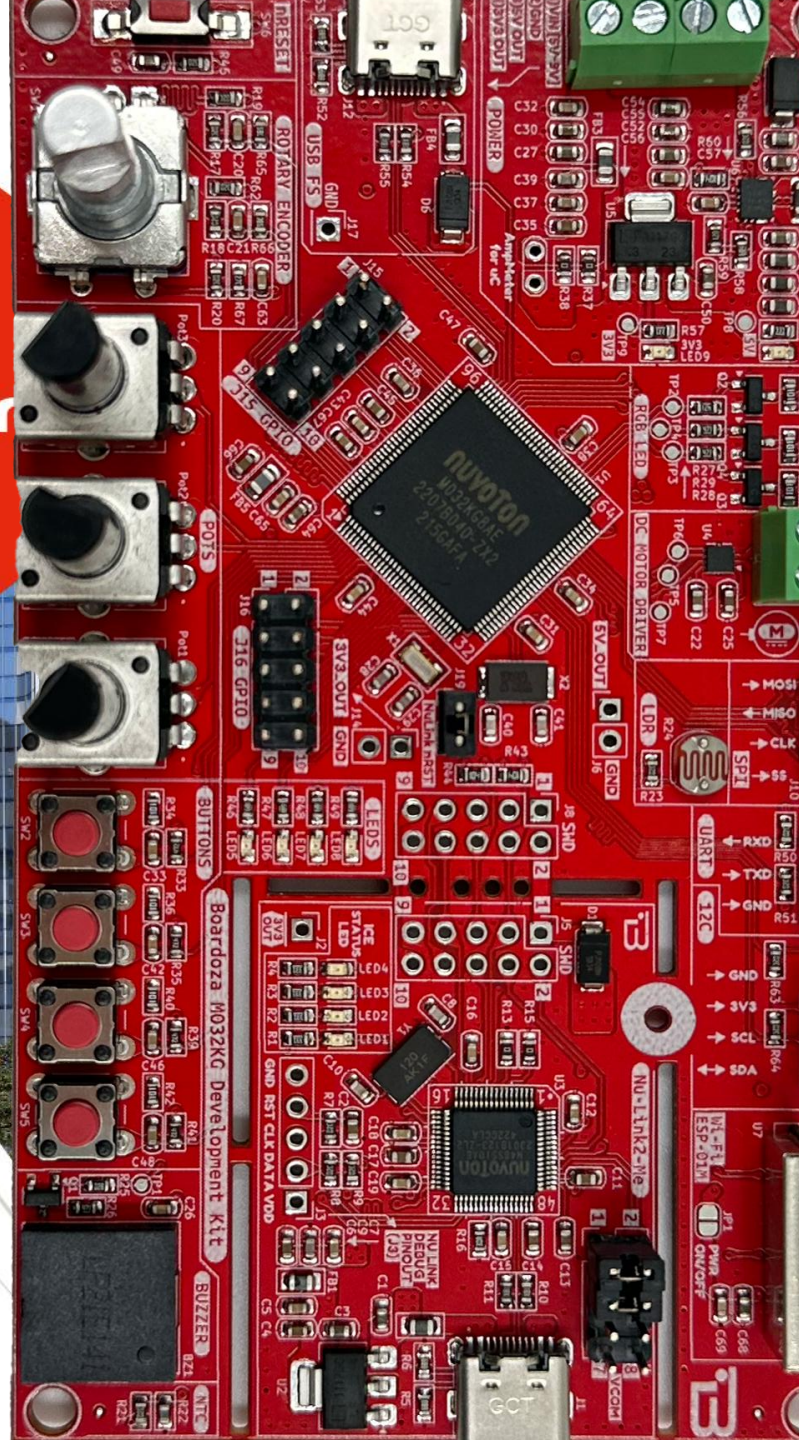
ozdisan

ARM KEIL
Microcontroller Tools

Joy of innovation
nuvoTon



NuMicro® UART

M031/M480 Level 1 Training

UART HABERLEŞME

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

2021

Not: Bu dokümanın telif hakkı tamamen Nuvoton'a aittir. Bu eğitim dokümanı mikrodenetleiyici eğitimi kapsamında Türkçe 'ye orijinalinden çevrilmiştir. Dokümanın orijinali İngilizcedir ve oluşan çeviri hataları tamamen Dr. Barış GÖKÇE'ye aittir.

İçerik

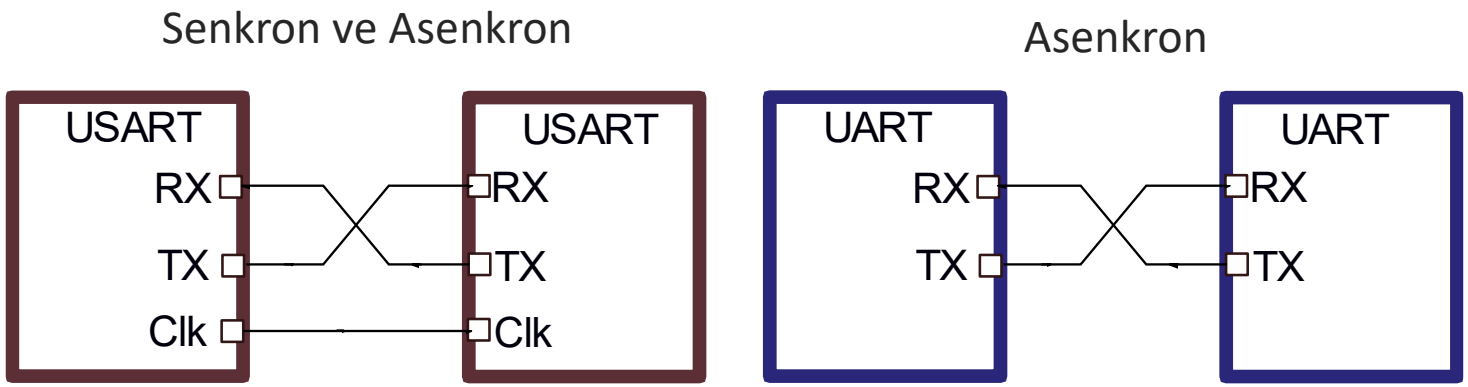
- UART nedir?
- Artıları ve Eksileri
- Özellikleri
- Seri İletişim Çerçevesi (Serial Communication Frame)
- Baud-Rate
- Clock Kaynağı
- Blok Diyagram
- Kesme Kullanımı (Interrupt)

