

متحكمات

STM32 6



موضوعات المحاضرة:

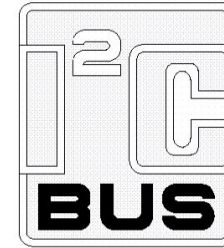
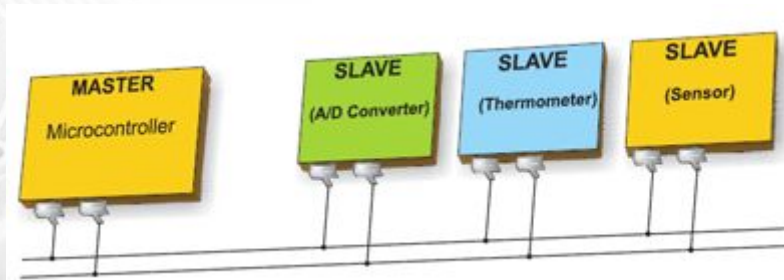
- ☐ مبدأ عمل البرتوكول I2C
- ☐ مزايا بروتوكول الاتصال التسلسلي I2C في متحكمات STM32
- ☐ المخطط الصندوقي لوحدة I2C في متحكمات STM32
- ☐ أنماط العمل المختلفة للوحدة I2C في متحكمات STM32
- ☐ دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C
- ☐ ضبط إعدادات وحدة I2C في متحكمات STM32

مقارنة بين بروتوكولات الاتصال التسلسلي الشائعة:

	RS232	RS485	I2C	SPI	M-wire	1-wire	USB	CAN
Sync/Async	Async	Async.	Sync.	Sync.	Sync.	Async.	Async.	Async.
Type	peer-peer	master/slaves	multi-master	multi-master	master/slaves	master/slaves	host/device	multi-master
Duplex	full	half	half	full	full	half	half	half
Signaling	single-ended	Differential	single-ended	single-ended	single-ended	single-ended	Differential	Differential
Max Devices No.	2	32, 128, 256	40 (cap=400pf)	8 (cap, circuit)	8 (cap, circuit)	20 (cap, power)	127 per controller	2048
Data Rate	Up to 115Kbps	Up to 35Mbps	Std.: 100kbps Fast: 400kbps Hi: 3.4Mbps	Up to 10Mbps	Up to 1Mbps	Std.: 16.3Kbps Overdrive: 142kbps	Low: 1.5Mbps Full: 12Mbps Hi : 480Mbps	Up to 1Mbps
Max. Length	15m	1200m (at 100kbps)	6m	3m	3m	300m	5m	1000m (at 62kbps)
Pin Count	2* (Tx, Rx)	2 (A, B)	2 (SDA, SCL)	3 + SS* (SI, SO, SCK)	3 + SS* (DI, DO, SK)	1 (IO)	2 (A+, A-)	2 (CAN_H, _L)
Interfacing	HW	HW	SW HW	HW SW	HW SW	HW & SW	protocol stack	HW & SW
Flow Control	HW or SW handshake	HW or SW handshake	Acknowledge from slave	None	None	CRC, Pulling	Polling by controller	CSMA / CDAMP

I2C

Inter Integrated Circuit



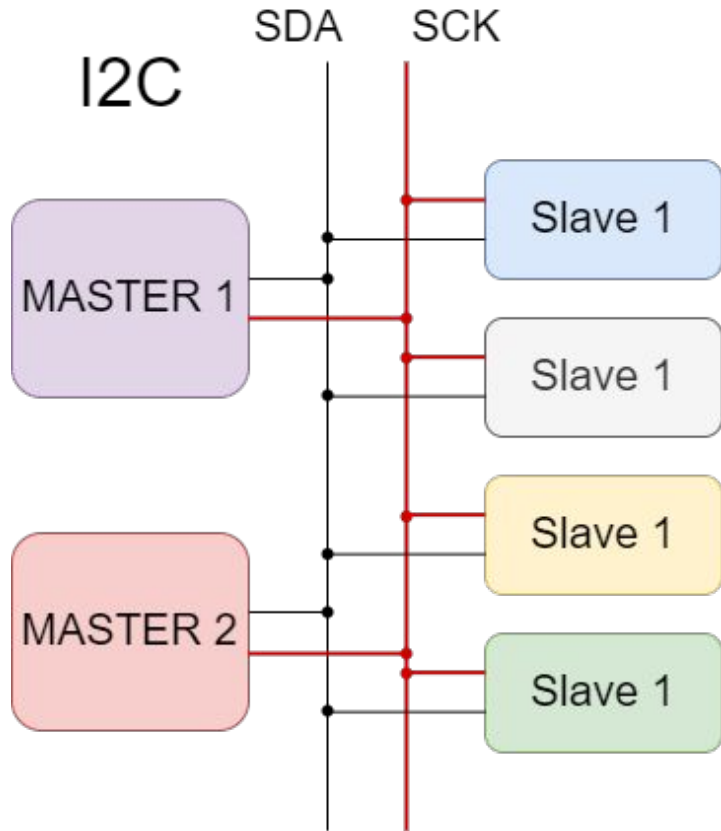
بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:

بروتوكول الاتصال التسلسلي I²C (Inter Integrated Circuit)

□ هو عبارة عن بروتوكول اتصال تسلسلي يُستخدم على نطاق واسع لتوصيل الدوائر المتكاملة الطرفية بالمتحكمات في الاتصالات قصيرة المدى.

