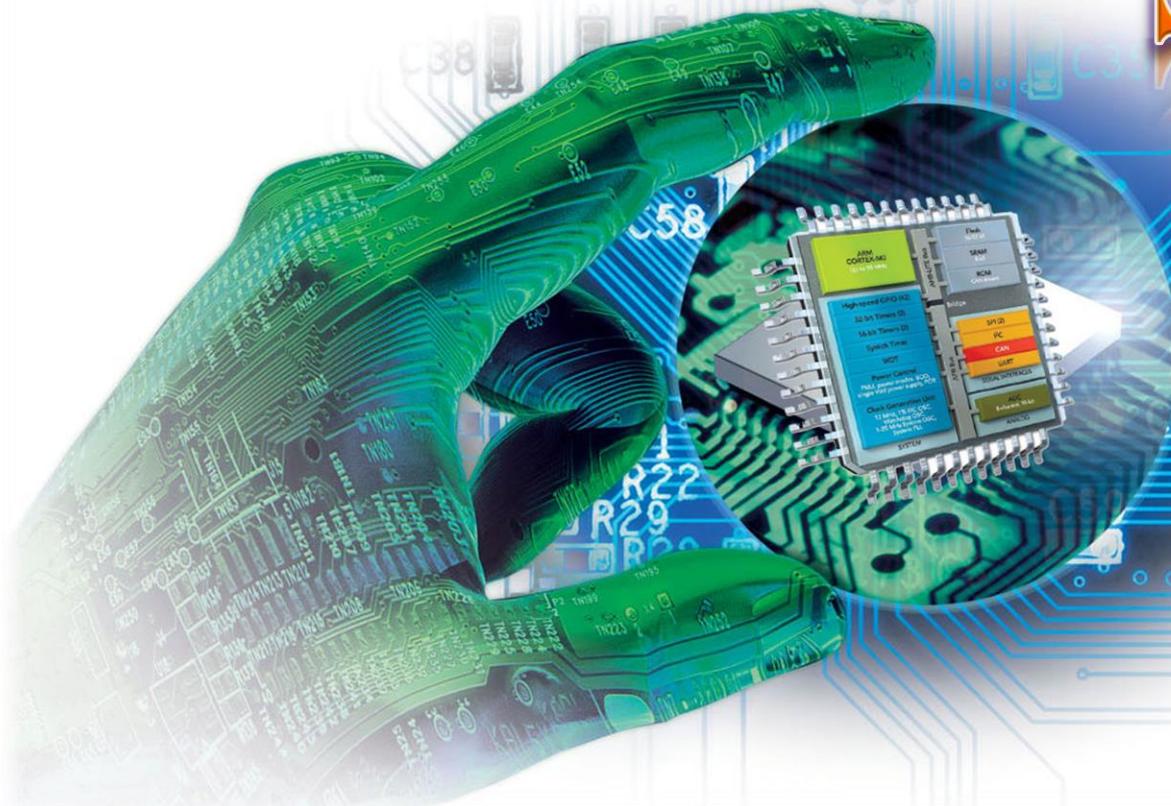
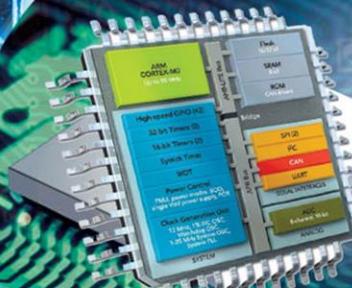


مُتَحَكِّمَات

STM32

10



موضوعات المحاضرة:

