NuMicro® GPIO

M031/M480 Seviye 1

Çeviren ve Anlatan: **Doç. Dr. Barış GÖKÇE**

Aralık 2024

Not: Bu doküman'ın telif hakkı tamamen Nuvoton'a aittir. Bu eğitim dökümanı mikrodenetleiyici eğitimi kapsamında Türkçe'ye orjinalinden çevrilmiştir. Dökümanın orijinali inngilizce'dir ve oluşan çeviri hataları tamamen Dr. Barış GÖKÇE'ye aittir.

Copyright © Nuvoton Technology Corporation | Photo by Jeffery Cheng

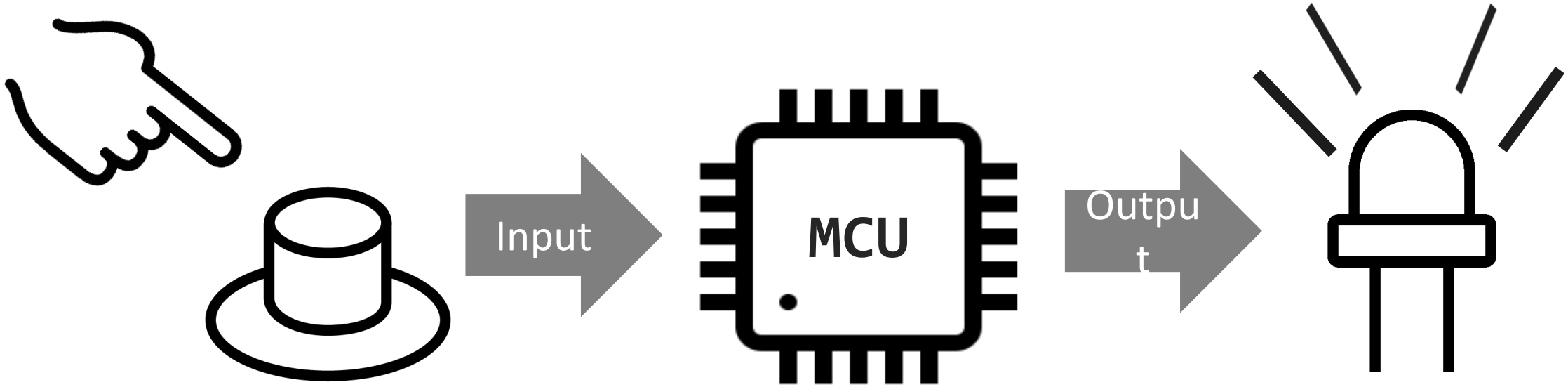
Agenda

- GPIO nedir?
- Özellikler
- Blok Diyagram
- GPIO Modu
- GPIO Fonksiyonları
- Örnek kod

Copyright © 2021 Nuvoton Technology Corporation

GPIO Nedir?

- General Purpose Input Output (Genel Amaçlı Giriş Çıkış)



Copyright © 2021 Nuvoton Technology Corporation

Özellikler

- 4 adet I/O modu:
  - Sadece yüksek empedanslı girişler, Push-Pull Output, Open-Drain Output, Quasi-bidirectional
- Tüm pinlerin yapılandırılabilir olarak sayılan I/O modu
  - tri-state or Quasi-bidirectional
- İnterrupt (kesme) ve wake-up (uyandırma) fonksiyonlarını detekleyen tüm pinler
  - Level trigger or Edge trigger
- TTL/Schmitt* trigger input selectable

* : M031 only Schmitt trigger input

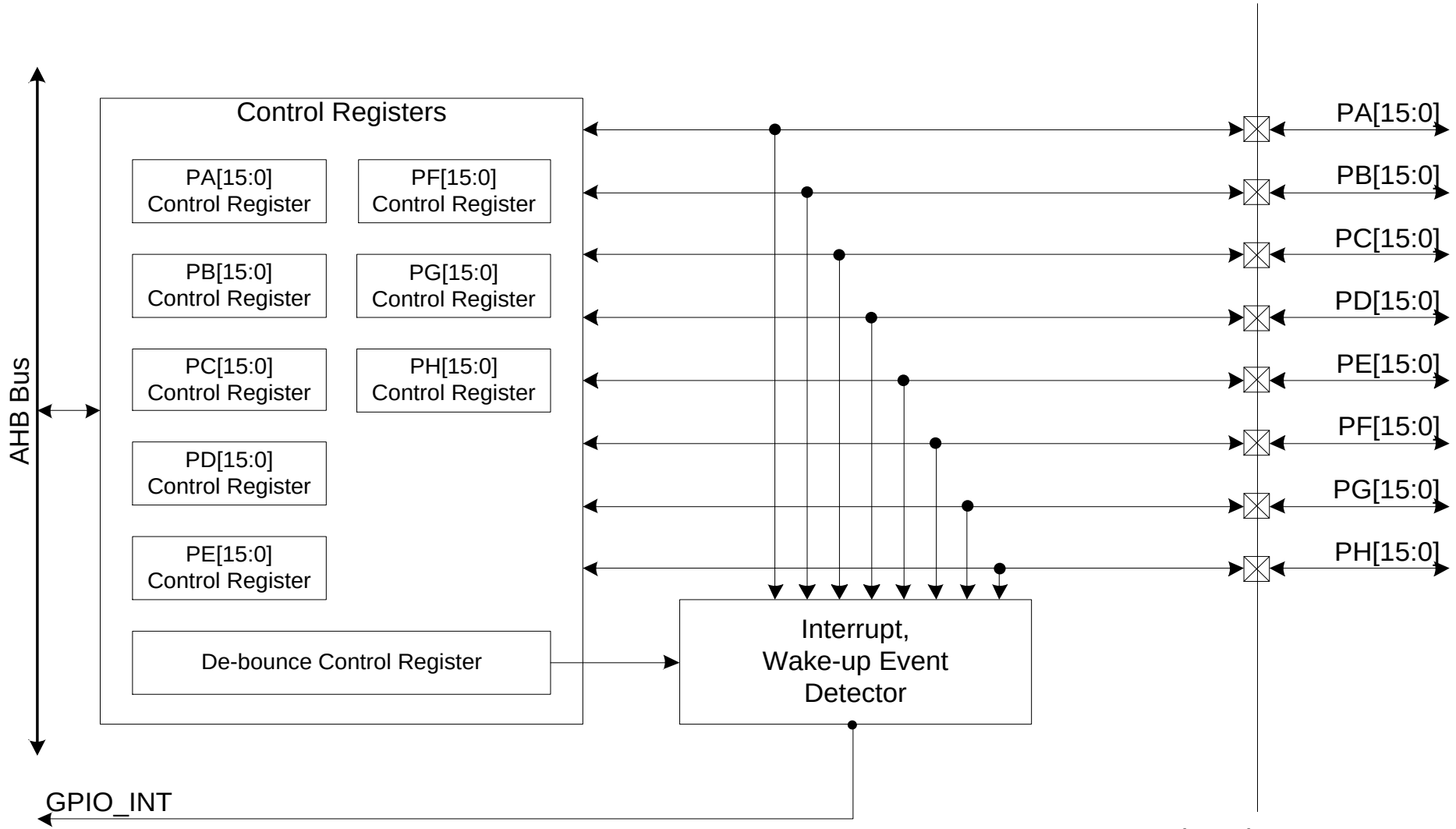
Copyright © 2021 Nuvoton Technology Corporation