UART nedir?

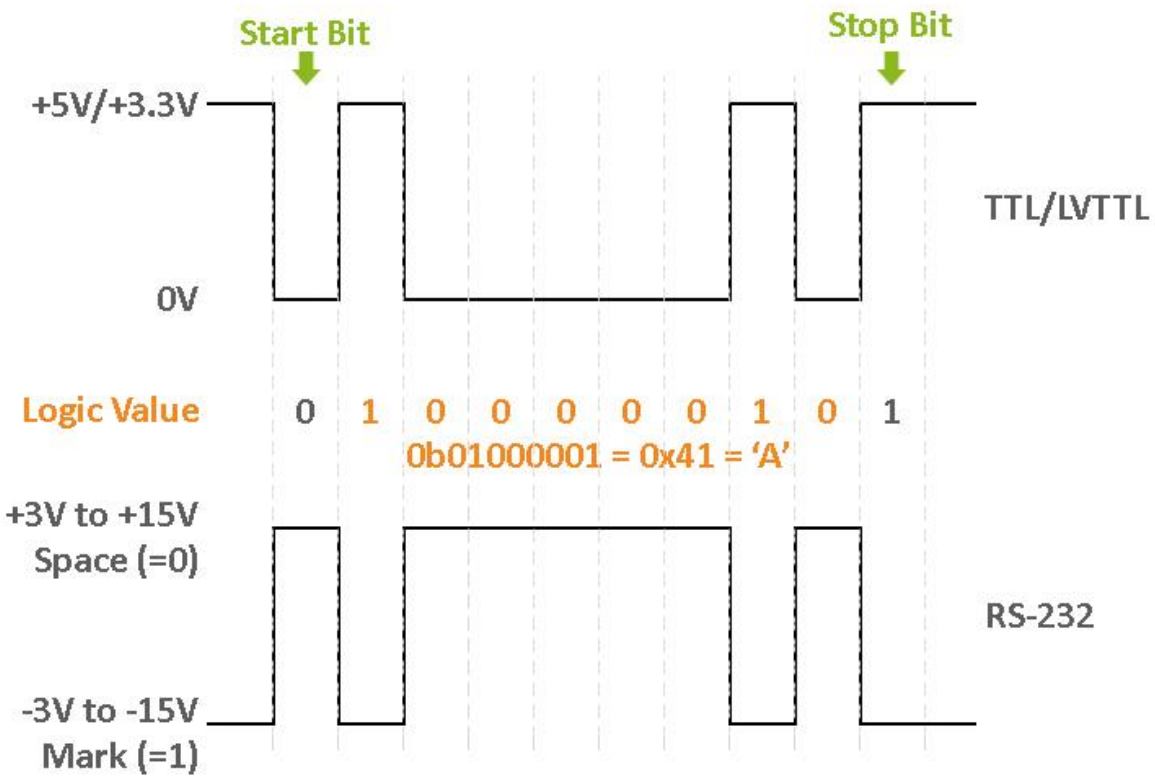
- UART
  - Universal Asynchronous Receiver/ Transmitter (Evrensel Asenkron Alıcı/Verici)
  - İki Kablolu (Two-Wired) (Tx/ Rx) Arayüz
  - Seri iletim
  - Asenkron
  - Full-Duplex
- Amacı
  - Veri(paket) Alımı/İletimi ve Kontrolü

USART ve UART

- USART: Evrensel senkron ve asenkron Alıcı-verici (Universal synchronous and asynchronous Receiver-transmitter)
  - Hem eşzamanlı hem de eşzamansız iletişimi destekler
- UART: Evrensel asenkron Alıcı-verici(Universal asynchronous Receiver-transmitter)