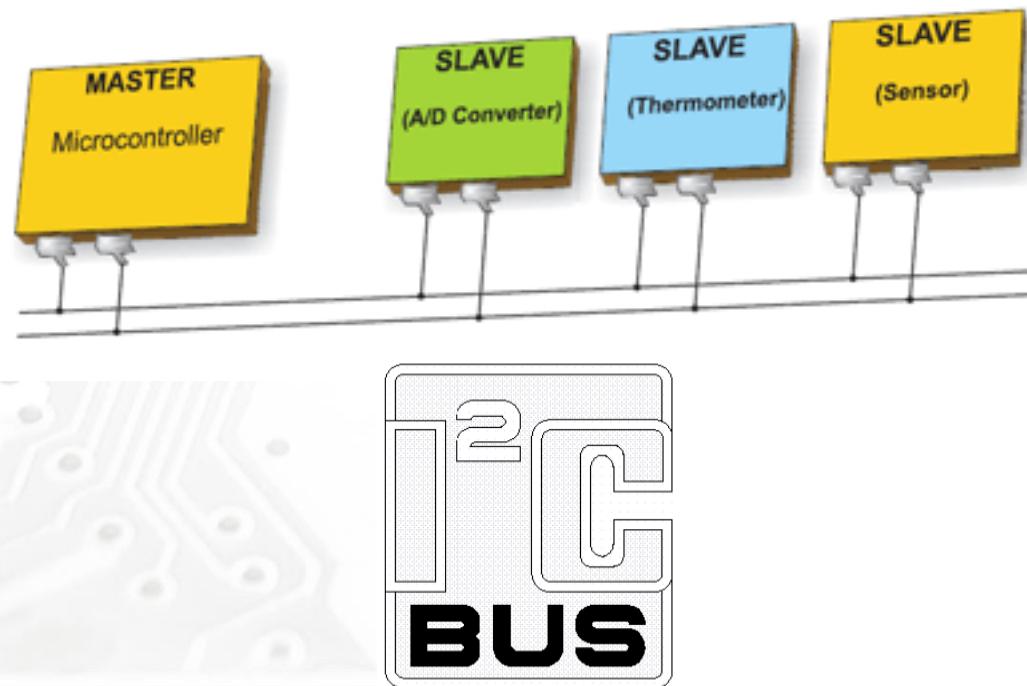
- مبدأ عمل البرتوكول I2C
- مزايا بروتوكول الاتصال التسلسلي I2C في متحكمات STM32
- المخطط الصندوقي لوحدة I2C في متحكمات STM32
- أنماط العمل المختلفة للوحدة I2C في متحكمات STM32
- دوال مكتبة HAL المستخدمة للتعامل مع وحدة I2C
- ضبط إعدادات وحدة I2C في متحكمات STM32

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 (Inter Integrated Circuit)

- تم تطويره من قبل شركة Philips في أوائل 1980.
- وضع أساساً لربط متحكم مع بعض المحيطيات في أجهزة التلفاز.
- هو بروتوكول متزامن h-duplex يعتمد على خطين (SDA, SCL).
- خط البيانات **SDA** هو خط ثبائي الاتجاه.
- خط التزامن **SCL** تتولد إشارته من عنصر Master وله حالتين:
 - أحادي الاتجاه في الأنظمة **Single-master** (1)
 - ثبائي الاتجاه في الأنظمة **Multi-master** (2)