Nuvoton'da Tri-State veya Quasi-Bidirectional GPIO Nedir?

- Nuvoton mikrodenetleyicilerinde Tri-State veya Quasi-Bidirectional GPIO (Genel Amaçlı Giriş/Çıkış), normal giriş/çıkış pinlerinin özelliklerini genişleten bir konfigürasyondur. Bu konfigürasyon, pinin üç farklı durumu almasına izin verir:
  1. **Giriş Modu:** Pin, dışarıdan gelen sinyali okuyabilir.
  2. **Çıkış Modu:** Pin, mikrodenetleyici tarafından kontrol edilir ve belirli bir seviye (genellikle yüksek veya düşük) çıkarır.
  3. **Yüksek Empedans Modu (Hi-Z):** Pin, giriş veya çıkış olarak aktif değildir. Bu modda, pinin üzerindeki voltaj, harici bir kaynaktan etkilenir veya "yüzen" (floating) olarak kalır.
- **Quasi-Bidirectional (Yarı Çift Yönlü)** terimi, genellikle tri-state özelliğine benzer bir işlevi ifade eder. Ancak, bazı mikrodenetleyicilerde, "quasi-bidirectional" terimi, giriş/çıkış pininin hem giriş hem de çıkış olarak kullanılabildiği, ancak aynı anda her iki yönde de aktif olamayacağı durumları tanımlamak için kullanılabilir.

