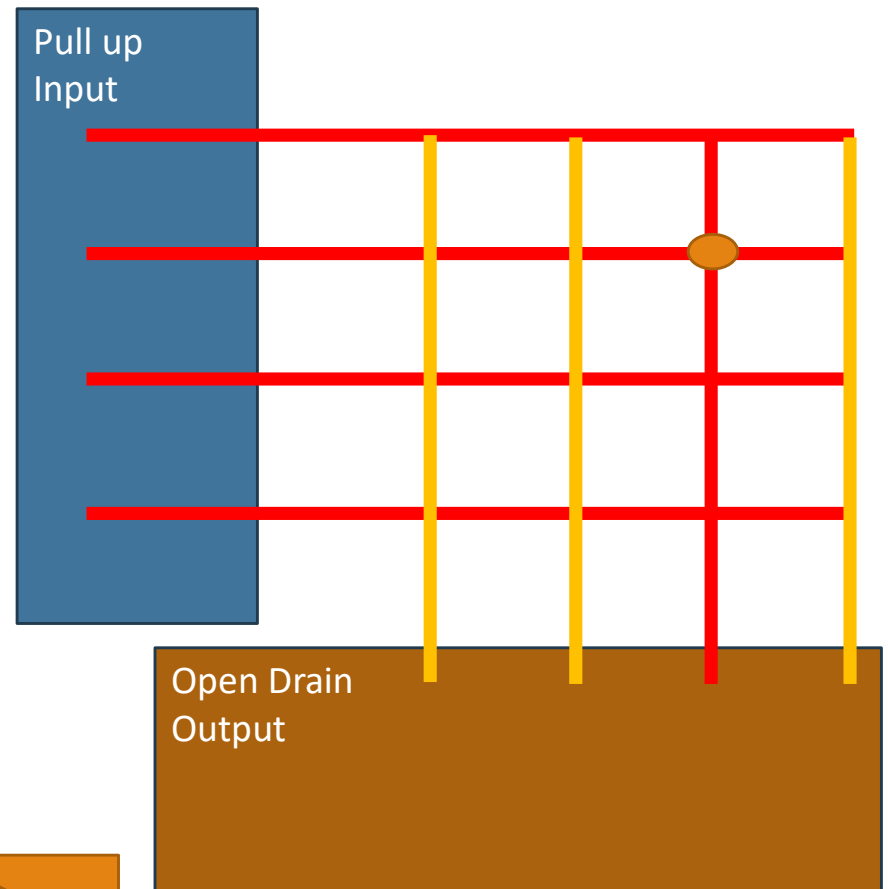


- Hi (Because Weak Pull up)
- Hi - Z (Floating)
- LO





- Hi (Because Weak Pull up)
- Hi – Z (Floating)
- LO



RESET R1
DELAY (For electronic change)
READ L1 – L4 SAVE TO BUTTON 0 – 3
SET R1 (For open drain is Hi - Z)