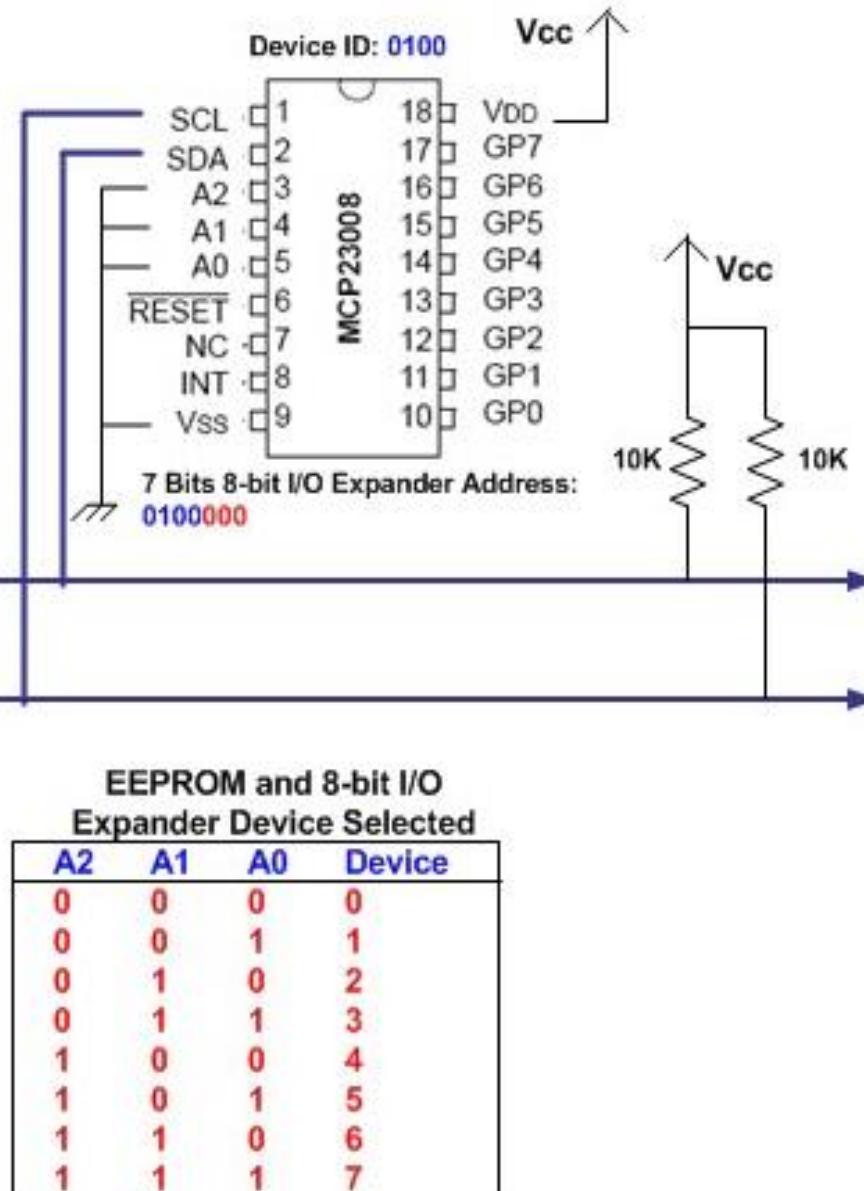
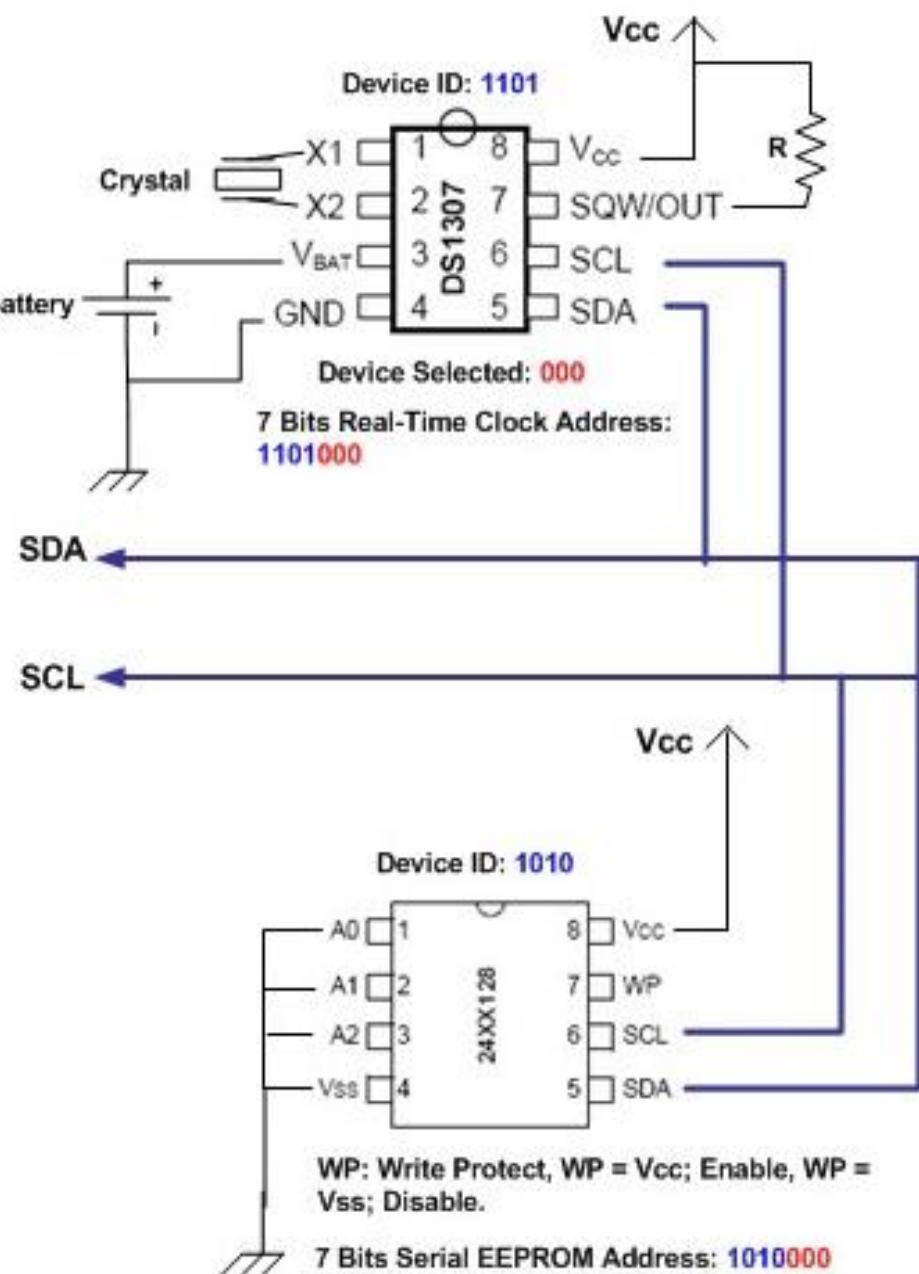
بروتوكول الاتصال التسلسلي I²C (Inter Integrated Circuit)