Copyright © 2021 Nuvoton Technology Corporation

Blok Diyagram

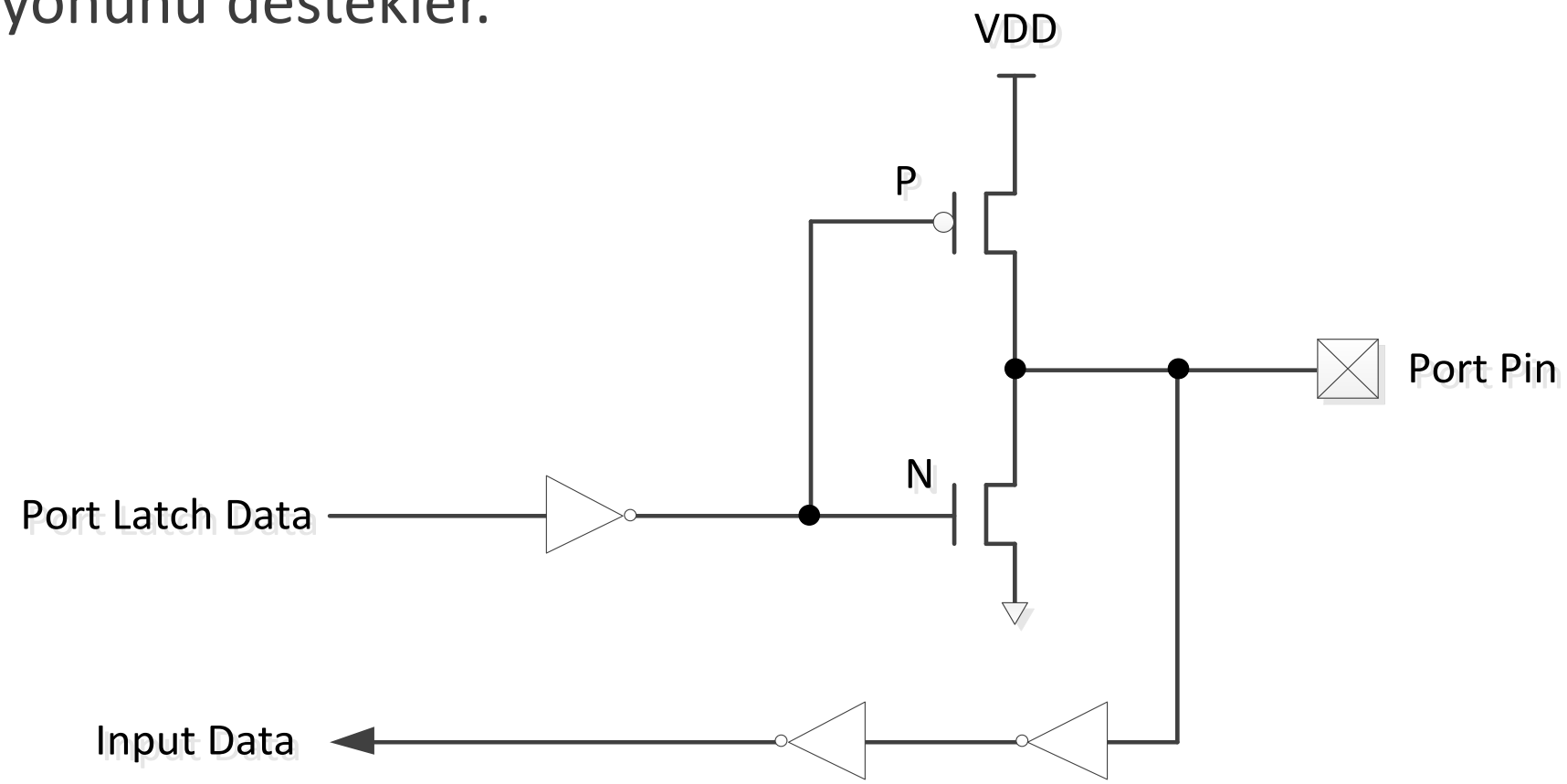


Note: M031 PH/PG/PC.8~PC.13/PC.15/PD.4~PD.14/ PE.0~PE.15/PF.7~PF13 pins are ignored.

Copyright © 2021 Nuvoton Technology Corporation

Push-pull Output Modu

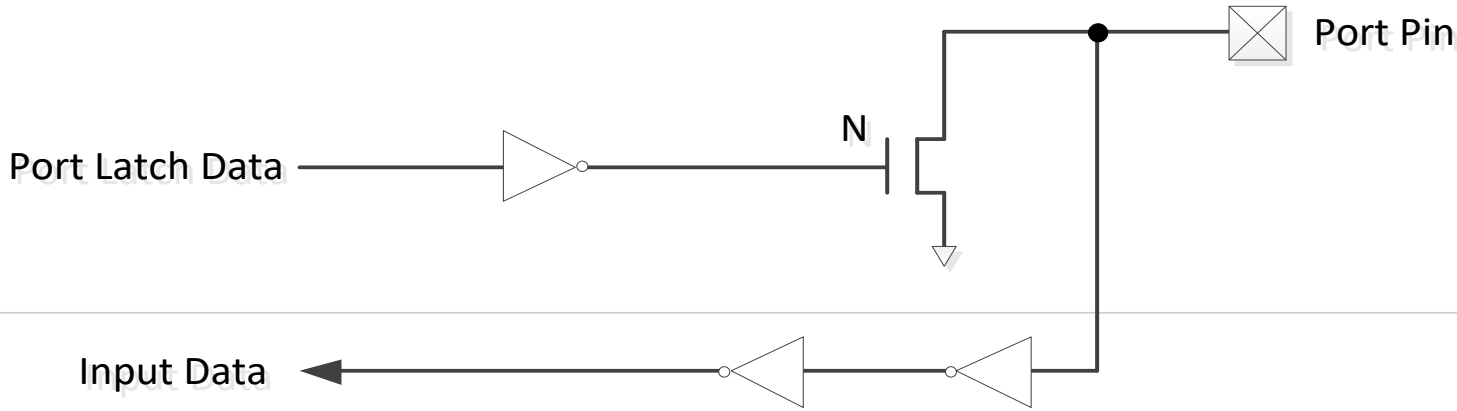
- I/O pini soruce/sink (kaynak/batarya) akımı kapasitesiyle dijital çıkış fonksiyonunu destekler.



Copyright © 2021 Nuvoton Technology Corporation

Open-drain Modu

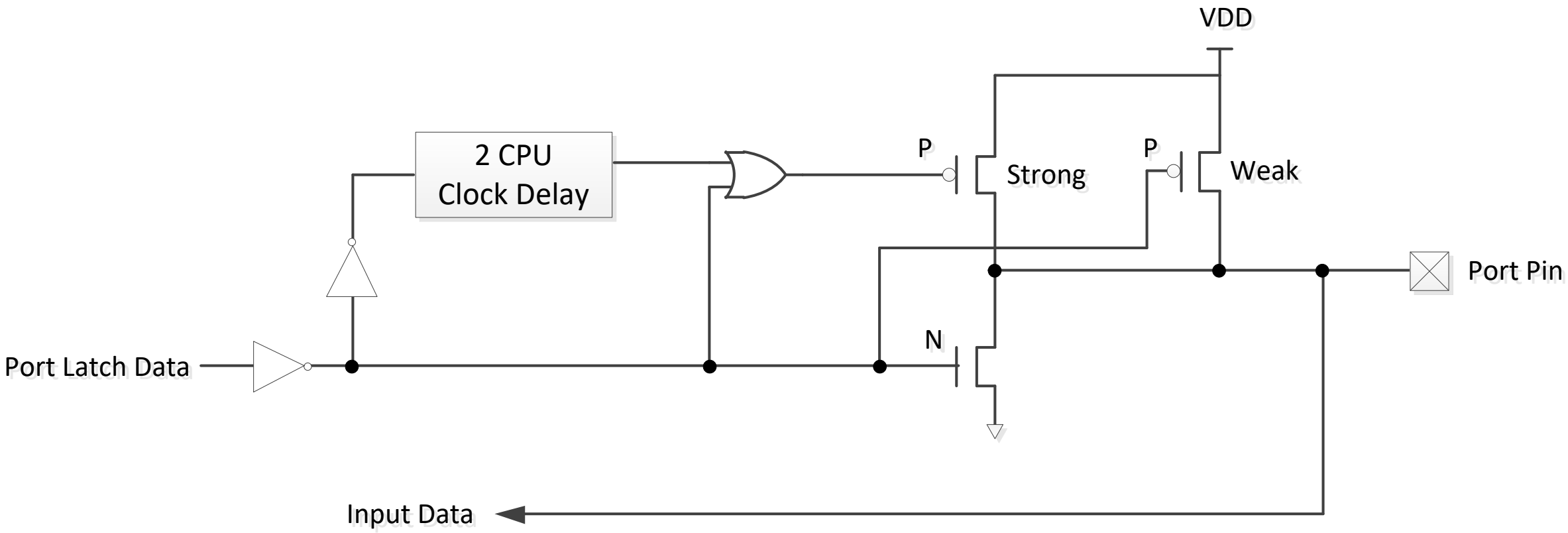
- **Open-Drain Çıkış** (Açık Drenaj Çıkışı), Çıkış pini bir N-MOSFET transistörünün drenajına bağlanmıştır ve bu transistörün kaynağı topraklanmıştır. Yani, çıkış pini düşük seviyeye çekilebilir ancak yüksek seviyeye çekemez. Yüksek seviyeye çekmek için harici bir pull-up direnci kullanılması gerekmektedir.
- **NMOS Sink Akım Kapasitesi**, NMOS transistörünün, çıkışını düşük seviyeye çekmek için sağlayabileceği maksimum akım değeridir.