1111

RESET R2
DELAY
READ L1 – L4 SAVE TO BUTTON 4 – 7
SET R2

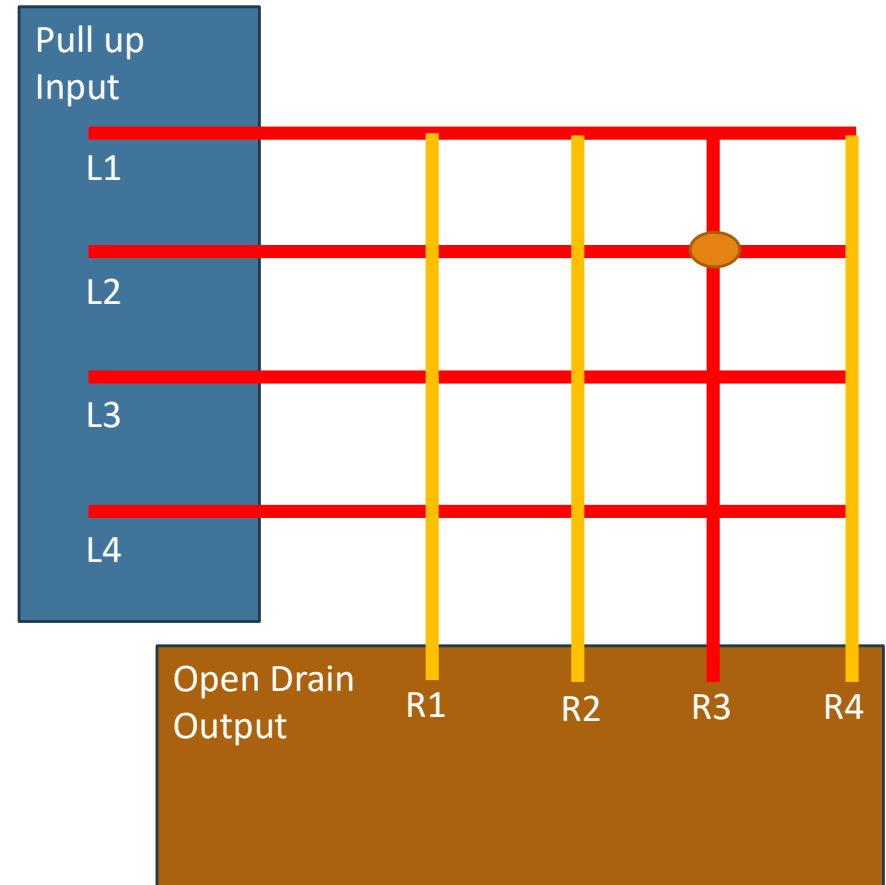
1111

RESET R3
DELAY
READ L1 – L4 SAVE TO BUTTON 8 – 11
SET R3

1011

RESET R4
DELAY
READ L1 – L4 SAVE TO BUTTON 12 – 15
SET R4

1111



Concept to program

RESET R1

DELAY (For electronic change)

READ L1 – L4 SAVE TO BUTTON 0 – 3

SET R1 (For open drain is Hi - Z)

RESET R2

DELAY

READ L1 – L4 SAVE TO BUTTON 4 –7

SET R2

RESET R3

DELAY

READ L1 – L4 SAVE TO BUTTON 8 –11

SET R3

RESET R4

DELAY

READ L1 – L4 SAVE TO BUTTON 12 – 15

SET R4



DELAY (For electronic change)

READ L1 – L4 SAVE TO BUTTON 0 – 3

SET R1 (For open drain is Hi - Z)

RESET R2

DELAY

READ L1 – L4 SAVE TO BUTTON 4 –7

SET R2

RESET R3

DELAY

READ L1 – L4 SAVE TO BUTTON 8 –11

SET R3

RESET R4

DELAY

READ L1 – L4 SAVE TO BUTTON 12 – 15

SET R4

RESET R1

DELAY (For electronic change)

READ L1 – L4 SAVE TO BUTTON 0 – 3

SET R1 (For open drain is Hi - Z)

RESET R2



DELAY

READ L1 – L4 SAVE TO BUTTON 4 –7

SET R2

RESET R3

DELAY

READ L1 – L4 SAVE TO BUTTON 8 –11

SET R3

RESET R4

DELAY

READ L1 – L4 SAVE TO BUTTON 12 – 15

SET R4

RESET R1

1. DELAY → ไปทำอย่างอื่นรอ → เรียกฟังก์ชันด้วยความถี่คงที่

2. R1 R2 R3 R4 → R0 R1 R2 R3

3. create X = 0, 1, 2, 3

Function() {

Static int **X**;

READ L1 – L4 SAVE TO BUTTON **X***4 to **X***4+3

SET R(**X**) (For open drain is Hi - Z)

RESET R(**X**+1) %4

X++

X %= 4

}

อัปเดต 1 แถว / การเรียก 1 ครั้ง

READ L1 – L4

SET R(X)

RESET R(X+1) %4

SAVE TO BUTTON $X*4$ to $X*4+3$

Structure PortPin
-PORT
-PIN



Array of PortPin
PortPin R[4];
PortPin L[4];

SAVE TO BUTTON (X)



uint16_t
1 บิต = 1 ไบต์
Use bitwise operation

TASK

0. ต่อสายและอุปกรณ์

1. สร้าง **structure** เพื่อเก็บค่าของ **button matrix** ทั้งหมด

2. สร้าง **function** สำหรับอ่านปุ่มที่ละแถว

3. รัน **function** นั้นด้วยความถี่คงที่

4. กระพริบไฟไปด้วยยยยย

5. ดูค่าตัวแปรผ่าน **DEBUGGER**

Example 2

-สร้างปุ่มกด ขนาด **4 x 4** โดยแต่ละปุ่มจะถูกแปลงเป็น ข้อมูล 1 บิต ใน ตัวแปร ขนาด **uint16_t**
โดยปกติ ทุกๆบิตจะมีค่า 0 เมื่อมีปุ่มใดถูกกด บิตที่เกี่ยวข้องกับปุ่มนั้นๆ จะมีค่า 1

