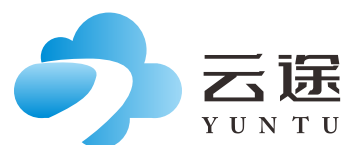


应用笔记

AN0019

FlexCAN 模块简介



目录

1	介绍	3
2	云途各系列 MCU FlexCAN 配置	3
2.1	YTM32B1LD 系列配置	3
2.2	YTM32B1LE 系列配置	3
2.3	YTM32B1ME 系列配置	3
2.4	YTM32B1MD 系列配置	3
3	消息缓存区 Message Buffer	4
3.1	经典 CAN 模式下消息缓冲区 Classical CAN Mode Message Buffer	4
3.1.1	FIFO 禁用	4
3.1.2	FIFO 使能	5
3.2	CANFD 模式下消息缓冲区 CANFD Mode Message Buffer	7
3.3	邮箱结构 Mail Box Structure	7
3.4	FIFO 数据结构 FIFO Structure	9
3.4.1	FIFO 邮箱信息段 (Header) 位定义	9
3.4.2	FIFO 滤波器接收码	10
3.5	接收滤波器 Filter	10
3.5.1	全局掩码模式 MCR[IRMQ] = 0	10
3.5.1.1	全局掩码模式下 FIFO 禁用时邮箱滤波器掩码和接受码	10
3.5.1.2	全局掩码模式下 FIFO 使能时 FIFO 滤波器掩码和接受码	10
3.5.2	独立掩码模式 MCR[IRMQ] = 1	11
3.5.2.1	独立掩码模式下 FIFO 禁用时邮箱滤波器掩码和接受码	11
3.5.2.2	独立掩码模式下 FIFO 使能时 FIFO 滤波器掩码和接受码	11
3.5.3	FIFO 滤波器格式 FIFO Filter Format	12
4	CAN 波特率 CAN Baud rate	12
4.1	FlexCAN 模块时钟	13
4.2	T_q 数-N	13
4.3	经典 CAN 模式波特率设定	13
4.4	CANFD 模式下波特率设定	14
4.5	CANFD 模式下 CAN_ETDC[ETDCEN] = 0 时 SSP 设定	15
4.5.1	SSP 计算方法	15
4.5.2	SSP 设定方法	16

历史版本

版本号	日期	修改
1.0	2022-10-28	初始版本

1 介绍

本文主要针对 FlexCAN 的工作原理及各种工作模式进行详细的讲解。

2 云途各系列 MCU FlexCAN 配置

2.1 YTM32B1LD 系列配置

特性	YTM32B1LD
CAN 通道数 CAN Channel	CAN0
缓冲区块数量 RAM Block Number	1
8 字节负载数量 8-Byte Payload Size	32
FIFO	Yes
增强型 FIFO Enhanced FIFO	No
PN Pretended Networking	No
内存错误监测及注入 Memory Error Detection and Correction	No

2.2 YTM32B1LE 系列配置

特性	YTM32B1LE
CAN 通道数 CAN Channel	CAN0
缓冲区块数量 RAM Block Number	1
8 字节负载数量 8-Byte Payload Size	32
FIFO	Yes
增强型 FIFO Enhanced FIFO	Yes
PN Pretended Networking	No
内存错误监测及注入 Memory Error Detection and Correction	No

2.3 YTM32B1ME 系列配置

特性	-	-	-	-	-	-
CAN 通道数 CAN Channel	CAN0	CAN1	CAN2	CAN3	CAN4	CAN5
缓冲区块数量 RAM Block Number	2	2	2	1	1	1
8 字节负载数量 8-Byte Payload Size	64	64	64	32	32	32
FIFO	Yes	Yes	Yes	Yes	Yes	Yes
增强型 FIFO Enhanced FIFO	Yes	Yes	Yes	No	No	No
PN 支持 Pretended Networking Support	Yes	No	No	No	No	No
内存错误监测及注入 Memory Error Detection and Correction	Yes	Yes	Yes	Yes	Yes	Yes

2.4 YTM32B1MD 系列配置

特性	-	-	-
CAN 通道数 CAN Channel	CAN0	CAN1	CAN2
缓冲区块数量 RAM Block Number	2	1	1
8 字节负载数量 8-Byte Payload Size	64	32	32
FIFO	Yes	Yes	Yes
增强型 FIFO Enhanced FIFO	Yes	No	No
PN 支持 Pretended Networking Support	No	No	No
内存错误监测及注入 Memory Error Detection and Correction	No	No	No

3 消息缓存区 Message Buffer

每通道 CAN 控制器的消息缓存区由 1~2 个缓冲区块（RAM Block/region）组成，每个缓冲区块占用 512 字节空间。每个缓冲区块可根据负载配置成若干个消息邮箱，邮箱从偏移地址 0x80 开始顺序排列。