□ تم تطويره من قبل شركة Philips في أوائل 1980.

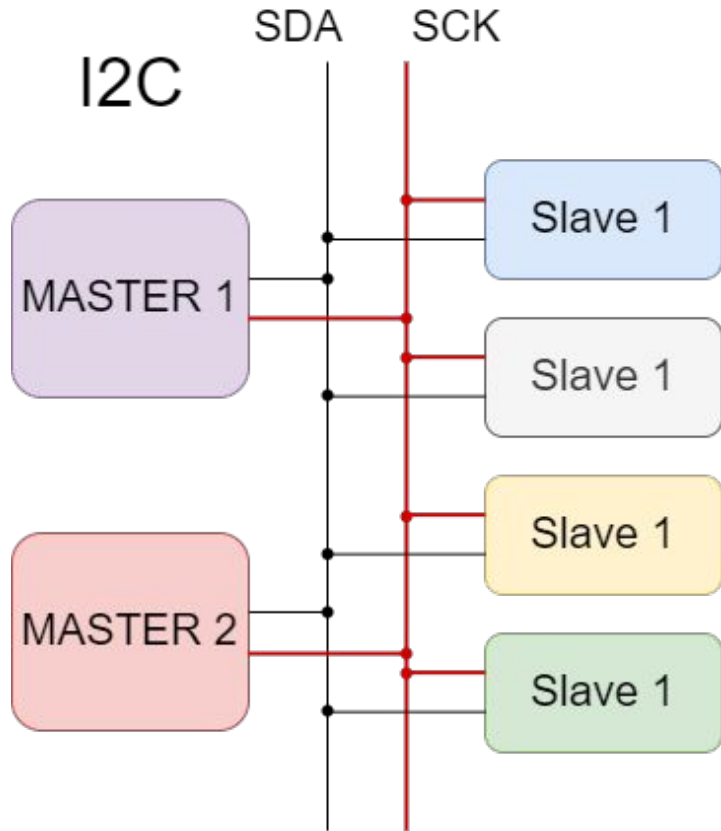
بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:



عبرة عن بروتوكول اتصال:

1. تسلسلي
2. متزامن
3. Multi-Slave و Multi-Masters
4. ثنائي الاتجاه
5. نصف مزدوج الاتجاه

بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:



□ هو ناقل بيانات **تسلسلي** يقوم بإرسال أو استقبال

البيانات على خط واحد على شكل بت بت كل مرة.

□ **متعدد Master ومتعدد Slave** لكن الوضع الشائع

هو ماستر واحد و Slave متعدد. حيث يقوم الماستر

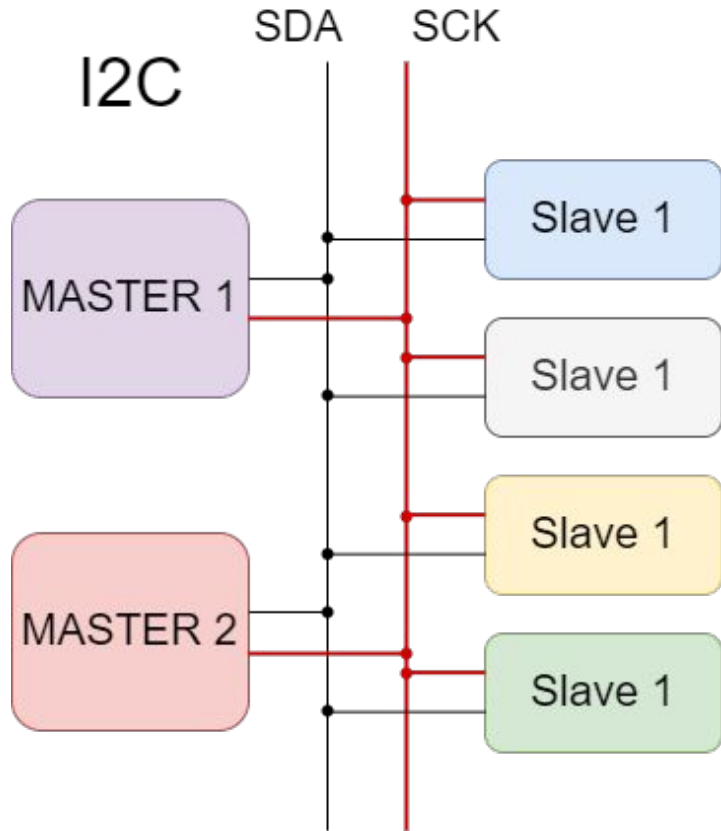
بطلب بيانات من Slave محدد عبر إرسال عنوان ال

Slave المطلوب التخاطب معه والذي يقوم بالرد.

□ يعتمد على خطين: خط SDA للبيانات وخط SCL

لنبضات الساعة مما يوفر **تزامناً** في نقل البيانات.

بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:



□ خط البيانات SDA هو خط **ثنائي الاتجاه** أي أن نقل

البيانات يتم إما من الماستر إلى الـ Slave أو من الـ Slave إلى الماستر.

□ خط التزامن SCL تتولد إشارته من عنصر Master

□ **نصف مزدوج الإتجاه** أي أن النقل باتجاه واحد فقط في

كل مرة وليس بالاتجاهين في نفس الوقت.

بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:

- ❑ في الحالة الخاملة تكون القيمة المنطقية على كلا الخطين "1".
- ❑ كل جهاز موصول على هذا الناقل يملك عنوان فريد (Unique Address) يستخدم لتحديد جهاز الـ **Slave** المراد التخاطب معه.
- ❑ العنوان الخاص بكل جهاز يكون مؤلف من 7 بتات وقد يكون بطول 10 بت. بالتالي إن الحد الأقصى لعدد Slaves يمكن أن يكون 128 لعنونة 7 بتات أو 1024 لعنونة 10 بتات.
- ❑ يمكن أن يصل الطول الأعظمي لناقل I2C إلى 6 أمتار.

بروتوكول الاتصال التسلسلي I2C في متحكمات STM32:

تنقل البيانات بأنماط سرعات مختلفة

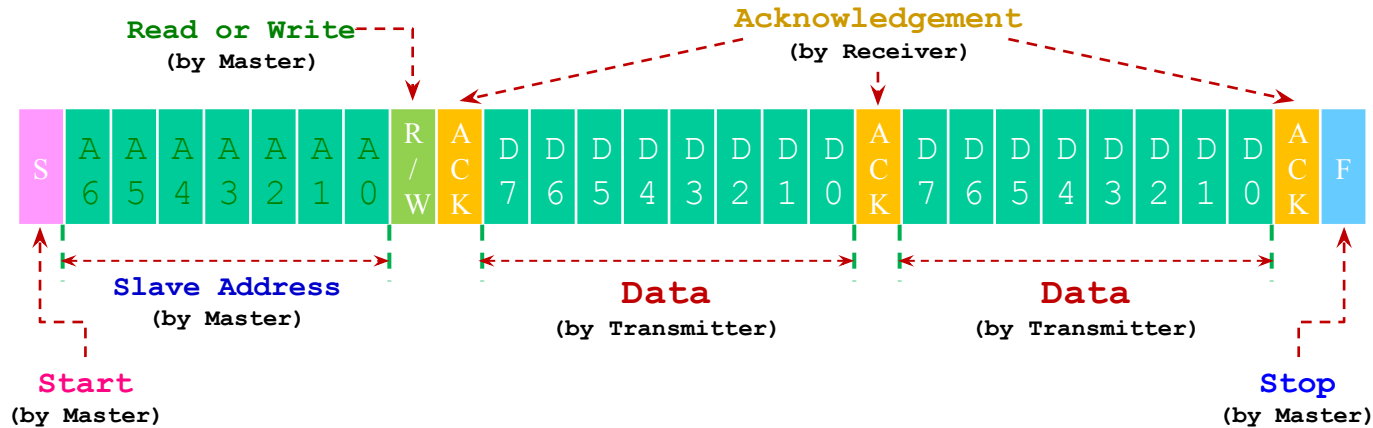
النمط	السرعة
Standard Mode	100 kbps
Fast Mode	400 kbps
Fast Mode Plus	1 mbps
High-Speed Mode	3.4 mbps
Ultra-Fast Mode	5 mbps

النمطين **Standard Mode** و **Fast Mode** مدعومين بشكل واسع في معظم الأجهزة

إطار البيانات في I2C

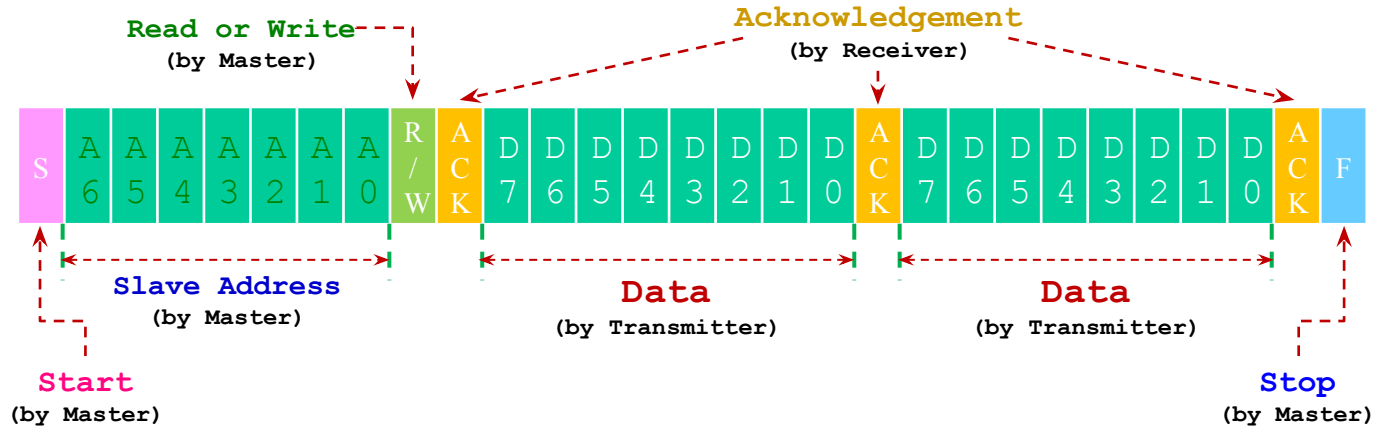
تتكون رسالة I2C من بعض العناصر الأساسية (الشروط) التي تحدث على الناقل بشكل متسلسل:

1. تبدأ دائماً بشرط بداية.
2. يتبع ذلك إطار يحوي عنوان الـ Slave المطلوب التواصل معه (7 بتات أو 10 بتات)