- خرج الأجهزة في البروتوكول I²C هو من نوع .Open collector
- يجب ربط مقاومات رفع خارجية لكلا .SDA, SCK
- في حال البطالة تكون القيمة المنطقية على كلا الخطين "1".
- كل جهاز يملك عنوان فريد (Unique Address) يستخدم لتحديد جهاز الـ "Master" المراد التخاطب معه من الجهاز "Slave".
- العنوان الخاص بكل جهاز يكون مؤلف من 7-Bit (112 nodes).
- يمكن أيضاً أن يكون العنوان الخاص بطول 10-bit (1008-node).

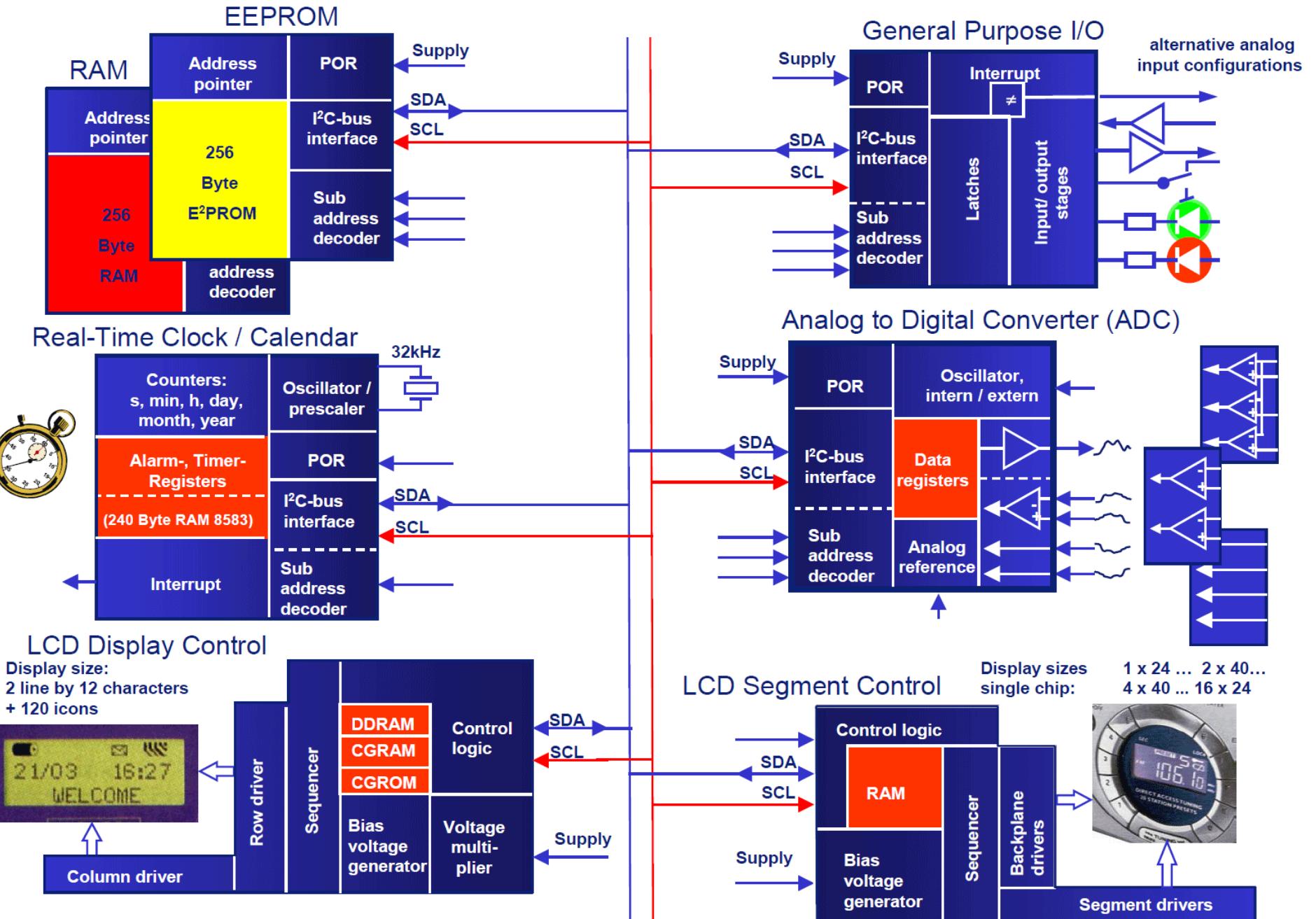
بروتوكول الاتصال التسلسلي I²C (Inter Integrated Circuit)

- إن عدد الأجهزة في البروتوكول I2C على خط النقل يعتمد مباشرة على سعة الخط، حيث أن القيمة الأعظمية للسعة يجب أن لا تتجاوز .400pF، وغالباً تكون سعة كل جهاز بحدود 10pF (40 جهاز).
- الطول الأقصى على ناقل I2C يمكن أن يصل إلى 6 أمتار.
- تختلف سرعة نقل البيانات على ناقل I2C حسب المعيار الشريحة:
 - Std.: 100kbps (1)
 - Fast: 400kbps (2)
 - Hi: 3.4Mbps (3)