uint16_t ButtonState

0000 0000 0100 1001 = K1 ,K4 , K7ถูกกด

TASK

0. ต่อสายและอุปกรณ์

1. สร้าง **structure** เพื่อเก็บค่าของ **button matrix** ทั้งหมด

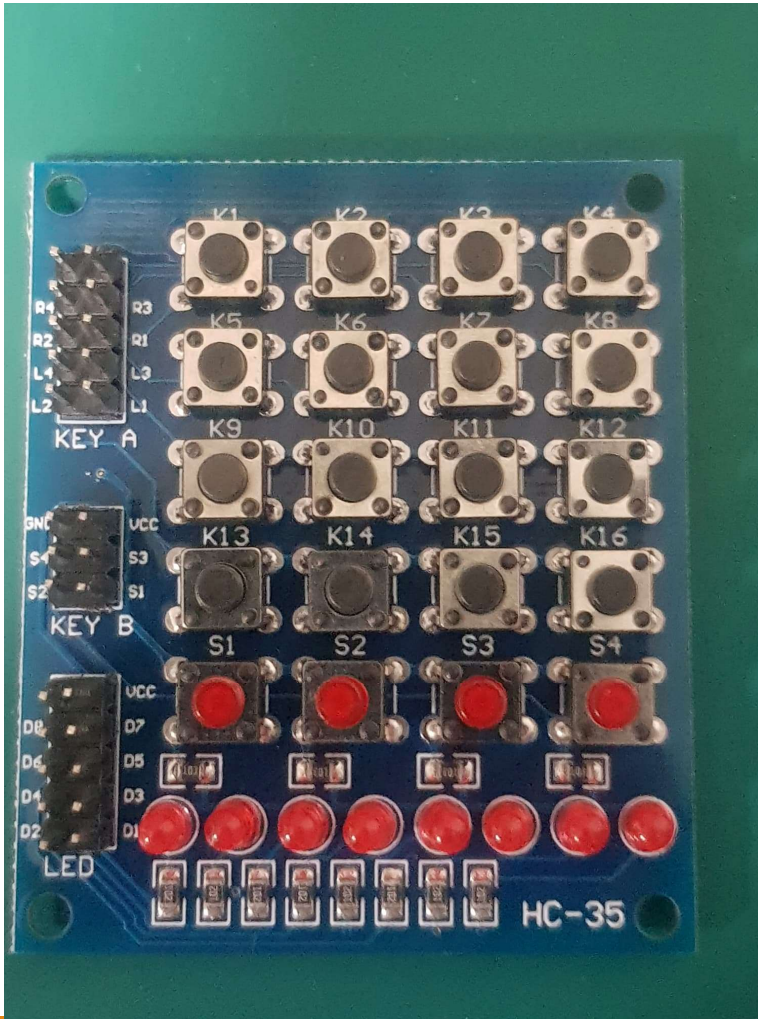
2. สร้าง **function** สำหรับอ่านปุ่มที่ละแถว

3. รัน **function** นั้นด้วยความถี่ที่ 50Hz

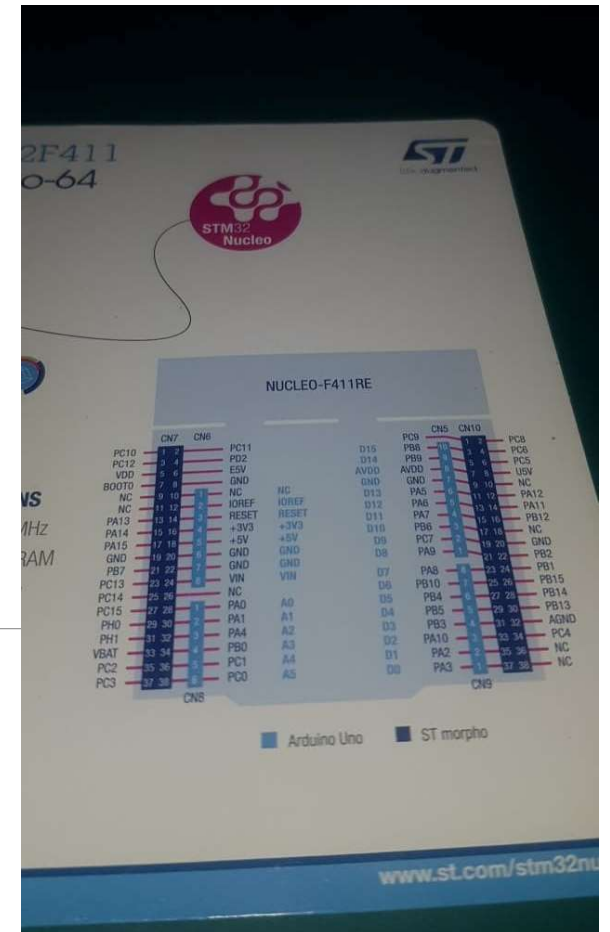
4. กระพริบไฟไปด้วยยยย 5Hz

5. ดูค่าตัวแปรผ่าน **DEBUGGER**

เข้ามาแล้ว ต่อสายอุปกรณ์รอได้เลย



L1 – PA9 - Input
L2 – PC7
L3 – PB6
L4 – PA7
R1 – PA10 Output
R2 – PB3
R3 – PB5
R4 – PB4