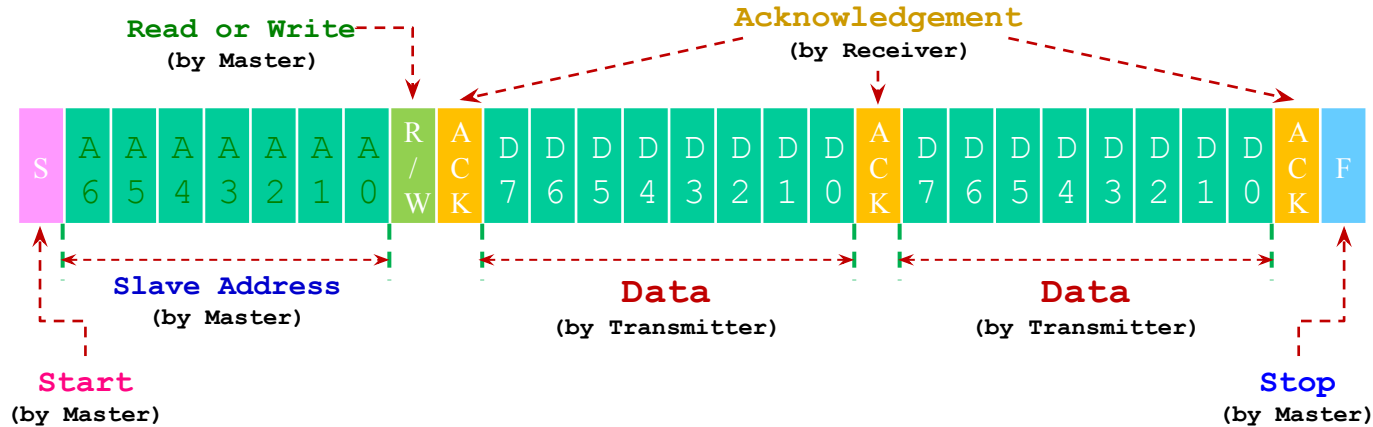
إطار البيانات في I2C

3. ثم بت قراءة/كتابة لتحديد ما إذا كان الماستر (الذي بدأ شرط S للاتصال) يريد القراءة (1) أو الكتابة (0) إلى هذا الـ Slave الذي يحمل هذا العنوان.
4. إذا كان الـ Slave ذو العنوان المحدد في إطار العنوان موجود فسوف يعيد بت (0) للمصادقة ACK أو للتأكيد الإيجابي وإلا فإن حالة البت تكون (1) وهذا يعني تأكيد سلبي NACK أي أنه لا يوجد Slave بهذا العنوان.



إطار البيانات في I2C

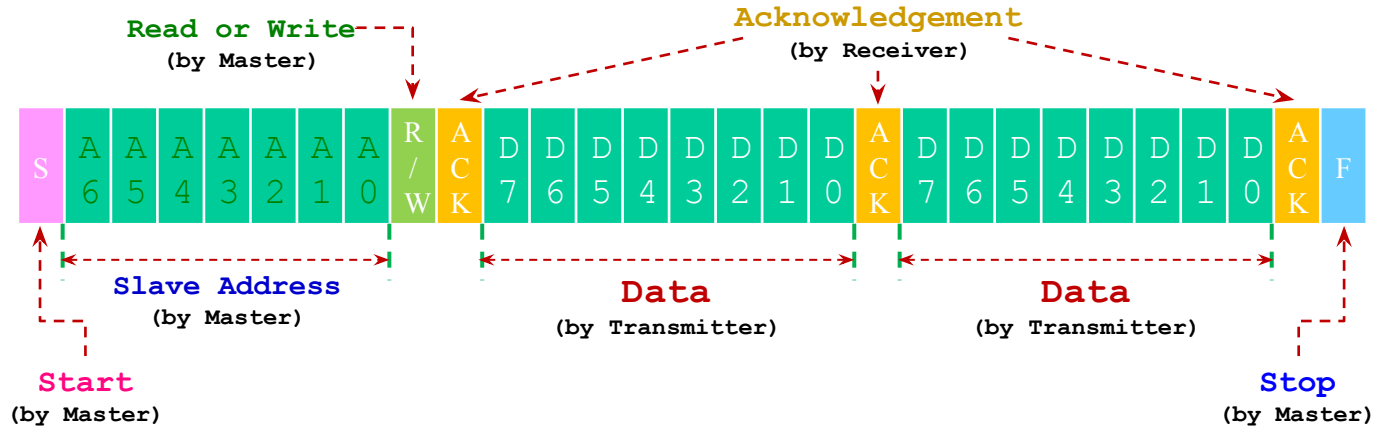
5. بعد ذلك، يتم إرسال البيانات عبارة عن بايت بايت ويكون كل بايت متبوعاً ببِت **ACK** لتأكيد استقبال البيانات من المستقبل. وأخيراً، يمكن للماستر إنهاء الاتصال عن طريق إرسال تسلسل شرط التوقف **(F)**.