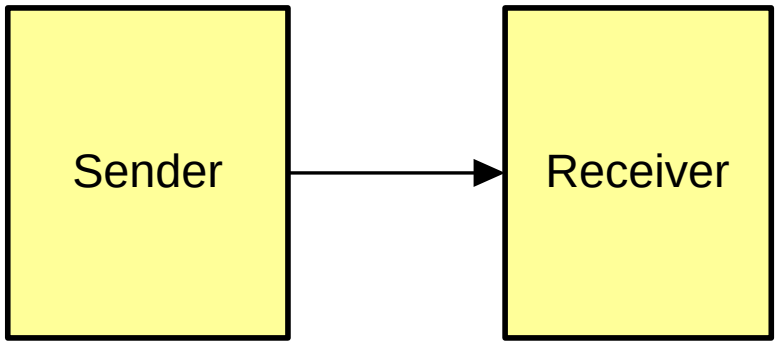
Voltaj Seviyeleri



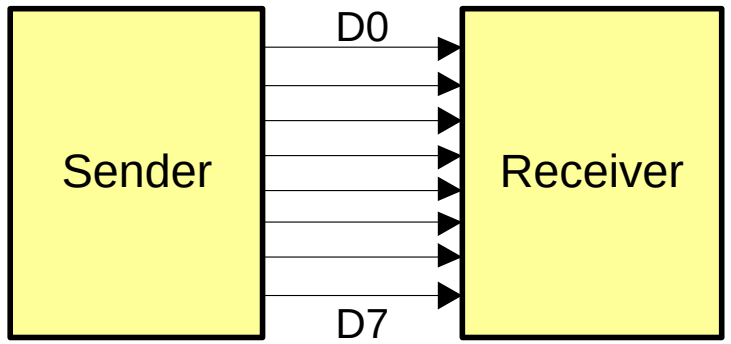
UART protokolünün mantığı 1 ve 0, birkaç voltaj gösterimi kullanılarak temsil edilir.
Örneğin mantık 1 için 5v ve mantık 0 için 0v anlamına gelen TTL. Ayrıca mantık 1'in 3.3v ve mantık 0'ın 0v olduğu LVTTL de vardır.
Yaygın olarak kullanılan bir başka voltaj seviyesi RS232'dir. Bu mantık düzeyinde, 0 + 3v ve + 15v arasında pozitif voltaj olarak temsil edilir ve 1, -3v ve -15v arasında negatif voltaj olarak temsil edilir.