Pull up

Input

L1

L2

L3

L4

Open Drain

Output

R1

R2

R3

R4



READ L1 – L4

SET R(X)

RESET R(X+1) %4

SAVE TO BUTTON $X*4$ to $X*4+3$

Structure PortPin
-PORT
-PIN



Array of PortPin
PortPin R[4];
PortPin L[4];

SAVE TO BUTTON (X)



uint16_t
1 บิต = 1 ไบต์
Use bitwise operation

```
Function() {  
  Static int X;
```

```
  READ L1 – L4 SAVE TO BUTTON X*4 to X*4+3
```

```
  SET R(X) (For open drain is Hi - Z)
```

```
  RESET R(X+1) %4
```

```
  X++
```

```
  X %= 4
```

```
}
```

TASK

0.ต่อสายและอุปกรณ์

1.สร้างstructure เพื่อเก็บค่าของ button matrix ทั้งหมด

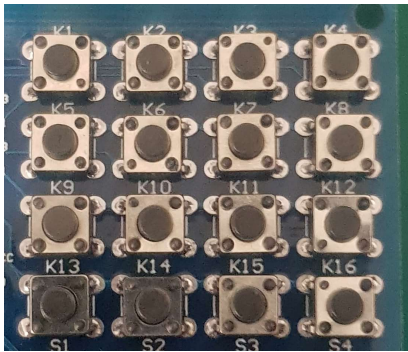
2. สร้างfunction สำหรับอ่านปุ่มทีละแถว

3. รัน function นั้นด้วยความถี่ที่ 100Hz

4. กระพริบไฟไปด้วยยยย 5Hz

5.ดูค่าตัวแปรผ่าน DEBUGGER

LAB 1 Key pad (Easy)



7	8	9	Clear
4	5	6	BS<<
1	2	3	
0			ok

- กดรหัสนักศึกษาตัวเอง ให้ถูกต้อง เมื่อถูกต้อง แล้ว กด **ok** จะทำให้ไฟ LD2 บนบอร์ดจะติด
- ไม่มีการกดมั่วๆ แล้ว ไฟติด ถ้ากดผิดจะต้องกด **Clear** แล้ว กดใหม่ให้ถูกต้องทั้งหมด เท่านั้น
- (ยัง)ไม่มีการกด2 ปุ่มพร้อมๆ กัน

[Challenge] มีปุ่ม **backspace** ลบข้อมูลที่กดผิดได้โดยไม่ต้องกด **Clear**

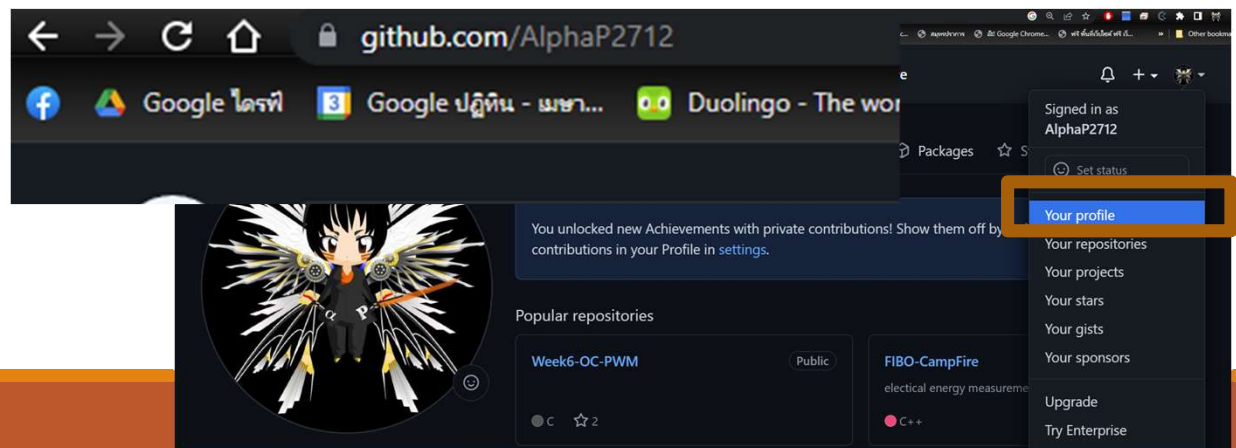
LAB!!