إطار البيانات في I2C

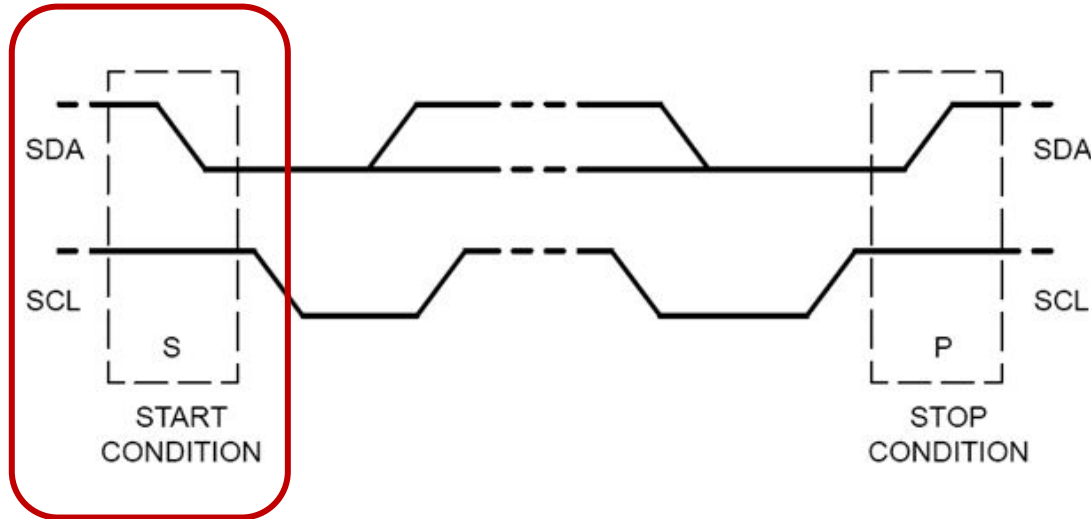
إذا كان بايت البيانات هو بايت كتابة على الـ **Slave** فإنه (أي الـ **Slave**) يقوم بالصادقة على هذا البايت عبر بت المصادقة **ACK/NACK**

أما إذا كان بايت البيانات هو بايت قراءة من الـ **Slave** فإن الماستر هو الذي يقوم بالصادقة على هذا البايت عبر بت المصادقة **ACK/NACK**



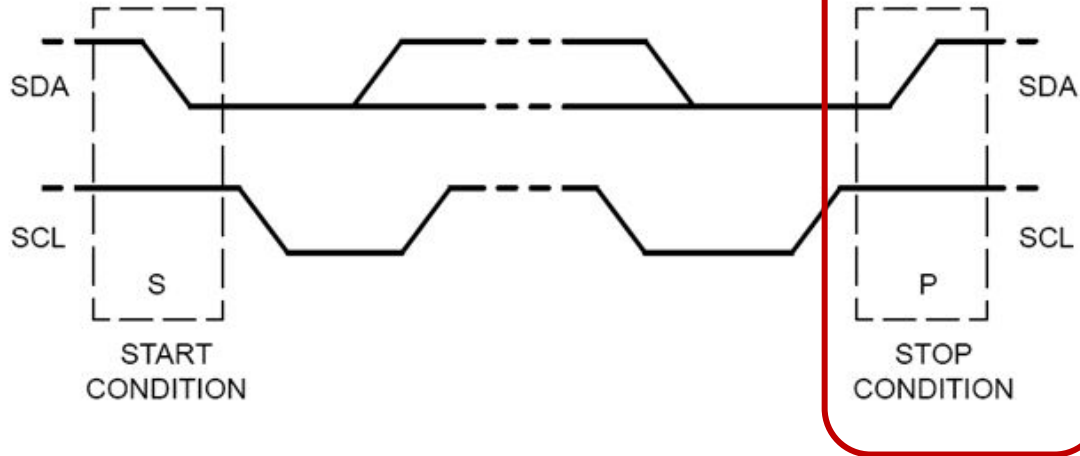
إطار البيانات في I2C

شرط بدء نقل البيانات: يقوم الماستر بسحب SDA إلى المستوى المنخفض ثم سحب SCL إلى المستوى المنخفض. يشير هذا التسلسل إلى أن هذا الماستر يطالب بنقل I2C للاتصال، مما يجبر أجهزة الماستر الأخرى الموجودة على الانتظار وهذا يمنع حدوث تصادم.