بروتوكول الاتصال التسلسلي I2C في متحكمات STM32



بروتوكول الاتصال التسلسلي I2C في متحكمات STM32



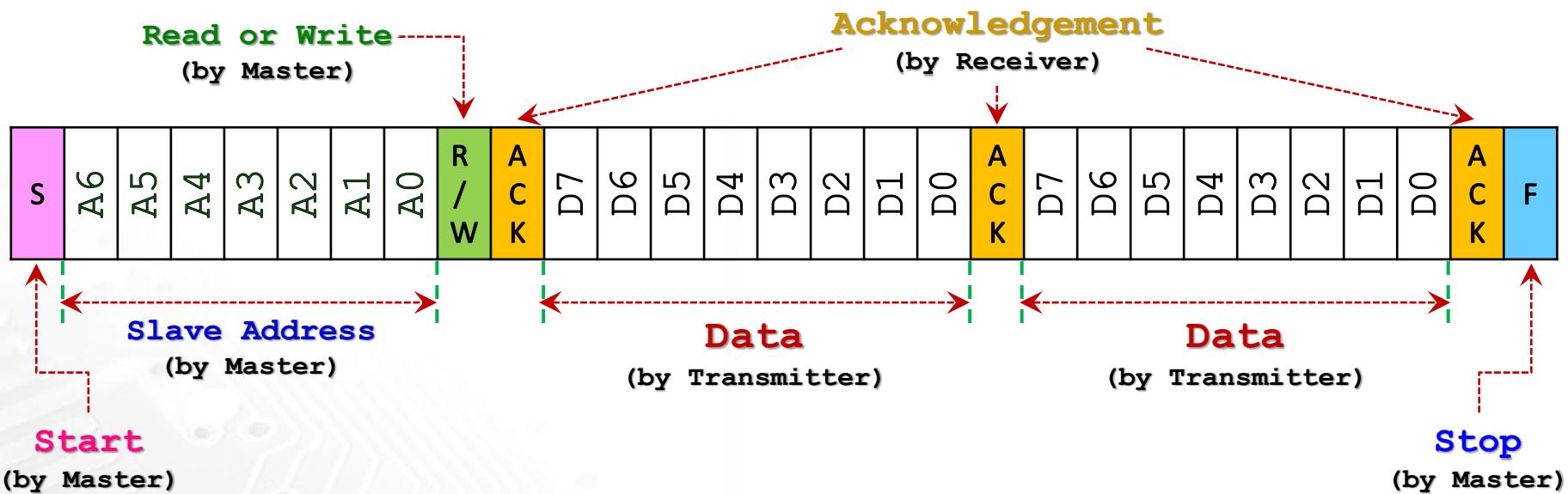
(مبدأ العمل في بروتوكول الاتصال التسلسلي I²C)

- إن مبدأ العمل يعتمد على بروتوكول Master-Slave حيث أن جهاز الـ Master يبتدئ عملية الاتصال مع جهاز Slave ذو عنوان محدد وفقاً للتسلسل التالي:
- (1) يقوم Master بإرسال حالة "Start" (بت بداية الإرسال) يقوم بإعلام جميع أجهزة Slaves الموجودة على Bus للاستعداد للبيانات القادمة على SDA.
 - (2) يقوم Master بإرسال عنوان (7-bit) الجهاز المعني "Slave Address" إضافة إلى بت تحديد العملية (قراءة "1" | كتابة "0").
 - (3) يقوم Slave المعني بالعنوان المرسل من الـ Master بالاستجابة بإشارة مصادقة "ACK" يعلم من خلالها الـ Master بوجوده على الناقل.

(مبدأ العمل في بروتوكول الاتصال التسلسلي I²C)