消息缓冲区在不同的 CAN 工作模式下，具备不同的数据结构及特征。在初始化时，可通过 MCR[FDEN] 配置 CAN 工作模式。

- MCR[FDEN] = 0 时：CAN 工作在经典 CAN 模式下
- MCR[FDEN] = 1 时：CAN 工作在 CANFD 模式下

3.1 经典 CAN 模式下消息缓冲区 Classical CAN Mode Message Buffer

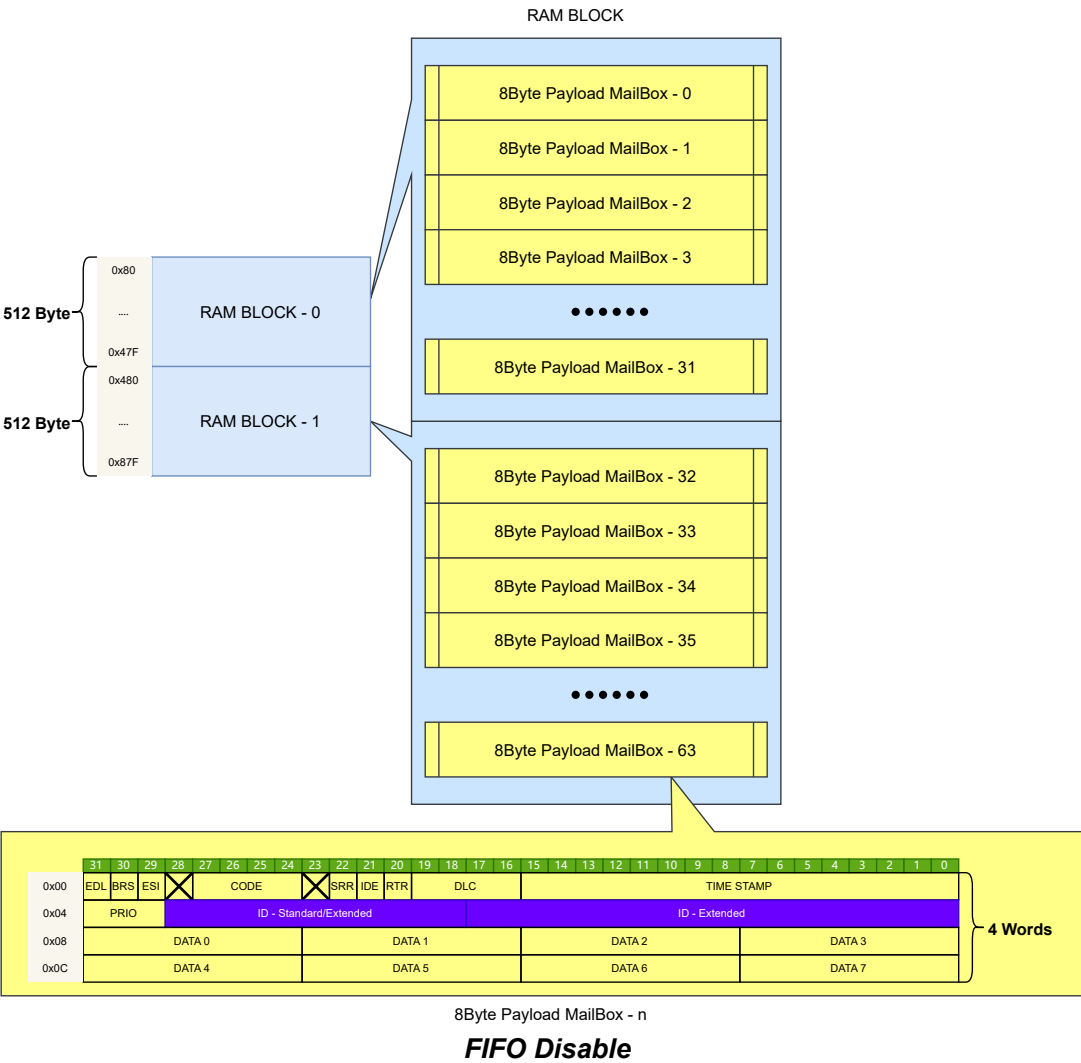
邮箱负载在经典 CAN（Classical CAN）模式下无法配置，默认为 8 字节负载。

在经典 CAN 模式下，消息缓冲区可被分配为仅邮箱模式（即 FIFO 禁用模式）、FIFO 使能模式。

- MCR[RFEN] = 0 时，消息缓冲区为仅邮箱模式（即 FIFO 禁用模式）
- MCR[RFEN] = 1 时，消息缓冲区为 FIFO 使能模式

3.1.1 FIFO 禁用

当 FIFO 禁用时，缓冲区块（RAM Block/region）被分配为 32 个 8 字节负载邮箱，每个邮箱占 4 个字（4 words）空间。包含 2 个缓冲区块的邮箱个数为 64。