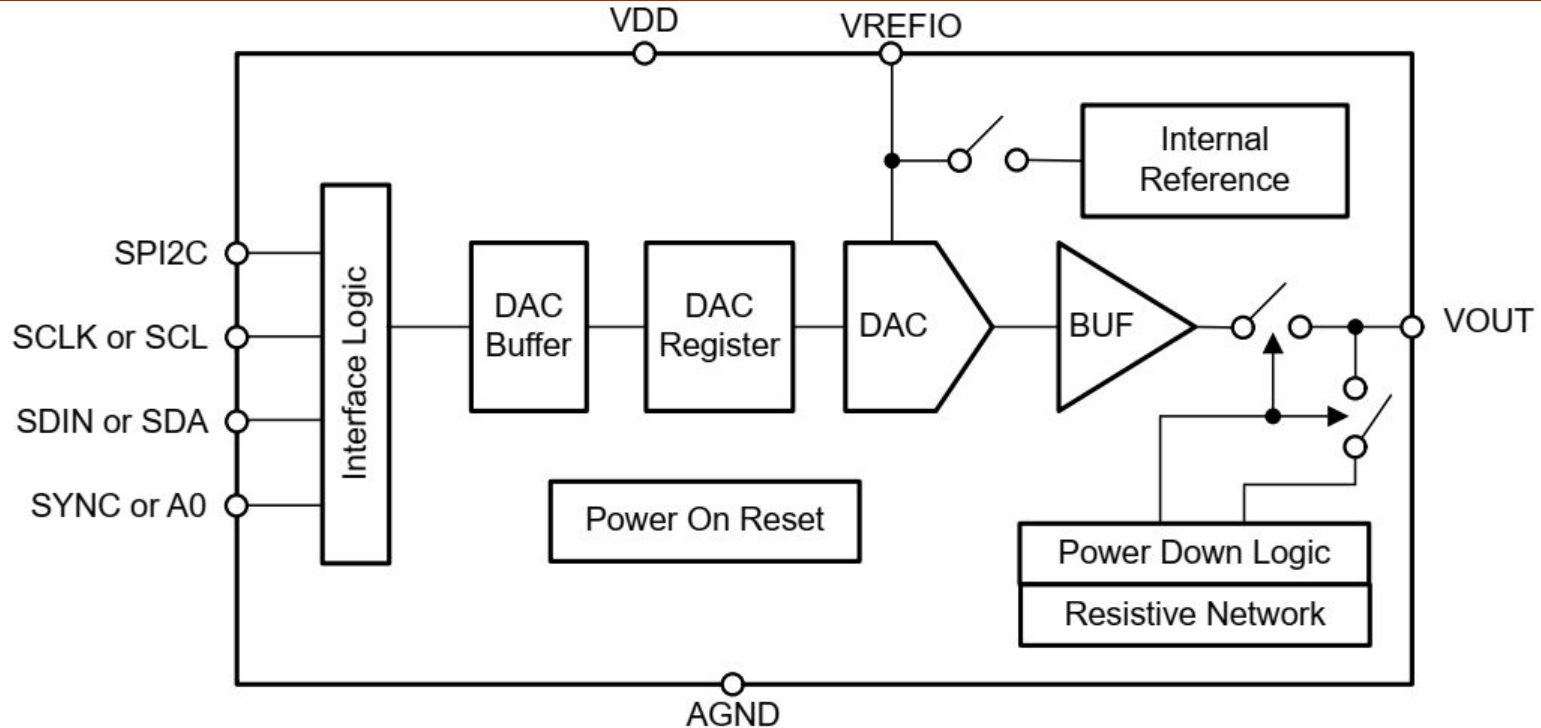
إطار البيانات في I2C

شرط إيقاف نقل البيانات: يقوم الماستر بتحرير مستوى SCL بالتالي سوف تنتقل إلى المستوى المرتفع ثم يقوم بتحرير مستوى SDA. يشير هذا إلى حالة STOP. يؤدي هذا إلى تحرير الناقل للسماح لوحدات الماستر الأخرى بالتواصل أو للسماح لنفس الماستر بالتواصل مع جهاز آخر.



مثال كتابة البيانات في I2C

كتابة البيانات على المحول الرقمي التناظري DAC80501



مثال كتابة البيانات في I2C

كتابة البيانات على المحول الرقمي التشابهي DAC80501

هذا المبدل يستخدم بروتوكول I2C لاستقبال البيانات الرقمية والتي تخزن في مسجل DAC ثم يتم تحويلها إلى قيمة تشابهية. إن هذا المبدل يحول القيم الرقمية بطول 16 بت إلى قيم تشابهية. بالتالي إن القيمة الواردة للمسجل تحتوي على بايتين.

- بفرض أن القيمة الرقمية هي 0x4CCD
- وأن عنوان المبدل هو 1001001
- وعنوان المسجل DAC هو 1000 0000

عندها إن المخطط التالي يوضح إطار بيانات I2C في عملية الكتابة هذه.