Seri ve Parelel İletim

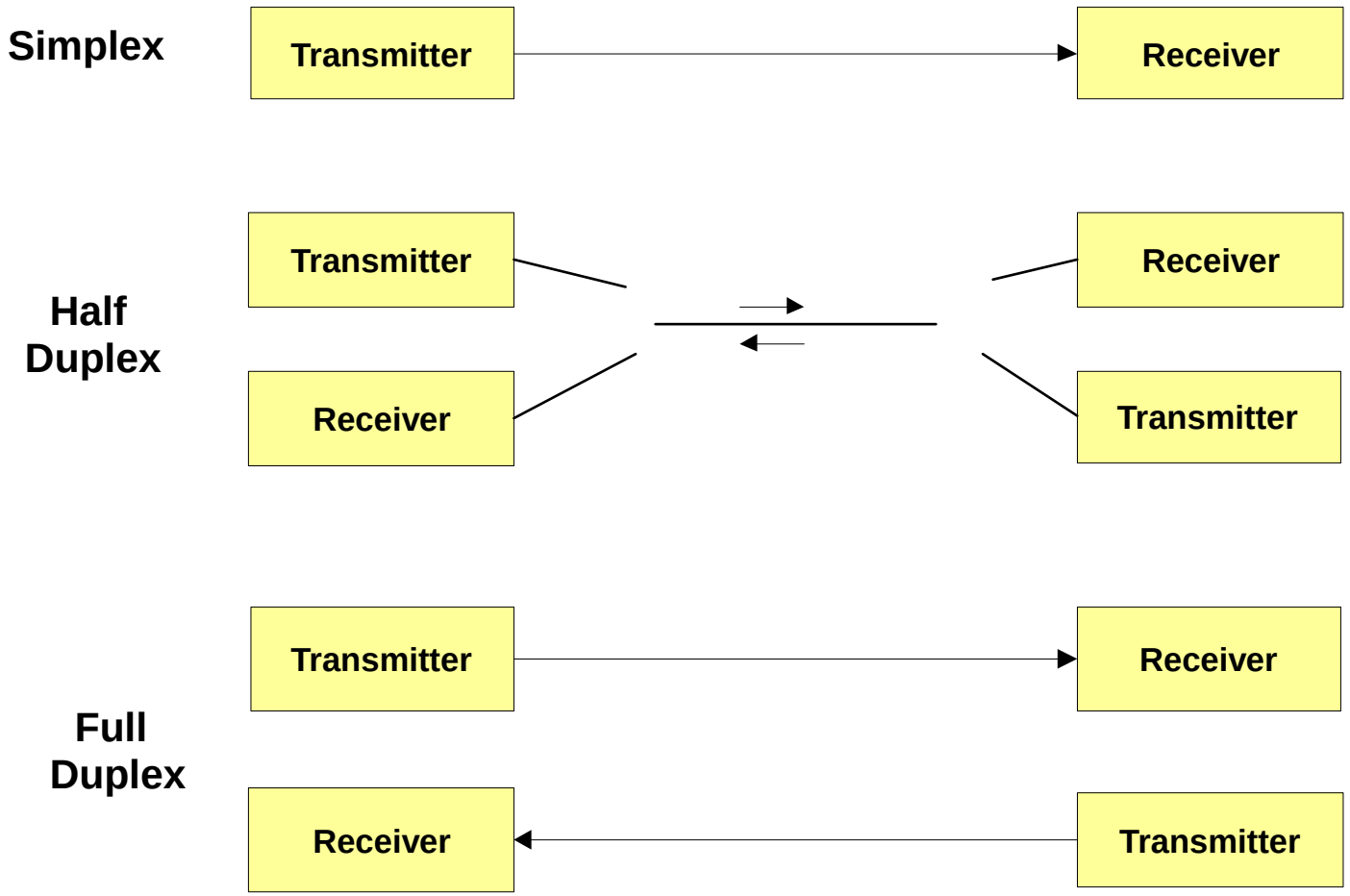
Serial Transfer



Parallel Transfer

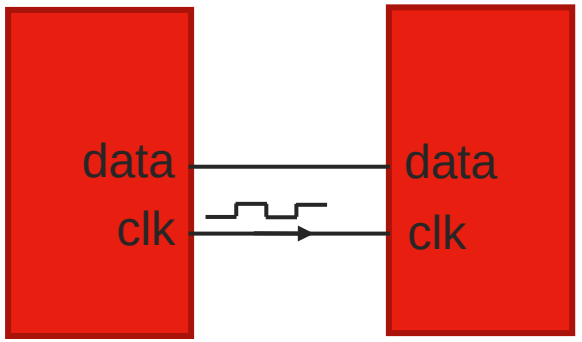


Veri İletim Yönü

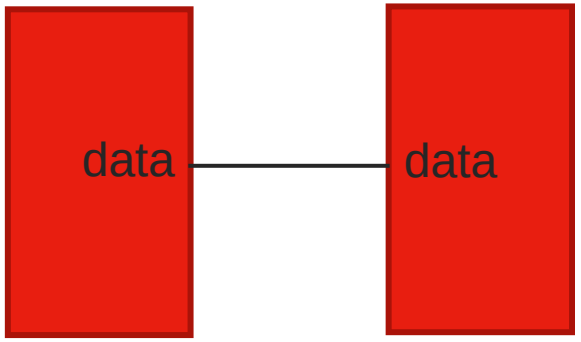


Senkron ve Asenkron İletim

- Senkron



- Asenkron
  - Saat Darbesi Yok (No clock)



Artıları ve Eksileri

- Artıları
  - I^2C ve SPI'dan daha uzak mesafe veri iletimi/alımı yapabilir
  - Two-wired arayüz (Tx/ Rx)
  - Full-duplex ileşim mümkündür
  - Kullanımı çok kolaydır
- Eksileri
  - Asenkron iletim veri(paket) kaybına neden olabilir.
  - Veri iletim hızı I^2C ve SPI'dan daha düşük.

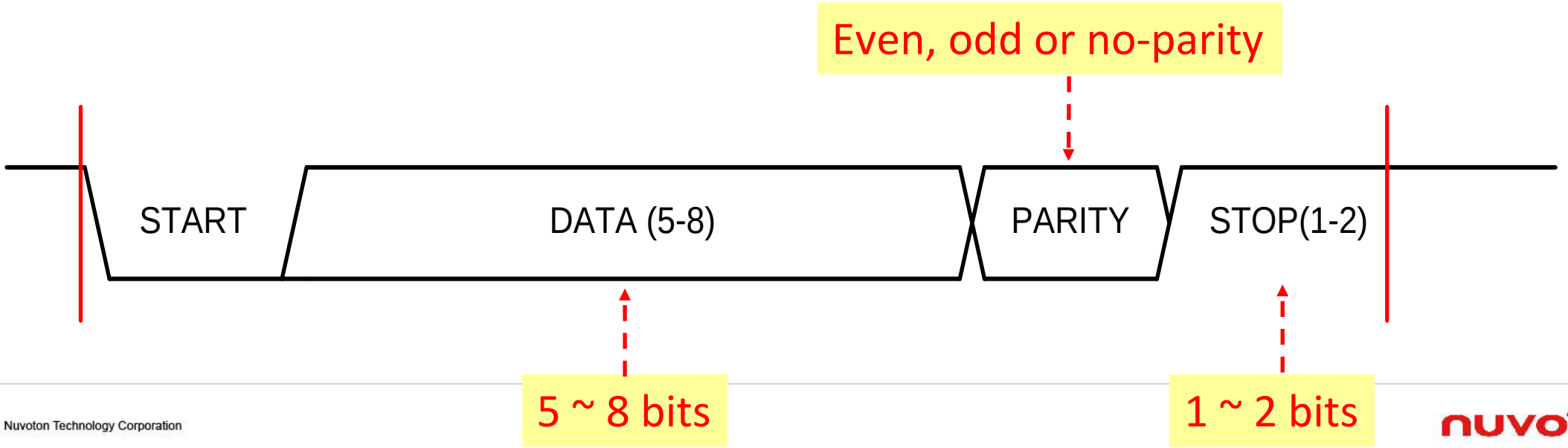
Özellikler

- 3 Adet’e kadar tamamen programlanabilir UART kanalı vardır
  - Programlanabilir baud-rate üreteci.
  - Programlanabilir hızlı arayüz.
  - Tx/Rx verileri için firiş FIFO su.
- Auto-baud rate algılama özelliği
- Otomatik akış kontrol fonksiyonu (nCTS, nRTS)
- nCTS ve Rx verileriyle uyandırma (wake-up) fonksiyonu
- IrDA SIR fonksiyon modu
- RS-485 fonksiyon modu
- Tek kablolu (Single-wire) fonksiyon modu

* M480 için 8 kanal

Seri iletişim çerçevesi(Serial Communication Frame)

- Çerçeve
  - Start bit:İletimin başlangıcını bildirmek için
  - Data bits: 5, 6, 7, veya 8 bit veri biti
  - Parity (Eşlik) bit: İletim hatası olup olmadığının kontrolü için
  - Stop bits: 1, 1.5, veya 2 bit durdurma biti