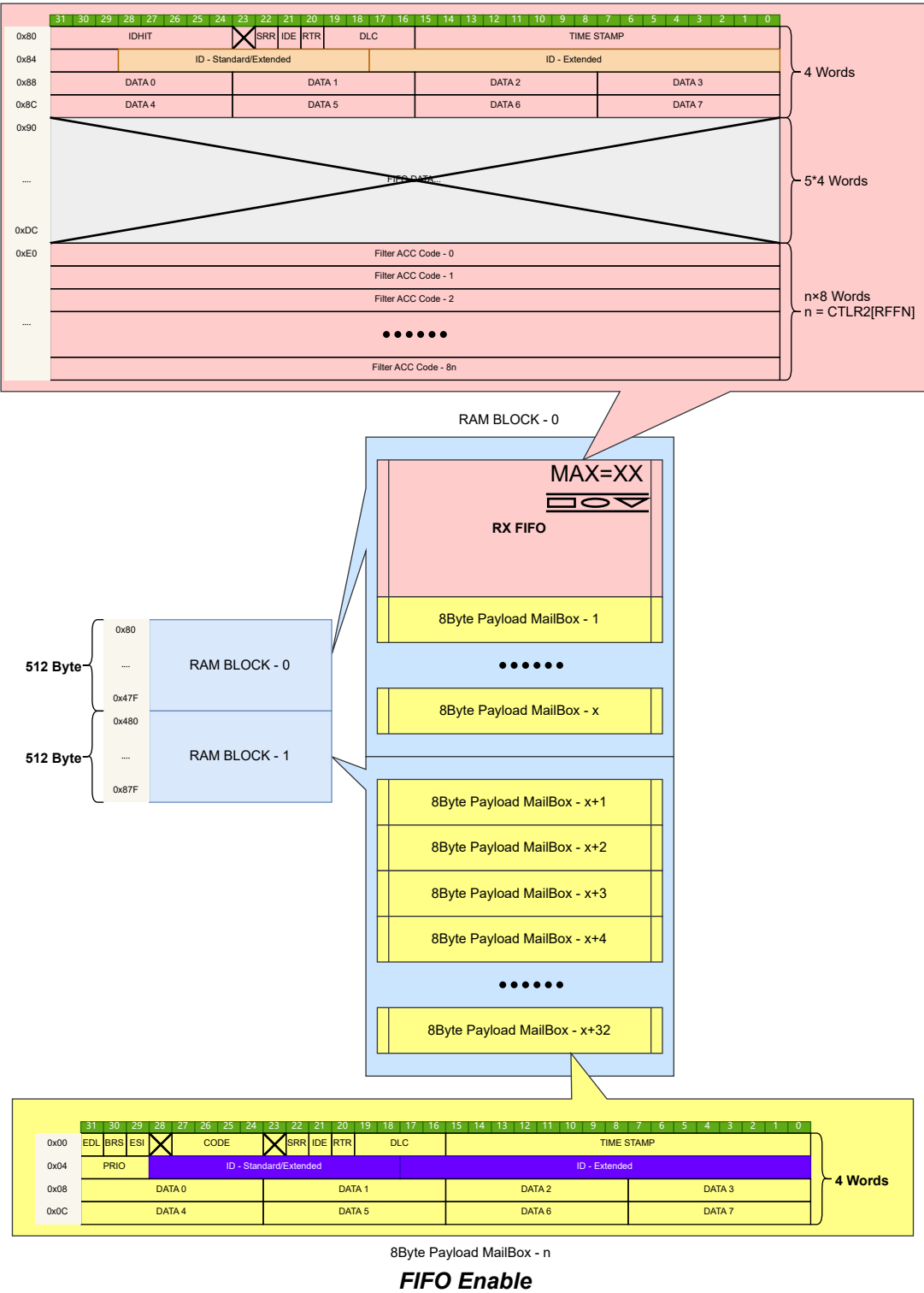
3.1.2 FIFO 使能

当 FIFO 使能时，前 6+n 个邮箱被分配为接收 FIFO，接受 FIFO 包含接受消息队列和接收滤波器接收码。消息队列固定占用 6 个消息邮箱。接收滤波器接收码数量通过寄存器位域 CTLR2[RFFN] 配置，故 FIFO 占用的消息邮箱的个数取决于配置的接收滤波器接收码的个数。

滤波器接收码数量：

$$N_{=8n} = 8 \times (CTLR2[RFFN] + 1)$$

其中：- $N_{FIFOFilterAcc}$: 滤波器数量 - n : 滤波器数量代码， $n = CTLR2[RFFN] + 1$
剩余的 RAM 空间被分配为消息邮箱。



CTRL2[RFFN]	滤波器接收码数量	首个消息邮箱起始偏移地址	剩余最大邮箱数量（最大 32 个邮箱）	剩余最大邮箱数量（最大 64 个邮箱）
0	8	0x100	24	56
1	16	0x120	22	54
2	24	0x140	20	52
3	32	0x160	18	50
4	40	0x180	16	48
5	48	0x1A0	14	46
6	56	0x1C0	12	44
7	64	0x1E0	10	42
8	72	0x200	8	40

CTRL2[RFFN]	滤波器接收码数量	首个消息邮箱起始偏移地址	剩余最大邮箱数量（最大 32 个邮箱）	剩余最大邮箱数量（最大 64 个邮箱）
9	80	0x220	6	38
10	88	0x240	4	36
11	96	0x260	2	34
12	104	0x280	0	32
13	112	0x2A0	0	30
14	120	0x2C0	0	28
15	128	0x2E0	0	26

3.2 CANFD 模式下消息缓冲区 CANFD Mode Message Buffer

当 CAN 控制器被配置为工作在 CANFD 模式下，每个缓冲区块（RAM Block/region）可被配置为不同的邮箱负载大小（Payload size）。通过 FDCTRL[MBDSRx] 分别定义邮箱负载大小。

20 - 19 MBDSR1	Message Buffer Data Size for Region 1 – Selects the data size. For region R1 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR1 can be written in Freeze mode only.
17 - 16 MBDSR0	Message Buffer Data Size for Region 0 – Selects the data size. For region R0 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR0 can be written in Freeze mode only.

当缓冲区块（RAM Block/region）被定义大小后，每个邮箱的数据结构及占用的 RAM 空间大小不同，那么每个缓冲区块（RAM Block/region）所包含的邮箱个数也不同。

FDCTRL[MBDSRx]	邮箱负载大小	缓冲区块最大邮箱数量
0b00	8 字节负载（8 byte payload size）	32
0b01	16 字节负载（16 byte payload size）	21
0b10	32 字节负载（32byte payload size）	12
0b11	64 字节负载（64 byte payload size）	7