Copyright © 2021 Nuvoton Technology Corporation

Quasi-bidirectional Modu

- Dijital giriş ve çıkış işlevlerini aynı anda destekler



Copyright © 2021 Nuvoton Technology Corporation

GPIO mode

§ GPIO_SetMode()

```
void GPIO_SetMode ( GPIO_T * port,
                    uint32_t u32PinMask,
                    uint32_t u32Mode
                  )
```

Set GPIO operation mode.

Parameters

[in]	**port**	GPIO port. It could be It could be PA, PB, PC, PD, PE, PF, PG or PH.
[in]	**u32PinMask**	The single or multiple pins of specified GPIO port. It could be BIT0 ~ BIT15 for PA, PB, PC, PD, PF and PH GPIO port. It could be BIT0 ~ BIT13 for PE GPIO port. It could be BIT0 ~ BIT11 for PG GPIO port.
[in]	**u32Mode**	Operation mode. It could be GPIO_MODE_INPUT, GPIO_MODE_OUTPUT, GPIO_MODE_OPEN_DRAIN, GPIO_MODE_QUASI.

Returns

None

```
/* Set PC.3 ~ PC.5 to GPIO output */
GPIO_SetMode(PC, (BIT3 | BIT4 | BIT5), GPIO_MODE_OUTPUT);
```


Copyright © 2021 Nuvoton Technology Corporation

GPIO interrupt

◆ GPIO_EnableInt()

```
void GPIO_EnableInt ( GPIO_T *  port,
                      uint32_t  u32Pin,
                      uint32_t  u32IntAttribs
                    )
```

Enable GPIO interrupt.

Parameters

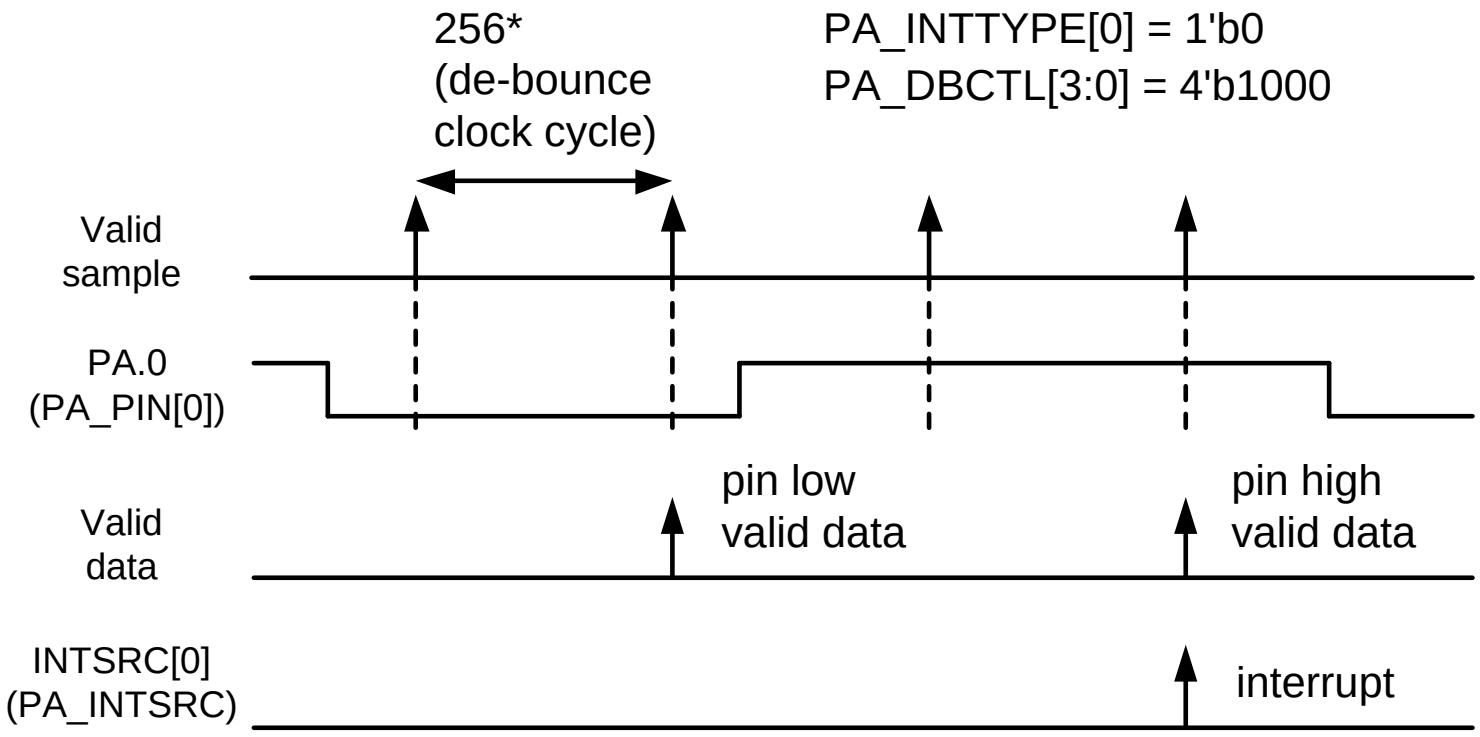
[in]	**port**	GPIO port. It could be PA, PB, PC, PD, or PF.
[in]	**u32Pin**	The pin of specified GPIO port. It could be 0 ~ 15 for PA and PB. It could be 0 ~ 7, and 14 for PC. It could be 0 ~ 3, and 15 for PD. It could be 0 ~ 6, 14, and 15 for PF.
[in]	**u32IntAttribs**	The interrupt attribute of specified GPIO pin. It could be • **GPIO_INT_RISING** • **GPIO_INT_FALLING** • **GPIO_INT_BOTH_EDGE** • **GPIO_INT_HIGH** • **GPIO_INT_LOW**

```
/* Enable PB.4 interrupt with falling edge trigger */
GPIO_EnableInt(PB, 4, GPIO_INT_FALLING);
```


Copyright © 2021 Nuvoton Technology Corporation

GPIO De-bounce Fonksiyonu

- GPIO de-bounce fonksiyonu gürültü kaynaklı beklenmedik kesintilerin önlenmesini sağlar.