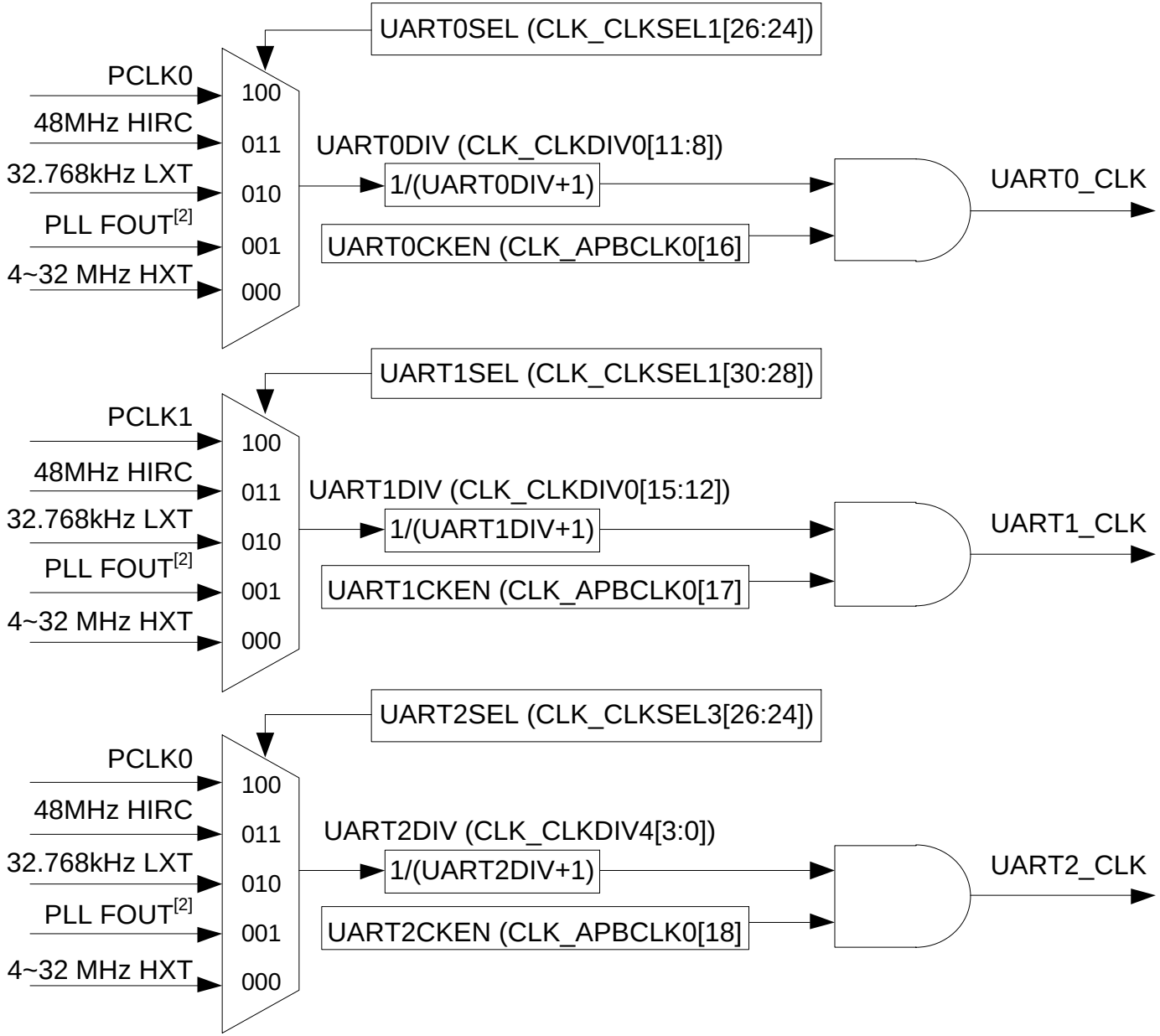


Baud-Rate

- Standart clock frekansları 1.8432 MHz nin katlarıdır
  - 3.6864 MHz, 7.3728 MHz, 14.7456 MHz, 18.432 MHz, 22.1184 MHz…
- Standart baud-rates 9600 bps'nin katlarıdır.
  - 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps…