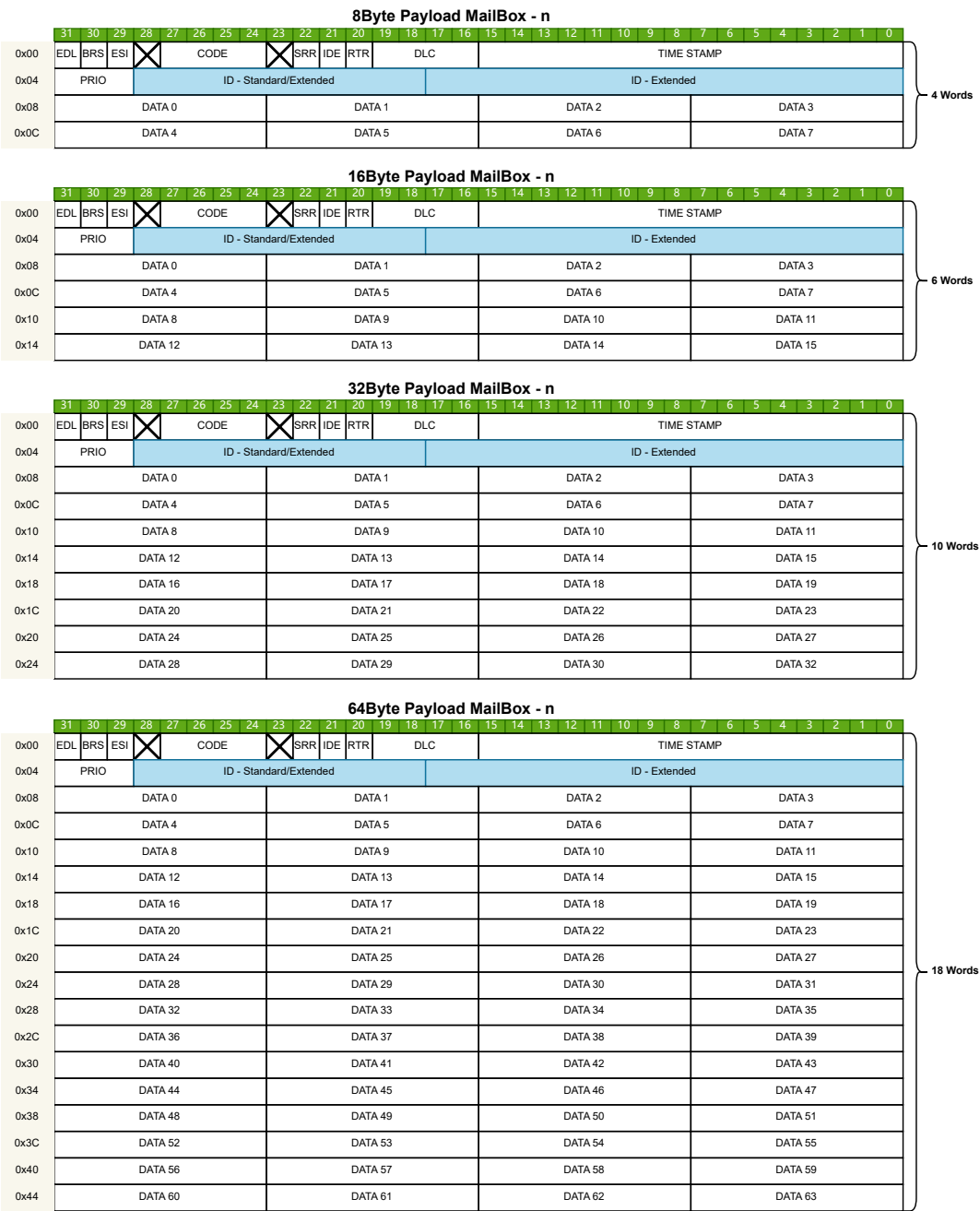
当开启 CANFD 模式时，FIFO 功能被禁用，MCR[FDEN] 必须设置为 0。可通过配置 ERFCDR [ERFEN] 开启增强型 FIFO（Enhanced Rx FIFO）。

3.3 邮箱结构 Mail Box Structure

不同邮箱负载大小的（Payload size）邮箱数据段长度不同。
邮箱数据结构主要包含 2 部分内容，信息段（Header）和数据段（Data）
信息段（Header）位定义

- EDL：区分经典 CAN 帧和 CANFD 帧
 - 0：经典 CAN 帧
 - 1：CANFD 帧
- BRS：指示是否为可变波特率帧

- 0：非可变波特率帧
 - 1：可变波特率帧
- ESI：错误状态指示，仅用于发送邮箱
 - 0：无错误
 - 1：发生错误
- CODE：邮箱控制或状态代码，详细参见参考手册 FlexCAN 章节 Message Buffer Structure 小结
- SRR：
 - 当被用作发送邮箱时，此位必须置 1，SRR = 1
- IDE：扩展帧标识
 - 0：标准帧
 - 1：扩展帧
- RTR：当用于发送邮箱时，指示为远程帧；当用于接收邮箱时，作为远程帧滤波器
 - 0：当用于发送邮箱时，此邮箱为非远程帧发送邮箱；当用于接收邮箱时，此邮箱仅接受非远程帧
 - 1：当用于发送邮箱时，此邮箱为远程帧发送邮箱；当用于接收邮箱时，此邮箱仅接受远程帧
- DLC：指示数据长度信息
- TIME STAMP：发送或接收时间戳
- PRIO：邮箱仲裁优先级
- ID：接收时，作为该邮箱接收滤波器接收码；发送时，作为发送报文 ID
- DATA 0~DATA 63：CAN 帧数据段



3.4 FIFO 数据结构 FIFO Structure

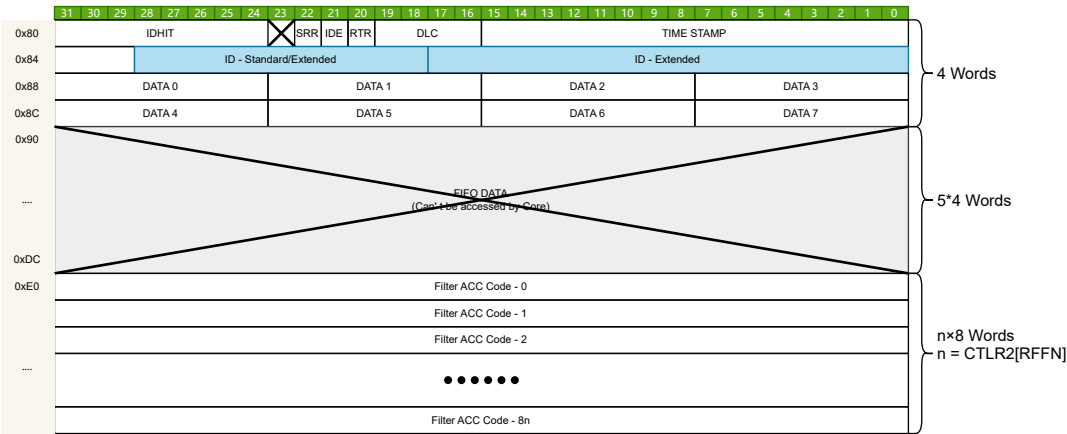
FIFO 由 6 个 FIFO 邮箱和 FIFO 滤波器接收码组组成，FIFO 邮箱包含信息段（Header）和数据段（Data），滤波器接收码组由 8n 个滤波器接收码组成。