Copyright © 2021 Nuvoton Technology Corporation

GPIO De-bounce Fonksiyonu

◆ GPIO_ENABLE_DEBOUNCE

```
#define GPIO_ENABLE_DEBOUNCE ( port,
                               u32PinMask
                             )
```

Enable Pin De-bounce Function.

Parameters

[in] **port** GPIO port. It could be PA, PB, PC, PD, or PF.

[in] **u32PinMask** The single or multiple pins of specified GPIO port. It could be BIT0 ~ BIT15 for PA and PB. It could be BIT0 ~ BIT7, and BIT14 for PC. It could be BIT0 ~ BIT3, and BIT15 for PD. It could be BIT0 ~ BIT6, BIT14, and BIT15 for PF.

Returns

None

§ GPIO_SET_DEBOUNCE_TIME

```
#define GPIO_SET_DEBOUNCE_TIME ( u32ClkSrc,
                                 u32ClkSel
                               )
```

Set De-bounce Sampling Cycle Time.

Parameters

[in] **u32ClkSrc** The de-bounce counter clock source. It could be GPIO_DBCTL_DBCLKSRC_HCLK or GPIO_DBCTL_DBCLKSRC_LIRC.

[in] **u32ClkSel** The de-bounce sampling cycle selection. It could be

- GPIO_DBCTL_DBCLKSEL_1
- GPIO_DBCTL_DBCLKSEL_2
- GPIO_DBCTL_DBCLKSEL_4
- GPIO_DBCTL_DBCLKSEL_8
- GPIO_DBCTL_DBCLKSEL_16
- GPIO_DBCTL_DBCLKSEL_32
- GPIO_DBCTL_DBCLKSEL_64
- GPIO_DBCTL_DBCLKSEL_128
- GPIO_DBCTL_DBCLKSEL_256
- GPIO_DBCTL_DBCLKSEL_512
- GPIO_DBCTL_DBCLKSEL_1024
- GPIO_DBCTL_DBCLKSEL_2048
- GPIO_DBCTL_DBCLKSEL_4096
- GPIO_DBCTL_DBCLKSEL_8192
- GPIO_DBCTL_DBCLKSEL_16384
- GPIO_DBCTL_DBCLKSEL_32768

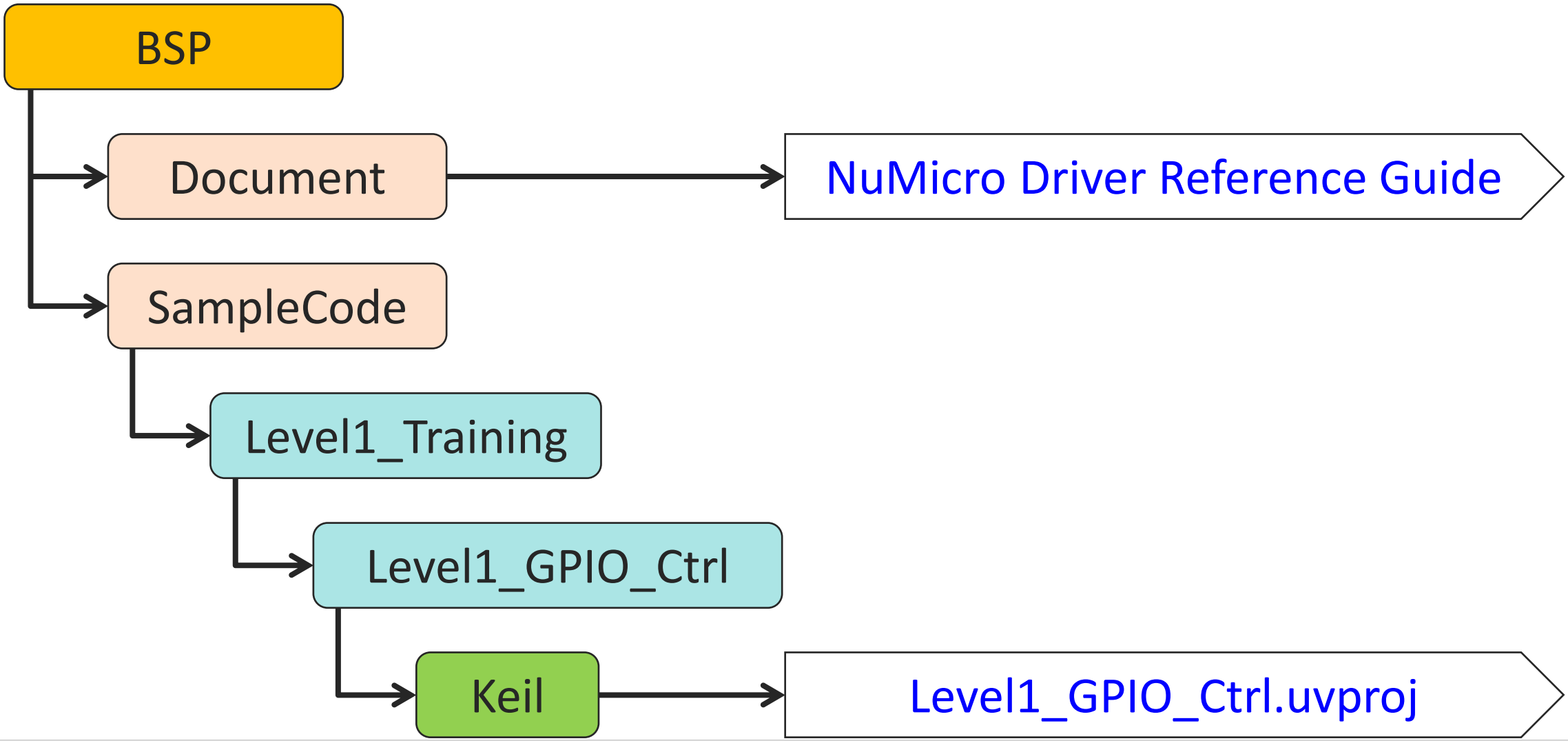
```
/* Set de-bounce function */
GPIO_SET_DEBOUNCE_TIME(GPIO_DBCTL_DBCLKSRC_LIRC, GPIO_DBCTL_DBCLKSEL_512);
GPIO_ENABLE_DEBOUNCE(PB, BIT4);
```