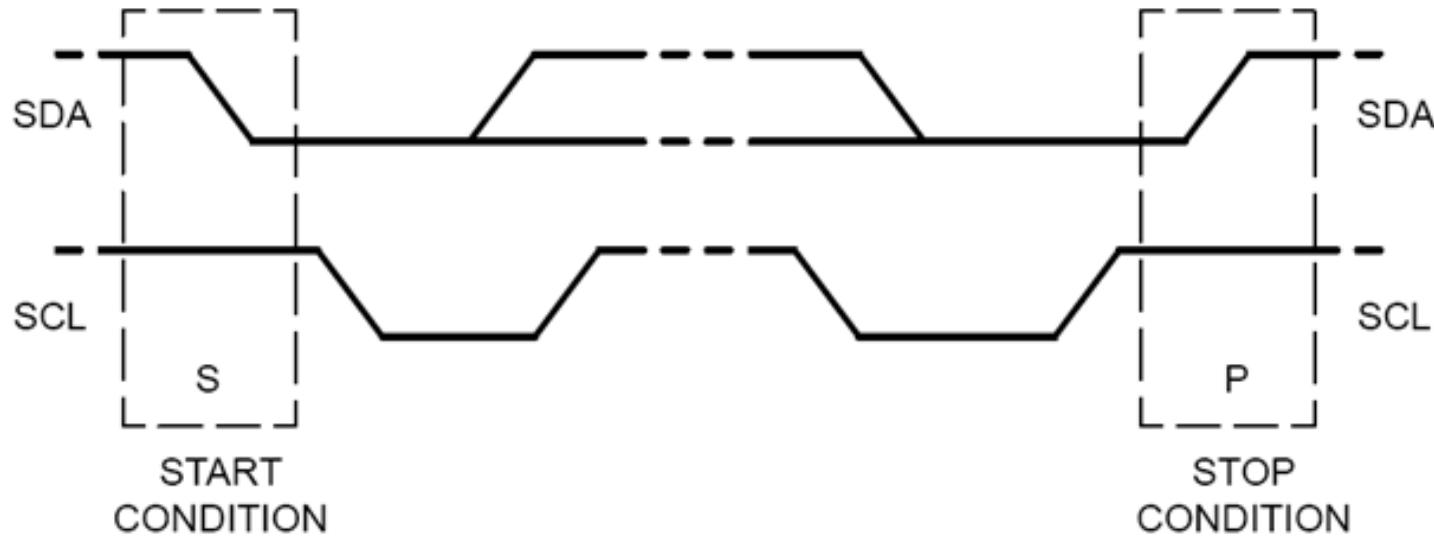
- (4) بعد أن تم تحديد طرف في التخاطب، تبدأ عملية تراسل البيانات بين الطرفين ويتم تحديد المرسل والمستقبل من خلال بت تحديد العملية (قراءة "1" \ كتابة "0") في بait العنوان.
- (5) كلما تم إرسال 8-bit من المرسل يقوم المستقبل (أياً يكن S or M) بالتأكيد بإرسال إشارة مصادقة "ACK" (1-bit). وتستمر التراسل هكذا...
- (6) عندما تنتهي عملية التراسل بين الطرفين يقوم **Master** بإرسال حالة "Stop" (بت نهاية الإرسال) مشيراً إلى انتهاء العملية الجارية.

(مبدأ العمل في بروتوكول الاتصال التسلسلي I²C)