3.4.1 FIFO 邮箱信息段（Header）位定义

- IDHIT：接收消息所使用的滤波器索引
- SRR：FD 帧指示位
- IDE：扩展帧指示位
- RTR：远程帧指示位
- DLC：指示数据长度信息

- TIME STAMP：接收时间戳
- ID：接收消息 ID
- DATA0~DATA8：CAN 帧数据段

3.4.2 FIFO 滤波器接收码



3.5 接收滤波器 Filter

接收滤波器由滤波器掩码（Mask Code）和接受码（Acceptance Code）组成，当 FlexCAN 在不同的配置情况下接收滤波器的掩码和接受码组合是不同的。通过 MCR[IRMQ] 可配置接收滤波器的掩码为全局掩码或独立掩码。

- MCR[IRMQ] = 0 时：全局掩码模式。
 - 仅 MailBox-14、MailBox-15 使用掩码寄存器 RX14MASK、RX15MASK 作为掩码；当 FIFO 禁用时，其他所有邮箱的掩码均使用掩码寄存器 RXMGMASK。当 FIFO 使能时，FIFO 的
- MCR[IRMQ] = 1 时：独立掩码模式。
 - 当 FIFO 禁用时，
 - 当 FIFO 使能时，FIFO 采用 RXIMR(0)~ RXIMR(n) 及 RXMGMASK 作为其滤波器组（Filter Table）掩码。MailBox-1 ~ MailBox-(x+32) 使用 RXIMR-n+1 ~ RXIMR-(x+32) 作为剩余邮箱滤波器。

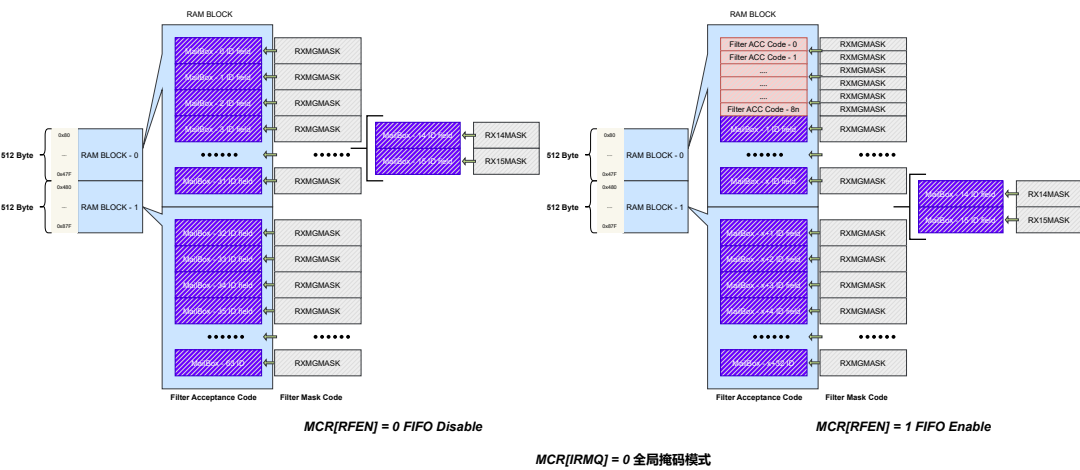
3.5.1 全局掩码模式 MCR[IRMQ] = 0

3.5.1.1 全局掩码模式下 FIFO 禁用时邮箱滤波器掩码和接受码

- 掩码：当 FIFO 禁用时，全局掩码模式下，仅 MailBox-14、MailBox-15 使用掩码寄存器 RX14MASK、RX15MASK 作为掩码，其他所有邮箱的掩码均使用掩码寄存器 RXMGMASK。
- 接受码：邮箱中的信息段（Header）中 ID 位域作为接受码。

3.5.1.2 全局掩码模式下 FIFO 使能时 FIFO 滤波器掩码和接受码

- FIFO 掩码：FIFO 滤波器组的掩码均为 RXMGMASK
- FIFO 接受码：Filter ACC Code – 0 ~ Filter ACC Code – 8n



CTRL2[RFEN] 滤波器接收码数量		分配到 FIFO 的独立滤波器掩码寄存器数量	分配到 FIFO 的独立滤波器掩码寄存器	全局掩码寄存器用于滤波器组的元素编号
0	8	8	RXIMR0~RXIMR7	No Use
1	16	10	RXIMR0~RXIMR9	Filter Table Element 10~15
2	24	12	RXIMR0~RXIMR11	Filter Table Element 12~23
3	32	14	RXIMR0~RXIMR13	Filter Table Element 14~31
4	40	16	RXIMR0~RXIMR15	Filter Table Element 16~39
5	48	18	RXIMR0~RXIMR17	Filter Table Element 18~47
6	56	20	RXIMR0~RXIMR19	Filter Table Element 20~55
7	64	22	RXIMR0~RXIMR21	Filter Table Element 22~63
8	72	24	RXIMR0~RXIMR23	Filter Table Element 24~71
9	80	26	RXIMR0~RXIMR25	Filter Table Element 26~79
10	88	28	RXIMR0~RXIMR27	Filter Table Element 28~87
11	96	30	RXIMR0~RXIMR29	Filter Table Element 30~95
12	104	32	RXIMR0~RXIMR31	Filter Table Element 32~103
13	112	32	RXIMR0~RXIMR31	Filter Table Element 32~111
14	120	32	RXIMR0~RXIMR31	Filter Table Element 32~119
15	128	32	RXIMR0~RXIMR31	Filter Table Element 32~127