Copyright © 2021 Nuvoton Technology Corporation

Örnek Kod

LED'I debounce olmadan yakıp söndürme

Example - Path



Copyright © 2021 Nuvoton Technology Corporation

Örnek Fonksiyon

- LED_R On / Off'u kontrol etmek için SW1'e bas
  - De-bounce ile
- LED_G On / Off'u kontrol etmek için SW2'ye bas
  - De-bounce kullanmadan

Function	NuMaker-M031SD	NuMaker-M480-ETM
SW1	PB.4	PB.9
SW2	PB.0	PB.0
LED_R	PC.4	PC.9
LED_G	PC.5	PC.10
LED_B	PC.3	PC.11

Copyright © 2021 Nuvoton Technology Corporation

Example - LED_Init()

Use mask to set multi-function pin to avoid affecting same group other pins

```
void LED_Init(void)
{
    /* Set PC.3 ~ PC.5 to GPIO */
    SYS->GPC_MFPL = (SYS->GPC_MFPL & ~(SYS_GPC_MFPL_PC3MFP_Msk |
                     SYS_GPC_MFPL_PC4MFP_Msk | SYS_GPC_MFPL_PC5MFP_Msk)) |
                    (SYS_GPC_MFPL_PC3MFP_GPIO | SYS_GPC_MFPL_PC4MFP_GPIO |
                     SYS_GPC_MFPL_PC5MFP_GPIO);

    /* Set PC.3 ~ PC.5 to GPIO output */
    GPIO_SetMode(PC, (BIT3 | BIT4 | BIT5), GPIO_MODE_OUTPUT);

    /* Let LED off after initialize */
    LED_R = LED_OFF;
    LED_G = LED_OFF;
    LED_B = LED_OFF;
}
```


Copyright © 2021 Nuvoton Technology Corporation

Example - BTN_Init()

```
/**************  SW1 ***************/
/* Set PB.4 to GPIO */
SYS->GPB_MFPL = (SYS->GPB_MFPL & ~(SYS_GPB_MFPL_PB4MFP_Msk)) |
                (SYS_GPB_MFPL_PB4MFP_GPIO);
/* Set PB.4 to GPIO intput */
GPIO_SetMode(PB, BIT4, GPIO_MODE_INPUT);
GPIO_EnableInt(PB, 4, GPIO_INT_FALLING);
NVIC_EnableIRQ(GPIO_PAPB_IRQn);

/**************  SW2 ***************/
/* Set PB.0 to GPIO */
SYS->GPB_MFPL = (SYS->GPB_MFPL & ~(SYS_GPB_MFPL_PB0MFP_Msk)) |
                (SYS_GPB_MFPL_PB0MFP_GPIO);
/* Set PB.0 to GPIO intput */
GPIO_SetMode(PB, BIT0, GPIO_MODE_INPUT);
GPIO_EnableInt(PB, 0, GPIO_INT_FALLING);

/* Set de-bounce function */
GPIO_SET_DEBOUNCE_TIME(GPIO_DBCTL_DBCLKSRC_LIRC, GPIO_DBCTL_DBCLKSEL_512);
GPIO_ENABLE_DEBOUNCE(PB, BIT4);
```

The de-bounce clock source now is set to “LIRC”, remember to enable LIRC first.

Example - main()

```
/* Init LED */
LED_Init();

/* Init BTN */
BTN_Init();

while(1) {
    /* Check if the SW1 is pressed */
    if (sw1_int_cnt != sw1_cnt) {
        sw1_cnt = sw1_int_cnt;
        printf("SW1 interrupt count: %d\n", sw1_cnt);
    }
    /* Check if the SW2 is pressed */
    if (sw2_int_cnt != sw2_cnt) {
        sw2_cnt = sw2_int_cnt;
        printf("SW2 interrupt count: %d\n", sw2_cnt);
    }
}
```


Copyright © 2021 Nuvoton Technology Corporation

Example - ISR

```
void GPAB_IRQHandler(void)
{
    /* Check if PB.4 the interrupt occurred */
    if(GPIO_GET_INT_FLAG(PB, BIT4)) {
        LED_R ^= 1;
        sw1_int_cnt++;
        /* Clear PB.4 interrupt flag */
        GPIO_CLR_INT_FLAG(PB, BIT4);
      /* Check if PB.0 the interrupt occurred */
    } else if(GPIO_GET_INT_FLAG(PB, BIT0)) {
        LED_G ^= 1;
        sw2_int_cnt++;
        /* Clear PB.0 interrupt flag */
        GPIO_CLR_INT_FLAG(PB, BIT0);
    } else {
        /* Un-expected interrupt. Just clear all PB interrupts */
        PB->INTSRC = PB->INTSRC;
        printf("Un-expected interrupts.\n");
    }
}
```