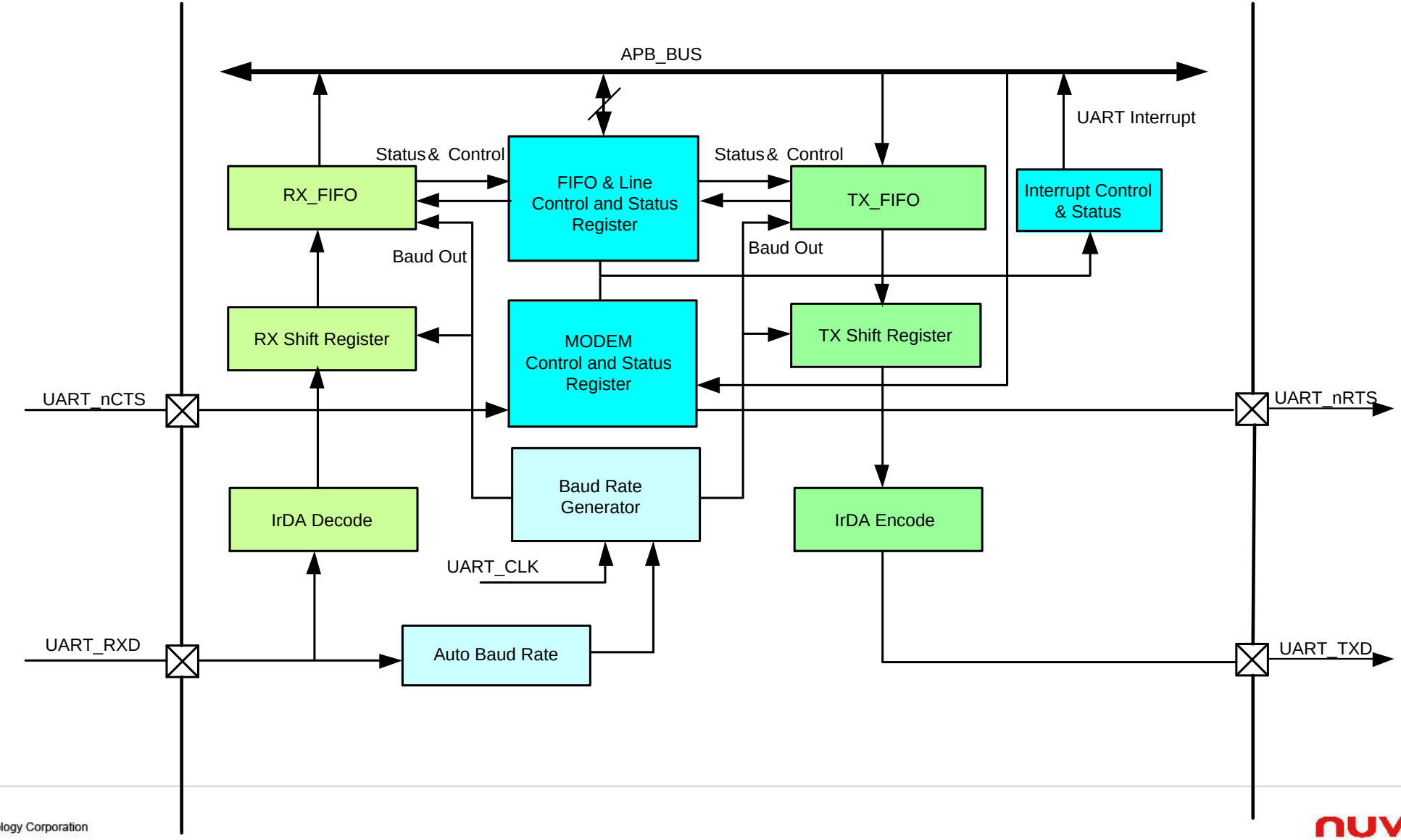
Clock Kaynağı

- Kullanıcı herhangi bir clock kaynağı üzerinden standart baud-rate' i kolayca üretebilir.



Note1: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.
Note2: If without PLL function, the clock source will be fixed at PCLK0/1, the same as UARTxSEL=3'b100 . Please refer to section 4.3 NuMicro® M031/M032 Series Selection Guide for detailed information.

Blok Diyagram



Örnek Kod – UART_Open

◆ UART_Open()

```
void UART_Open ( UART_T * uart,
                 uint32_t u32baudrate
               )
```

Open and set UART function.

Parameters

[in] **uart** The pointer of the specified UART module.

[in] **u32baudrate** The baudrate of UART module.

Returns

None

This function use to enable UART function and set baud-rate.

```
/* Configure UART0 and set UART0 baud rate
   */

UART_Open(UART0, 115200);
```

Kesme (Interrupt - Common Use)

Kesme Adı	Kesme Aktifleme Bit	Kesme Tetikleme Zamanlama
Transmitter Empty	**TXENDIEN**	TX send frame finished
Wake-up	**WKIEN**	MCU wake-up
Buffer Error	**BUFERRIEN**	Tx/Rx FIFO overflow
RX Time-out	**RXTOIEN**	The time that data place on RX FIFO is over the setting time
Transmit Holding Register Empty	**THREIEN**	TX FIFO is empty
Receive Data Available	**RDAIEN**	Received data in RX FIFO equals to trigger level (RFITL). It can be 1, 4, 8, 14 bytes.

API – Enable UART Interrupt

◆ UART_EnableInt()

```
void UART_EnableInt ( UART_T *  uart,
                      uint32_t  u32InterruptFlag
                    )
```

The function is used to enable UART specified interrupt and enable NVIC UART IRQ.

Parameters

`[in]` **uart** The pointer of the specified UART module.

`[in]` **u32InterruptFlag** The specified interrupt of UART module:

- **UART_INTSTS_SWBEINT_Msk** : Single-wire Bit Error Detect Interrupt
- **UART_INTEN_WKIEN_Msk** : Wake-up interrupt
- **UART_INTEN_BUFERRIEN_Msk** : Buffer Error interrupt
- **UART_INTEN_RXTOIEN_Msk** : Rx time-out interrupt
- **UART_INTEN_MODEMIEN_Msk** : Modem status interrupt
- **UART_INTEN_RLSIEN_Msk** : Receive Line status interrupt
- **UART_INTEN_THREIEN_Msk** : Tx empty interrupt
- **UART_INTEN_RDAIEN_Msk** : Rx ready interrupt *

◆ UART_SetTimeoutCnt()

```
void UART_SetTimeoutCnt ( UART_T *  uart,
                          uint32_t  u32TOC
                        )
```

Set Rx timeout count.

Parameters

`[in]` **uart** The pointer of the specified UART module.

`[in]` **u32TOC** Rx timeout counter.