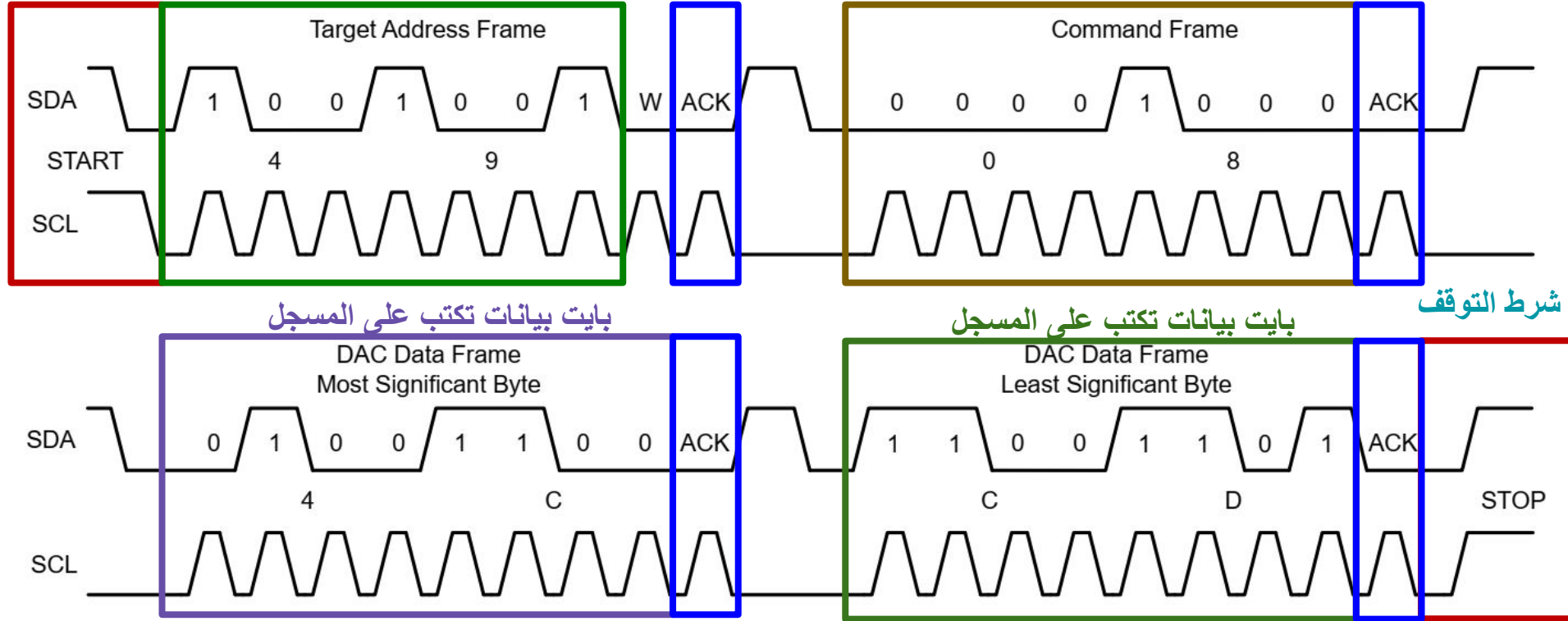
مثال كتابة البيانات في I2C

مصادقات ACK يقوم بها
ال Slave

شرط الإرسال

عنوان جهاز ال Slave

عنوان المسجل المطلوب الكتابة فيه



شرط التوقف

أنماط العمل المختلفة للوحدة I2C في متحكمات STM32

Slave transmitter ☐

Slave receiver ☐

Master transmitter ☐

Master receiver ☐

حيث يكون الوضع الافتراضي لوحدة I2C هو Slave mode ، كما يمكن أن تعمل
بنمط :

Polling, Interrupt, DMA ☐

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ DMA

□ فحص حالة جهاز مرتبط على الناقل I2C:

```
HAL_I2C_IsDeviceReady (I2C_HandleTypeDef * hi2c, uint16_t  
DevAddress, uint32_t Trials, uint32_t Timeout);
```

- وتستخدم هذه الدالة للتأكد من وجود slave device على الناقل I2C وأنه يعمل بشكل جيد أم لا.

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ Polling

Master Transmission

HAL_I2C_Master_Transmit (I2C_HandleTypeDef * hi2c, uint16_t DevAddress, uint8_t* pData, uint16_t Size, uint32_t Timeout);

DevAddress عنوان الجهاز وهو مكون من 7 بت ويجب إزاحته نحو اليسار بمقدار بت واحد قبل استدعاء الطرفية للتحويل من 8 بت إلى 7 بت. **pData**: مؤشر للبيانات المراد إرسالها، **Size**: حجم البيانات، **Timeout**: مدة المهلة قبل فشل الإرسال

Master Reception

HAL_I2C_Master_Receive (I2C_HandleTypeDef * hi2c, uint16_t DevAddress, uint8_t* pData, uint16_t Size, uint32_t Timeout);

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ Polling

Slave Transmission ☐

HAL_I2C_Slave_Transmit (I2C_HandleTypeDef * **hi2c**, uint8_t * **pData**, **uint16_t Size**, uint32_t **Timeout**);

Slave Reception ☐

HAL_I2C_Slave_Receive (I2C_HandleTypeDef * **hi2c**, uint8_t * **pData**, **uint16_t Size**, uint32_t **Timeout**);

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ Interrupt

Master Transmission ☐

```
HAL_I2C_Master_Transmit_IT (I2C_HandleTypeDef * hi2c, uint16_t  
DevAddress, uint8_t * pData, uint16_t Size);
```

- بعد استدعاء هذه الدالة تقوم وحدة I2C ببدء عملية الإرسال لكامل البايتات الموجودة ضمن الـ buffer وعند الانتهاء يتم استدعاء التابع التالي:

```
void HAL_I2C_MasterTxCpltCallback (I2C_HandleTypeDef * hi2c)  
{  
    // TX Done .. Do Something!  
}
```


دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع ال-Interrupt

Master Reception

```
HAL_I2C_Master_Receive_IT (I2C_HandleTypeDef * hi2c, uint16_t  
DevAddress, uint8_t * pData, uint16_t Size);
```

- بعد استدعاء هذه الدالة تقوم وحدة I2C ببدء عملية الاستقبال لكامل البايتات الموجودة ضمن ال-
buffer وعند الانتهاء يتم استدعاء التابع التالي:

```
void HAL_I2C_MasterRxCpltCallback (I2C_HandleTypeDef * hi2c)  
{  
    // RX Done .. Do Something!  
}
```

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ DMA

Master Transmission

HAL_I2C_Master_Transmit_DMA (I2C_HandleTypeDef * hi2c, uint16_t DevAddress, uint8_t * pData, uint16_t Size);

- بعد استدعاء هذه الدالة تقوم وحدة I2C ببدء عملية الإرسال لكامل البايتات الموجودة ضمن الـ buffer وعند الانتهاء يتم استدعاء التابع التالي:

```
void HAL_I2C_MasterTxCallback (I2C_HandleTypeDef * hi2c)
{
    // TX Done .. Do Something!
}
```

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C في وضع الـ DMA

Master Reception

```
HAL_I2C_Master_Receive_DMA (I2C_HandleTypeDef * hi2c,  
uint16_t DevAddress, uint8_t * pData, uint16_t Size);
```

- بعد استدعاء هذه الدالة تقوم وحدة I2C ببدء عملية الاستقبال لكامل البايتات الموجودة ضمن الـ buffer وعند الانتهاء يتم استدعاء التابع التالي:

```
void HAL_I2C_MasterRxCallback (I2C_HandleTypeDef * hi2c)  
{  
    // RX Done .. Do Something!  
}
```