Example – Exercise

- LED_B On / Off'ı kontrol etmek için SW1'e bas

Copyright © 2021 Nuvoton Technology Corporation

Ek olarak printf çıkış portunu değiştir

- Initialize UART port you used
- retarget.c
  - Change the definition of DEBUG_PORT to UART port you used

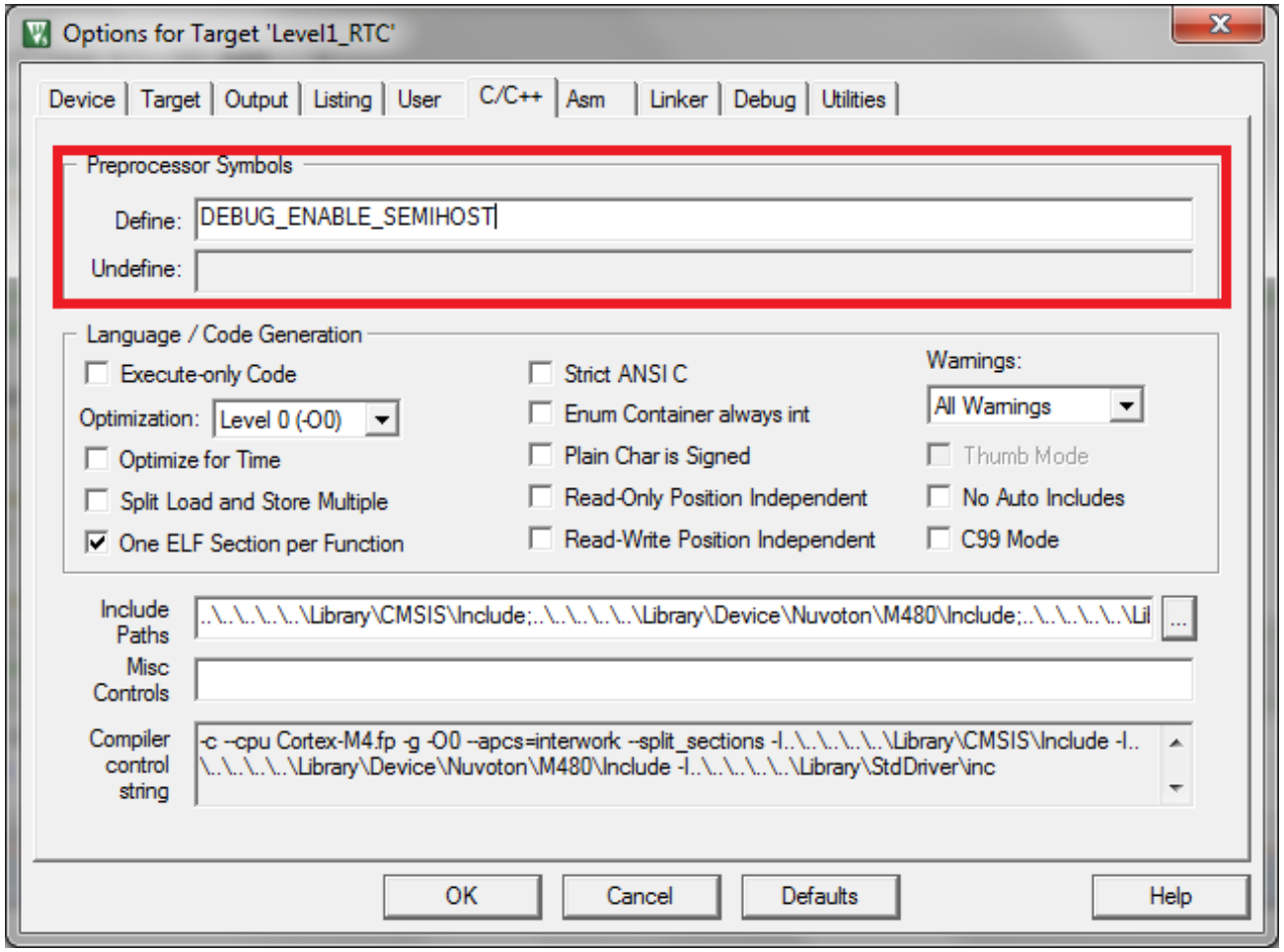
```
27 #if defined(DEBUG_ENABLE_SEMIHOST)
28     #ifndef DISABLE_UART
29         #define DISABLE_UART
30     #endif
31 #endif
32
33 #define DEBUG_PORT    UART0
34 /*---------------------------------------------------------------------------
35 /* Global variables
36 /*---------------------------------------------------------------------------
37 #if !(defined(__ICCARM__) && (__VER__ >= 6010000))
38 struct   FILE
```