Returns

None

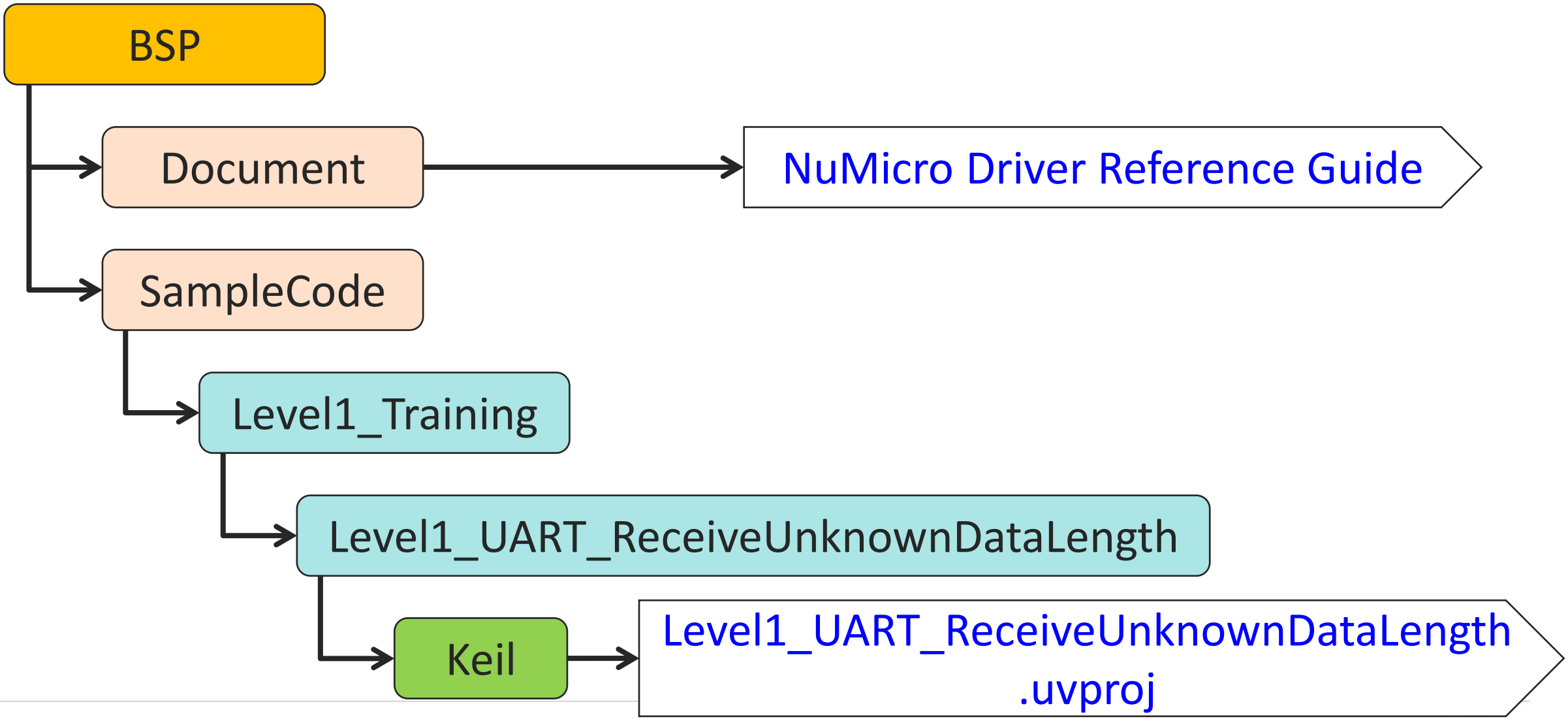
This function use to set Rx timeout count.

```
/* Enable UART RDA and RX timeout interrupt */
UART_EnableInt(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk);
/* Set RX Trigger Level as 4 bytes */
UART0->FIFO = ((UART0->FIFO & (~UART_FIFO_RFITL_Msk))|UART_FIFO_RFITL_4BYTES);
/* Set Timeout time counter in 60 bit-time and enable time-out counter */
UART_SetTimeoutCnt(UART0, RX_TIMEOUT_CNT);
```

ÖRNEK KOD

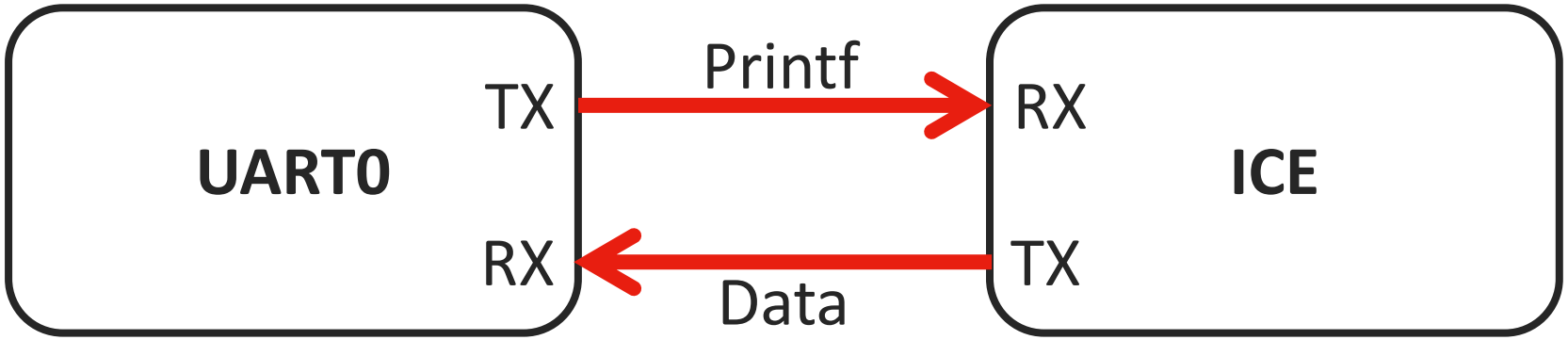
Bilinmeyen Veri Uzunluğunu Alma

Örnek – Proje Yolu



Bilinmeyen Veri Uzunluğu Paketi Alındı

- UART0(TX) receives data from ICE UART (PC console)
- Data Length : Unknown
- Baud-rate and Frame: 115200 - 8n1 (8bit, no parity, 1bit stop)