Due date

- SEC B – WED 1/2/2023 20:00
- SEC A – WED 2/2/2023 20:00

วิธีการส่งแลป

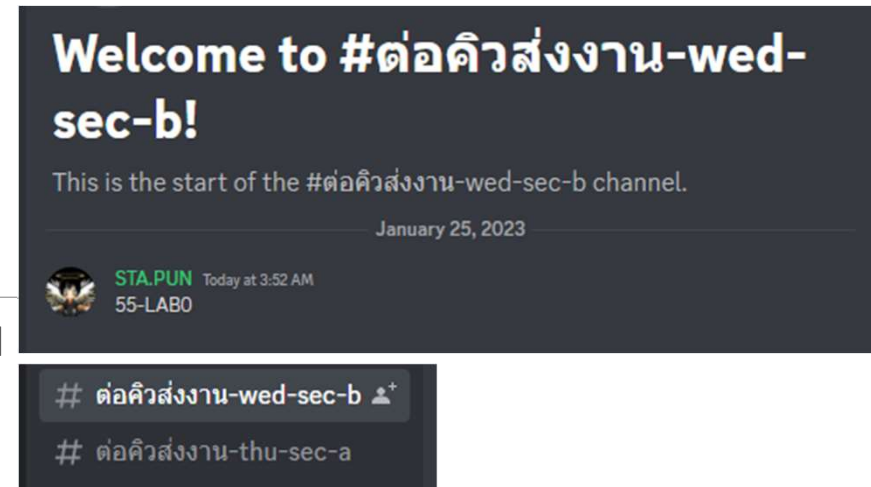
1. ทำแลปใน Github Repository แยกเล่มขอ และตั้ง Public ไปด้วย !!!
2. ส่ง link ไปยัง github profile ของตนเองมาใน Google form ใน classroom



LAB!!

3. ถ้าทำเสร็จ และต้องการตรวจ ให้ลงชื่อขอส่งงานที่ ห้อง ต่อคิวส่งงาน ในdiscord

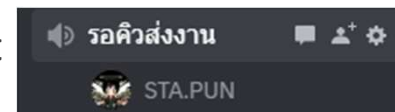
- - ระบุ เลขที่และแลปที่ต้องการส่ง
- ปิดคิวเวลา 20.00น. ของวันส่ง ให้ลงคิวหลังจากนั้นไม่รับตรวจ



4. ถ้าส่งในห้อง ให้เตรียมตัวให้เรียบร้อย ถ้ามีเวลาว่างหลังคาบเรียน พี่ๆTA จะตามเรียกหาตามลำดับคิว ไม่อยู่ขอข้าม
ถ้าส่งในห้องไม่ทัน จะส่งออนไลน์ ผ่าน Discord ช่วงเวลา 20.00 -22.00 น.ของวันที่ตนเองเรียน

5. ถ้าส่งในDiscord ให้เตรียมตัวออนไลน์และ join voice chat

- อย่าลืมตั้งเลขที่ ชื่อเล่นและชื่อจริงเอาไว้ด้วย
- ถ้ามี ID สำหรับ กล้อง เช่น กล้องโทรศัพท์ ให้ระบุ เลขที่และชื่อเหมือนกันกับเจ้าของแล้วต่อท้ายด้วย – cam เช่น 55 – พุฒินาถ – บั๊น – cam
- มาก่อนเวลาเริ่มนะครับ ไม่อยู่ขอข้าม



LAB!!

6. เมื่อถึงเวลาให้ **TA** ลากเข้าห้องเองโดยอัตโนมัติ เปิดลำโพงไว้รอฟังเสียงด้วยนา

- มีเวลาประมาณ คนละ 15 นาที +-
 - อธิบาย **lab** แนวคิดคร่าวๆ และ โค้ดที่ทำ
 - แสดงการทำงานให้ **TA** ดู และขั้นตอนการ**Dev** โค้ดของเรา
 - จะต้องมี **commit** และ**push origin** ของโค้ดที่ใช้ส่ง ซึ่งลงเวลาเอาไว้ก่อน 20.00น. ของวันที่ส่ง
 - ตอบคำถามนิดหน่อย

7. ถ้าส่งแล้ว จะได้รับ **reaction** ใน **discord**