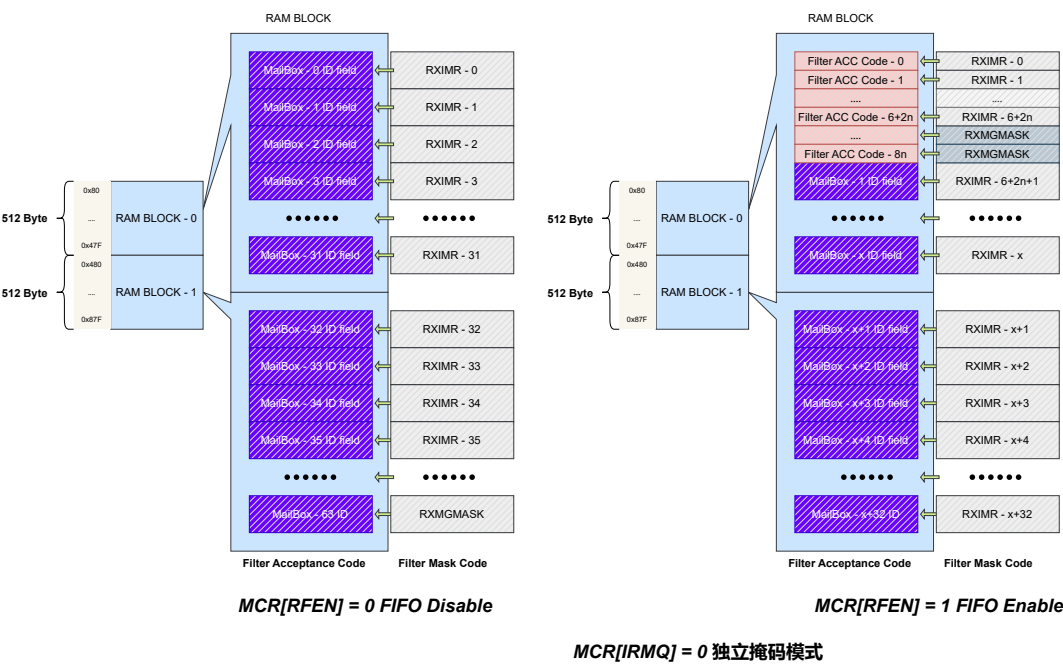
3.5.2 独立掩码模式 MCR[IRMQ] = 1

3.5.2.1 独立掩码模式下 FIFO 禁用时邮箱滤波器掩码和接受码

- 掩码：当 FIFO 禁用时，独立掩码模式下，MailBox-0 ~ MailBox-64 使用掩码寄存器 RXIMR(0)~ RXIMR(64) 作为掩码
- 接受码：邮箱中的信息段（Header）中 ID 位域作为接受码。

3.5.2.2 独立掩码模式下 FIFO 使能时 FIFO 滤波器掩码和接受码

- FIFO 掩码：当 FIFO 使能时，RXIMR(0)~ RXIMR(6+2n) 作为 FIFO 滤波器组前 6+2n 个滤波器掩码，RXMGMASK 作为 6+2n +1~8n 个滤波器掩码
- FIFO 接受码：Filter ACC Code – 0 ~ Filter ACC Code – 8n
- 邮箱掩码：剩余 RAM 如果被分配为接收邮箱，接收的掩码为 RXIMR(6+2n+1)~ RXIMR(x)
- 邮箱接受码：邮箱中的信息段（Header）中 ID 位域作为接受码。

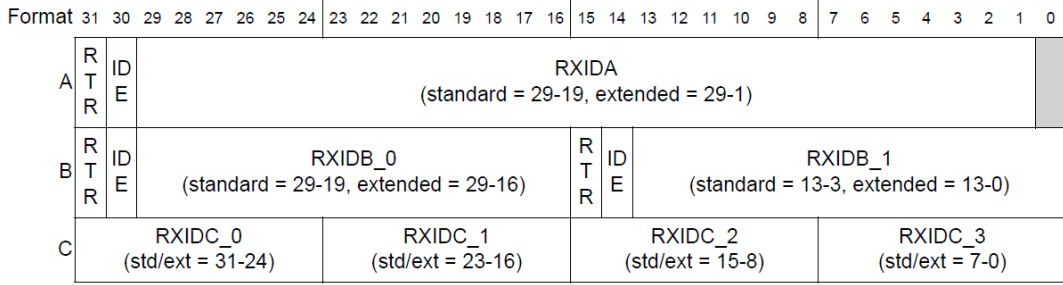


3.5.3 FIFO 滤波器格式 FIFO Filter Format

当 FIFO 使能时，滤波器组中的滤波器可通过 MCR[IDAM] 被设置为 3 中不同的格式。

滤波器格式 Filter Format	滤波模式 Filter Mode	滤波规则 ¹ Filter rule
A	单滤波器模式	所有 ID 位比较
B	双滤波器模式	标准帧时所有 ID 位比较扩展帧时 MSB 14 位 ID 比较
C	四滤波器模式	MSB 8 位比较

Format-B 及 Format-C 多应用 ID 段滤波



RTR：远程帧指示位
IDE：扩展帧指示位

4 CAN 波特率 CAN Baud rate

CAN 波特率由位时间决定，位时间由 T_q 值和 T_q 数决定 N ， T_q 值由 FlexCAN 模块时钟 f_{CANCLK} 及分频系数 $PRESDIV$ 确定。

$$T_q = \frac{PRESDIV + 1}{f_{CANCLK}}$$
$$Baud\ rate = \frac{1}{N \bullet T_q} = \frac{f_{CANCLK}}{N \bullet (PRESDIV + 1)}$$
$$Bit\ Time = N \bullet T_q$$

4.1 FlexCAN 模块时钟

FlexCAN 模块时钟可通过 CTRL1[CLKSRC] 选择

- CTRL1[CLKSRC] = 0: 外部高频晶振作为 CAN 模块时钟源, $f_{CANCLK} = f_{FXOSC}$
- CTRL1[CLKSRC] = 1: 外设时钟作为 CAN 模块时钟源, $f_{CANCLK} = f_{CANPE}$ 。此时钟源 FASTBUS Clock, 即 $f_{CANCLK} = f_{CANPE} = f_{FASTBUSCLK}$