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C

- إذا أردنا إرسال أو استقبال بيانات من مسجلات محددة ضمن الجهاز المرسل له ، على سبيل المثال إذا أردنا إرسال بيانات إلى مسجلات محددة من ذاكرة eeprom متصلة مع المتحكم عبر الناقل I2C نستخدم الدوال التالية:

□ لاستقبال البيانات (نمط polling)

```
HAL_I2C_Mem_Read(I2C_HandleTypeDef * hi2c, uint16_t  
DevAddress, uint16_t MemAddress, uint16_t MemAddSize, uint8_t *  
pData, uint16_t Size, uint32_t Timeout )
```

حيث:

MemAddress: عنوان الذاكرة الداخلية ضمن الجهاز المرسل له

MemAddSize: حجم الذاكرة الداخلية

ومن أجل نمط المقاطعة نستخدم الدالة: **HAL_I2C_Mem_Read_IT**

ومن أجل نمط الـ **DMA** نستخدم الدالة **HAL_I2C_Mem_Read_DMA**

دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C

□ لإرسال البيانات (نمط polling)

```
HAL_I2C_Mem_Write (I2C_HandleTypeDef * hi2c, uint16_t  
DevAddress, uint16_t MemAddress, uint16_t MemAddSize, uint8_t *  
pData, uint16_t Size, uint32_t Timeout )
```

حيث:

ومن أجل نمط المقاطعة نستخدم الدالة: HAL_I2C_Mem_Write_IT
ومن أجل نمط الـ DMA نستخدم الدالة HAL_I2C_Mem_Write_DMA

ضبط إعدادات وحدة I2C في متحكمات STM32

- تفعيل وحدة I2C:

I2C1 Mode and Configuration

Mode

I2C	I2C	▼
	Disable	
	I2C	
	SMBus-Alert-mode	
	SMBus-two-wire-Interface	

ضبط إعدادات وحدة I2C في متحكمات STM32

ضبط بارامترات وحدة I2C وتتضمن حالتين :

(1) عندما تكون الوحدة تعمل بنمط Master عندها يجب ضبط كل من سرعة نقل البيانات إما standard أو Fast كما يجب ضبط تردد الساعة

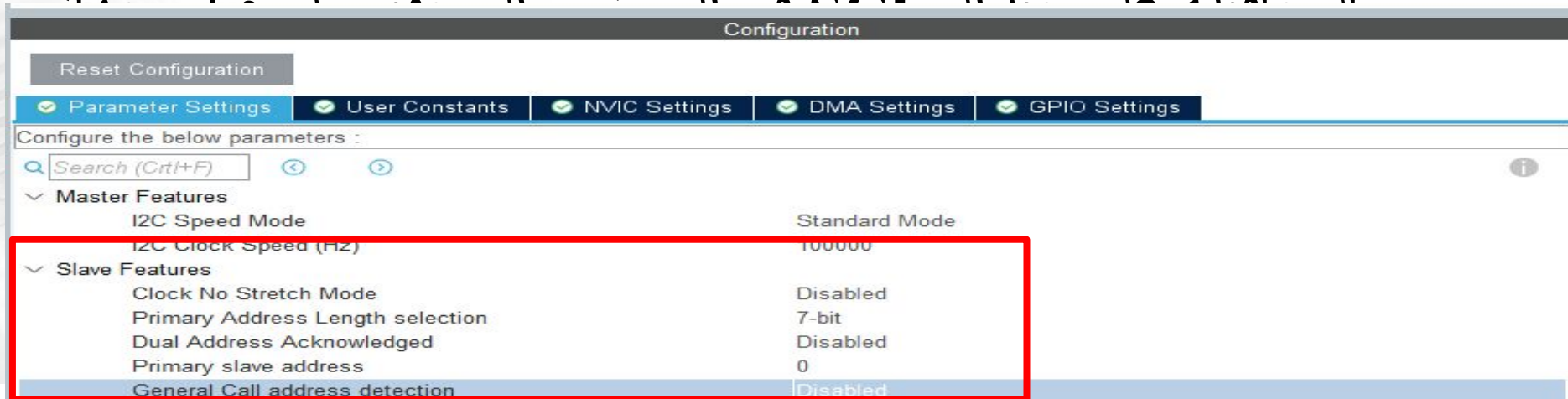
The screenshot shows the 'Configuration' window in STM32CubeMX. The 'Parameter Settings' tab is selected. Under 'Master Features', the 'I2C Speed Mode' is set to 'Standard Mode' and the 'I2C Clock Speed (Hz)' is set to '100000'. These two settings are highlighted with a red rectangle. Under 'Slave Features', several options are listed with their current states: 'Clock No Stretch Mode' is Disabled, 'Primary Address Length selection' is 7-bit, 'Dual Address Acknowledged' is Disabled, 'Primary slave address' is 0, and 'General Call address detection' is Disabled.

Parameter	Value
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

ضبط إعدادات وحدة I2C في متحكمات STM32

ضبط بارامترات وحدة I2C وتتضمن حالتين :

(2) عندما تكون الوحدة تعمل بنمط Slave عندها يمكن ضبط كل من عدد بتات العنوان وإعطاء الوحدة عنوان محدد بالإضافة إلى تفعيل أو تعطيل خاصية clock stretching والتي عند تفعيلها تقوم



Thank you for listening