Bilinmeyen Veri Uzunluğu Paketi Alındı

- Prensip
  - PC konsolu ICE'ye veri gönderir ve ICE UART, çip UART RX'e veri gönderir
  - RX, RX FIFO'dan RDA ve RXTO kesintisi ile veri alır
- Örnek : (40 byte ilet)

Process	1	2	3	4	5	6	7	8	……	38	39	40	41	42
RX FIFO	4	RDA INT	1	4	RDA INT	1	4	RDA INT	……	RDA INT	1	RXTO INT	0	Disable RDA, RXTO INT
RX Buffer	0		3	3		6	6		……		39		40	

UART Example – Initialize UART

- Step1: Enable UART TX/RX module clock and Select clock source
  - CLK_EnableModuleClock(UART0_MODULE) ;
  - CLK_SetModuleClock(UART0_MODULE, CLK_CLKSEL1_UART0SEL_HIRC, CLK_CLKDIV0_UART0(1));
- Step2: Set multi-function pins
- Step3: Initialize UART module
  - SYS_ResetModule(UART0_RST); /* Reset UART0 */
  - UART_Open(UART0, 115200); /* Configure UART1 and set UART1 baud rate */
  - UART_EnableInt(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk); /* Enable Interrupt */
  - UART0->FIFO = ((UART0->FIFO & (~UART_FIFO_RFITL_Msk)) | UART_FIFO_RFITL_4BYTES);
    /* Set RX Trigger Level as 4 bytes */
  - UART_SetTimeoutCnt(UART0, RX_TIMEOUT_CNT);
    /* Set Timeout time counter in 60 bit-time and enable time-out counter */
- Step4: Start to receive data

UART Example – Main Function

```
SYS_Init();    /* Init System, peripheral clock and multi-function I/O */
UART0_Init();  /* Init UART */
while(1)
{
    while(UART_RX_IDEL(UART0)); /* Wait to receive UART data */
    /* Start to received UART data */
    g_bUART_RX_Received_Data_State = eUART_RX_Received_Data_NOT_Finish;
    /* Wait for receiving UART message finished */
    while(g_bUART_RX_Received_Data_State != eUART_RX_Received_Data_Finish);
    printf("\nUART0 Rx Received Data : %s\n",g_au8UART_RX_Buffer);
    printf("UART0 Rx RDA (Fifofull) interrupt times : %d\n",g_u8UART_RDA_Trigger_Cnt);
    printf("UART0 Rx RXTO (Timeout) interrupt times : %d\n",g_u8UART_RXTO_Trigger_Cnt);

    /* Reset UART interrupt parameter */
    UART_EnableInt(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk);
    g_u8UART_RDA_Trigger_Cnt = 0;  // UART RDA interrupt times
    g_u8UART_RXTO_Trigger_Cnt = 0; // UART RXTO interrupt times
}
```

UART Example – Init UART Function

```
void UART0_Init(void)
{
    /* Reset UART0 */
    SYS_ResetModule(UART0_RST);

    /* Configure UART0 and set UART0 baud rate */
    UART_Open(UART0, 115200);

    /* Enable UART RDA and RX timeout interrupt */
    UART_EnableInt(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk);

    /* Set RX Trigger Level as 4 bytes */
    UART0->FIFO = ((UART0->FIFO & (~UART_FIFO_RFITL_Msk)) | UART_FIFO_RFITL_4BYTES);

    /* Set Timeout time counter in 60 bit-time and enable time-out counter */
    UART_SetTimeoutCnt(UART0, RX_TIMEOUT_CNT);
}
```

UART Example – RX Interrupt Handler

```
void UART02_IRQHandler(void){
    if(UART_GET_INT_FLAG(UART0, UART_INTSTS_RDAINT_Msk)) {
        // Move the data from Rx FIFO to sw buffer (RAM). Every time leave 1 byte data in FIFO for Rx timeout
        for(i = 0 ; i < (FIFO_THRESHOLD - 1) ; i++)
            g_au8UART_RX_Buffer[u16UART_RX_Buffer_Index++] = UART_READ(UART0);
    }
    else if(UART_GET_INT_FLAG(UART0, UART_INTSTS_RXTOINT_Msk))
    {
        /* When Rx timeout flag is set to 1, it means there is no data needs to be transmitted. */
        /* Move last data from Rx FIFO to sw buffer. */
        while(UART_GET_RX_EMPTY(UART0) == 0)
            g_au8UART_RX_Buffer[u16UART_RX_Buffer_Index++] = UART_READ(UART0);
        /* Clear UART RX parameter */
        UART_DISABLE_INT(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk);
        u16UART_RX_Buffer_Index = 0;
        g_bUART_RX_Received_Data_State = eUART_RX_Received_Data_Finish;
    }
}
```

Joy of innovation

NUVOTON

Thank You

Danke

Merci

ありがとう

Gracias

Kiitos

감사합니다

धन्यबाद

شكرا لك

הדות