4.2 T_q 数-N

根据 CAN 协议规范, 每个位分为 4 个段, 每个段由多个 T_q 组成。FlexCAN 每个段的 T_q 数均可通过寄存器进行配置。

$$N = SYNC + PTS + PBS1 + PBS2$$

$$SSP = \frac{SYNC + PTS + PBS1}{SYNC + PTS + PBS1 + PBS2}$$

段名称 Segment Name	Tq 数			
	Classic CAN Mode	Nominal bit time (Separate prescaler)	Nominal bit time (Shared prescaler)	Data bit time
预分频 (PRES DIV)	1~32	1~32	1~32	1~32
同步段 (SS: Synchronization Segment)	1Tq	1Tq	1Tq	1Tq
传播时间段 (PTS: Propagation Time Segment)	1Tq ~ 8Tq	1Tq ~ 48Tq	1Tq ~ 96Tq	0Tq ~ 8Tq
相位缓冲段 1 (PBS1: Phase Buffer Segment 1)	1Tq ~ 8Tq	1Tq ~ 16Tq	1Tq ~ 32Tq	1Tq ~ 8Tq
相位缓冲段 2 (PBS2: Phase Buffer Segment 2)	2Tq ~ 8Tq	2Tq ~ 16Tq	2Tq ~ 32Tq	2Tq ~ 8Tq
同步补偿宽度 (SJW: reSynchronization Jump Width)	1Tq ~ 4Tq	1Tq ~ 16Tq	1Tq ~ 32Tq	1Tq ~ 8Tq

4.3 经典 CAN 模式波特率设定

在经典模式下, 可通过 CTRL1 寄存器中设计预分频数及位时间各个段 T_q 数, 也可通过 CBT 寄存器设定, 取决于 CBT[BTF] 设定。

- CBT[BTF] = 0: CAN 波特率设定取自 CTRL1 寄存器中相关位域

- 预分频数:

$$PRES DIV = CTRL1[PRES DIV]$$

- 传播时间段 T_q 数-PTS T_q

$$PTS = CTRL1[PROPSEG] + 1$$

- 相位缓冲段 2 T_q 数 -PBS1 T_q

$$PBS1 = CTRL1[PSEG1] + 1$$

- 相位缓冲段 2 T_q 数-Bs2 T_q

$$PBS2 = CTRL1[PSEG2] + 1$$

- 同步补偿宽度 T_q 数-SJW T_q

$$SJW = CTRL1[RJW] + 1$$

- 位时间 T_q 数- N

$$N = CTRL1[PROPSEG] + CTRL1[PSEG1] + CTRL1[PSEG2] + 4$$

- 采样点-SP

$$SP = \frac{CTRL1[PROPSEG] + CTRL1[PSEG1] + 3}{CTRL1[PROPSEG] + CTRL1[PSEG1] + CTRL1[PSEG2] + 4}$$

- 波特率-Baud rate

$$\frac{f_{CANCLK}}{(CTRL1[PROPSEG] + CTRL1[PSEG1] + CTRL1[PSEG2] + 4) \bullet (CTRL1[PRES DIV] + 1)}$$

• **CBT[BTF] = 1: CAN 波特率设定取自 CBT 寄存器中相关位域**

- 预分频数:

$$PRES DIV = CBT[EPRES DIV]$$

- 传播时间段 T_q 数-PTS T_q

$$PTS = CBT[EPROPSEG] + 1$$

- 相位缓冲段 2T_q 数-PBS1 T_q

$$PBS1 = CBT[EPSEG1] + 1$$

- 相位缓冲段 2 T_q 数-BS2 T_q

$$PBS2 = CBT[EPSEG2] + 1$$

- 同步补偿宽度 T_q 数-SJW T_q

$$SJW = CBT[ERJW] + 1$$

- 位时间 T_q 数- N

$$N = CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4$$

- 采样点-SP

$$SP = \frac{CBT[EPROPSEG] + CBT[EPSEG1] + 3}{CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4}$$

- 波特率-Baud rate

$$\frac{f_{CANCLK}}{(CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4) \bullet (CBT[EPRES DIV] + 1)}$$

4.4 CANFD 模式下波特率设定

在 CANFD 模式下, CBT[BTF] 必须设定为 1, CAN nominal 段位时间取自 CBT 设定和 data 段位时间自 FDCBT 设定。- 预分频数:

$$PRES DIV(nominal) = CBT[EPRES DIV]$$

$$PRES DIV(data) = FDCBT[FPRES DIV]$$

- 传播时间段 T_q 数-PTS T_q

$$PTS(nominal) = CBT[EPROPSEG] + 1$$

$$PTS(data) = FDCBT[FPROPSEG]$$

- 相位缓冲段 2T_q 数-PBS1 T_q

$$PBS1(nominal) = CBT[EPSEG1] + 1$$

$$PBS1(data) = FDCBT[FPSEG1] + 1$$

- 相位缓冲段 2 T_q 数-BS2 T_q

$$PBS2(nominal) = CBT[EPSEG2] + 1$$

$$PBS2(data) = FDCBT[FPSEG2] + 1$$

- 同步补偿宽度 T_q 数-SJW T_q

$$SJW(nominal) = CBT[ERJW] + 1$$

$$SJW(data) = FDCBT[FRJW] + 1$$

- 位时间 T_q 数- N

$$N(nominal) = CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4$$

$$N(data) = FDCBT[FPROPSEG] + FDCBT[FPSEG1] + FDCBT[FPSEG2] + 3$$

- 采样点-SP

$$SP(nominal) = \frac{CBT[EPROPSEG] + CBT[EPSEG1] + 3}{CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4}$$