- Rebulid
- Run

Copyright © 2021 Nuvoton Technology Corporation

Ek olarak – Semihost (1/2) etkinleştir

- Options -> C/C++
- Rebuild
- Enter debug mode

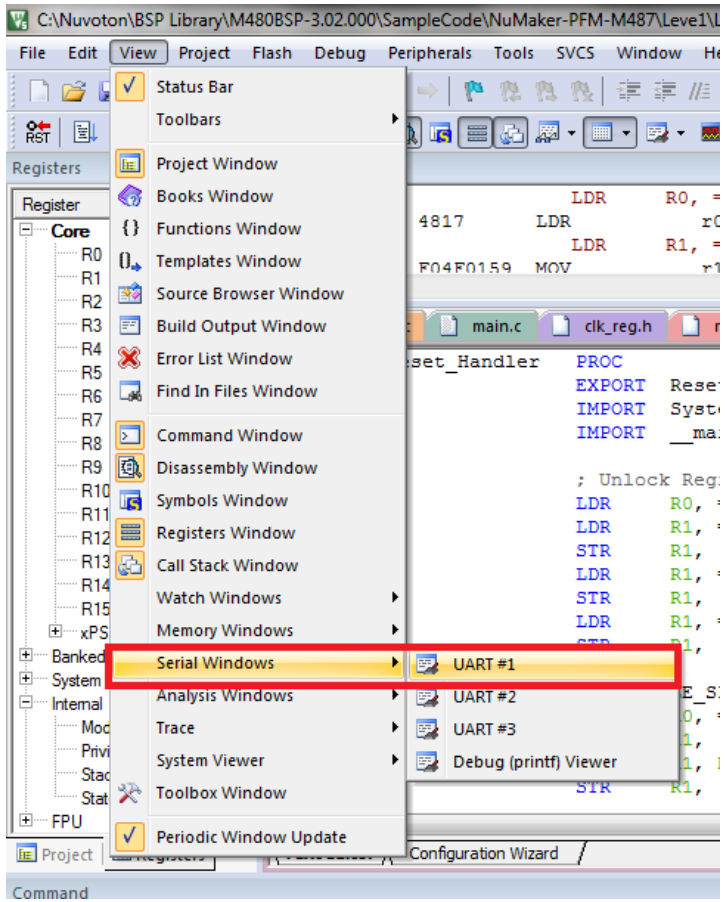


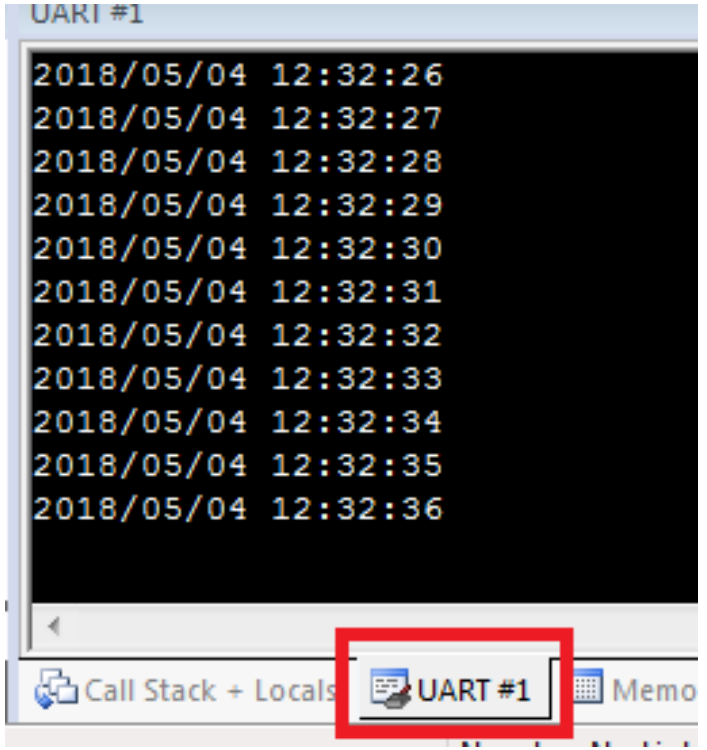
Note: It consumes CPU resources

Copyright © 2021 Nuvoton Technology Corporation

Additionally – Enable Semihost (2/2)

- View -> Serial Windows -> UART #1
- Run





Copyright © 2021 Nuvoton Technology Corporation

Port F1 ve Port F0 In-Circuit Emulator Pinleri

```
void gpio_pin_config_init_ice(void)
{
    SYS->GPF_MFPL &= ~(SYS_GPF_MFPL_PF1MFP_Msk | SYS_GPF_MFPL_PF0MFP_Msk);
    SYS->GPF_MFPL |= (SYS_GPF_MFPL_PF1MFP_ICE_CLK | SYS_GPF_MFPL_PF0MFP_ICE_DAT);
// GPIOF Low Byte Multiple Function Control Register
    return;
}
```


Copyright © 2021 Nuvoton Technology Corporation

Port F1 ve Port F0 In-Circuit Emulator Pinleri

Nuvoton mikrodenetleyicinizdeki **Port F** üzerindeki **PF1** ve **PF0** pinlerinin alternatif fonksiyonlarını değiştirerek, bu pinleri **In-Circuit Emulator (ICE)** yani devre içinde programlama ve hata ayıklama aracıyla iletişim kurmak için yapılandırmaktadır.

•**SYS->GPF_MFPL:** Bu, Port F'in alt kısmındaki çok fonksiyonlu pin (MFP) register'ına işaret eder. Bu register, port üzerindeki pinlerin alternatif fonksiyonlarını kontrol etmek için kullanılır.

•**SYS_GPF_MFPL_PF1MFP_Msk** ve **SYS_GPF_MFPL_PF0MFP_Msk:** Bu maskeler, sırasıyla PF1 ve PF0 pinlerinin alternatif fonksiyonlarını belirleyen bitleri izole etmek için kullanılır.

•**&= ~(...):** Bu satır, PF1 ve PF0 pinlerinin önceki alternatif fonksiyonlarını sıfırlar. Yani, bu pinlerin herhangi bir önceki ataması varsa, bu atama iptal edilir.

•**|= (...):** Bu satır ise PF1 pinini ICE saat pini olarak ve PF0 pinini ICE veri pini olarak yapılandırır.

Bu kod parçası, Nuvoton mikrodenetleyicinizin belirli iki pinini, bir programlama aracıyla iletişim kurabilmesi için özel bir moda geçirir.

Copyright © 2021 Nuvoton Technology Corporation

MFP (Çok Fonksiyonlu Pin)

Nuvoton mikrodenetleyicilerindeki birçok pin, sadece bir giriş veya çıkış olarak değil, aynı zamanda farklı özellikler için de kullanılabilir.

Örneğin, bir pin aynı anda bir GPIO pini, bir UART pini veya bir SPI pini olabilir. MFP register'ları, bu pinlerin hangi fonksiyonu kullanacağını belirlemek için kullanılır.

GPIO (Genel Amaçlı Giriş/Çıkış): Mikrodenetleyici üzerindeki pinlerin, yazılım tarafından kontrol edilerek giriş veya çıkış olarak kullanılabildiği en basit yapıdır.

Copyright © 2021 Nuvoton Technology Corporation

Joy of innovation

NUVOTON

Thank You

Danke

Merci

ありがとう

Gracias

Kiitos

감사합니다

धन्यबाद

شكرا لك

הדות