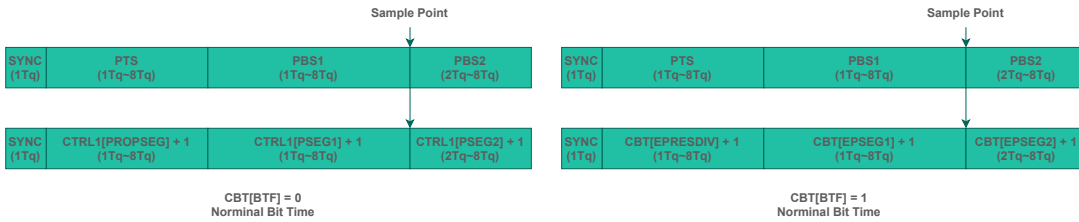
$$SP(data) = \frac{FDCBT[FPROPSEG] + FDCBT[FPSEG1] + 2}{FDCBT[FPROPSEG] + FDCBT[FPSEG1] + FDCBT[FPSEG2] + 3}$$

- Nominal 段波特率-Baud rate(nominal)

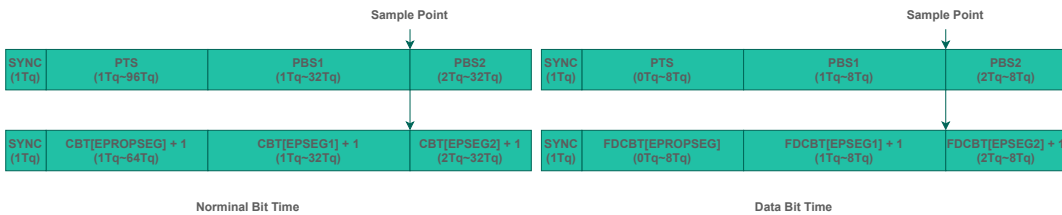
$$\frac{f_{CANCLK}}{(CBT[EPROPSEG] + CBT[EPSEG1] + CBT[EPSEG2] + 4) \bullet (CBT[EPRES DIV] + 1)}$$

- Nominal 段波特率-Baud rate(data)

$$\frac{f_{CANCLK}}{(FDCBT[FPROPSEG] + FDCBT[FPSEG1] + FDCBT[FPSEG2] + 3) \bullet (FDCBT[FPRES DIV] + 1)}$$



Classical CAN Mode



CANFD Mode

4.5 CANFD 模式下 CAN_ETDC[ETDCEN] = 0 时 SSP 设定

4.5.1 SSP 计算方法

- SSP 的 T_q 数包含 FlexCAN 控制器测量的 TDC T_q 数和设定的 TDC Offset 的 T_q 数之和

$$t_q(SSP) = t_q(TDC_{Measurement}) + t_q(TDC_{Offset})$$

- FlexCAN 控制器测量的 TDC 时间

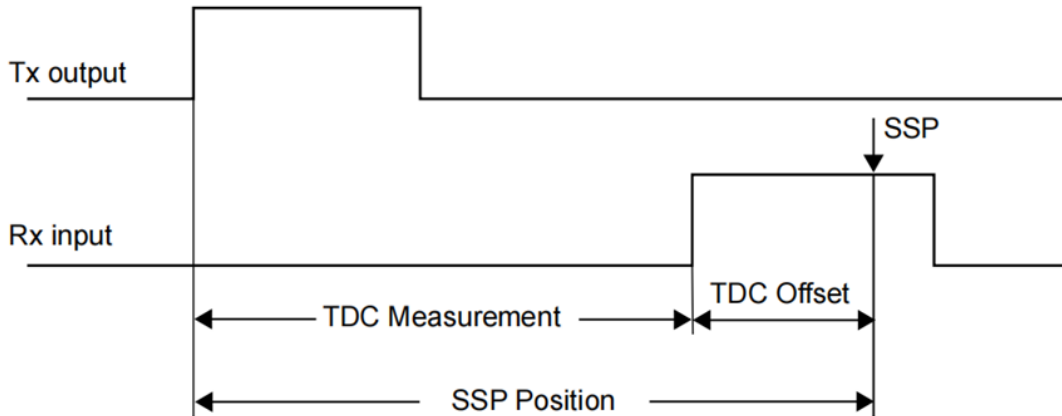
$$t(TDC_{Measurement}) = \frac{FDCTRL[TDCVAL]}{f_{CANCLK}}$$

- FlexCAN 控制器测量 TDC 的 Tq 数

$$t_q(TDC_{Measurement}) = \frac{FDCTRL[TDCVALUE]}{FDCBT[FPRES DIV] + 1}$$

- 设定的 TDC_Offset 的 Tq 数

$$t_q(TDC_{Offset}) = \frac{FDCTRL[TDCOFF]}{FDCBT[FPRES DIV] + 1}$$



4.5.2 SSP 设定方法

可以根据 data 段的采样点 SP(data) 设定来设置来设定 FDCTRL[TDCOFFSET] 值，一般保证 TDC_Offset 的 Tq 数与 data 段 SYNC+PTS+PBS1 的 Tq 数相同即可，即

$$t_q(TDC_{Offset}) = \frac{FDCTRL[TDCOFF]}{FDCBT[FPRES DIV] + 1} = FDCBT[FPROPSEG] + FDCBT[FPSEG1] + 2$$

那么：

$$FDCTRL[TDCOFF] = (FDCBT[FPROPSEG] + FDCBT[FPSEG1] + 2) \cdot (FDCBT[FPRES DIV] + 1)$$

版权与联系方式

云途半导体及其子公司保留随时对云途半导体产品或者本文档进行更改，更正，增强，修改和改进的权利，恕不另行通知。购买者在下单前应了解产品的最新相关信息。云途的产品根据云途的团队和订单确认时的销售条件进行销售。购买者对云途产品的选择权，挑选以及使用负全部责任，云途不对购买者的产品设计和应用协助承担任何责任。

云途在此未授予任何知识产权的任何明示或暗示许可。

转售与本文所述信息不同的云途产品将使云途对此类产品的任何保证失效。

云途半导体和云途半导体标志使云途半导体的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息将取代并替换之前在本文档的任何先前版本中提供的信息。

©2020 - 2022 云途半导体版权所有

联系我们:

主页: www.ytmicro.com