Master controls the clock signal

(I²C) مبدأ العمل في بروتوكول الاتصال التسلسلي

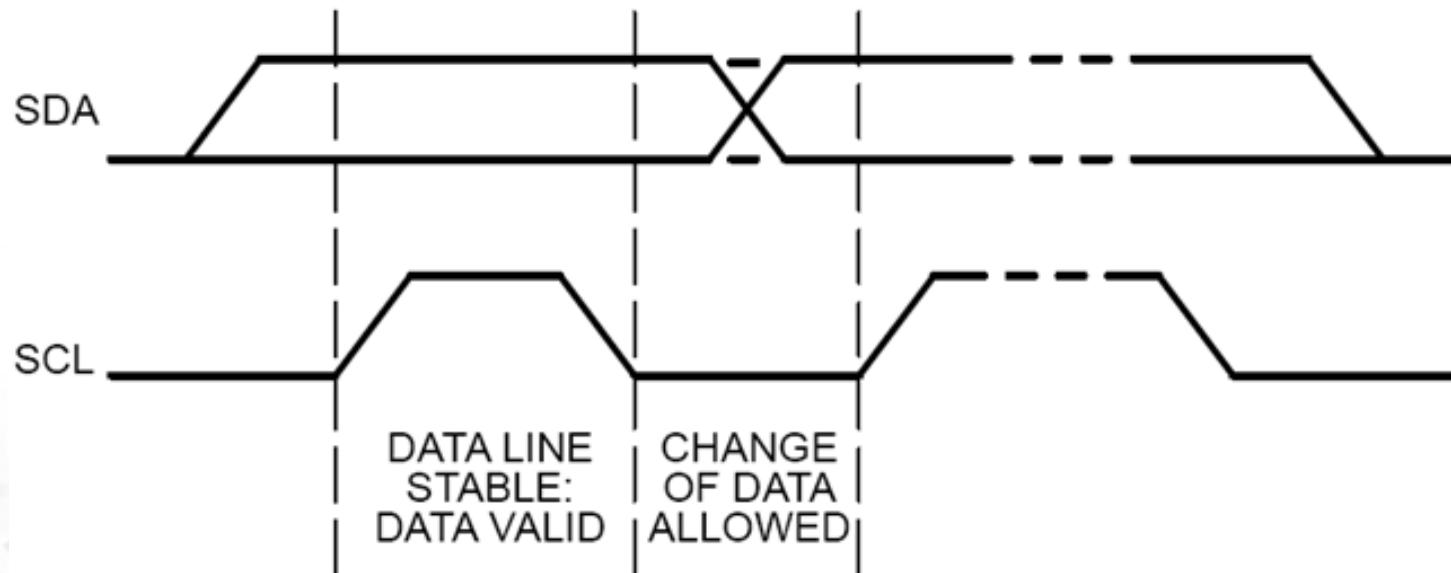


high-to-low transition
of the SDA line while
SCL line is high...

low-to-high transition
of the SDA line while
SCL line is high...

Ack: Receiver pulls SDA **Low** while
Transmitter allows it to float **High**.

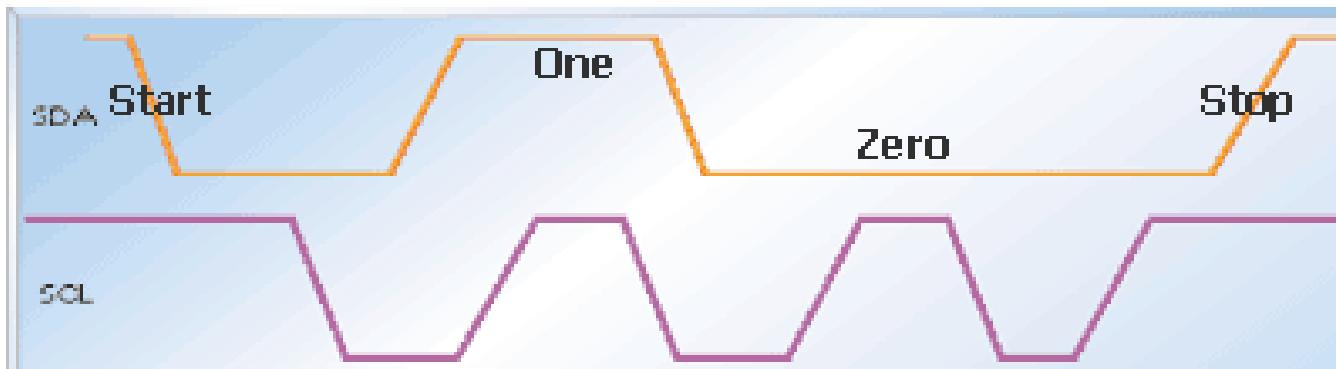
(مبدأ العمل في بروتوكول الاتصال التسلسلي I²C)



Data transition while **SCL** is **low**...

Data validation while **SCL** is **high**...

(مبدأ العمل في بروتوكول الاتصال التسلسلي I²C)



- ❑ **Write Transaction (Master Transmitting):** Master issues **Stop condition (P)** after last byte of data.
- ❑ **Read Transaction (Master Receiving):** Master does not acknowledge final byte, just issues **Stop condition (P)** to tell the slave the transmission is done.

مزايا بروتوكول الاتصال التسلسلي I2C في متحكمات STM32

يمكن أن تعمل الوحدة الطرفية Slave أو Master كـ I2C

بإمكانها توليد واكتشاف عناوين بطول 7bit أو 10bit

تحتوي على مرشحات ضجيج تشابهية ورقمية ويمكن ضبط إعداداتها من خلال الكود

يجب ضبط إعدادات أقطاب الـ Open Drain كـ I2C ووصل مقاومات رفع خارجية معها

مصدر الساعة للوحدة الطرفية قد يكون :

system clock (1)

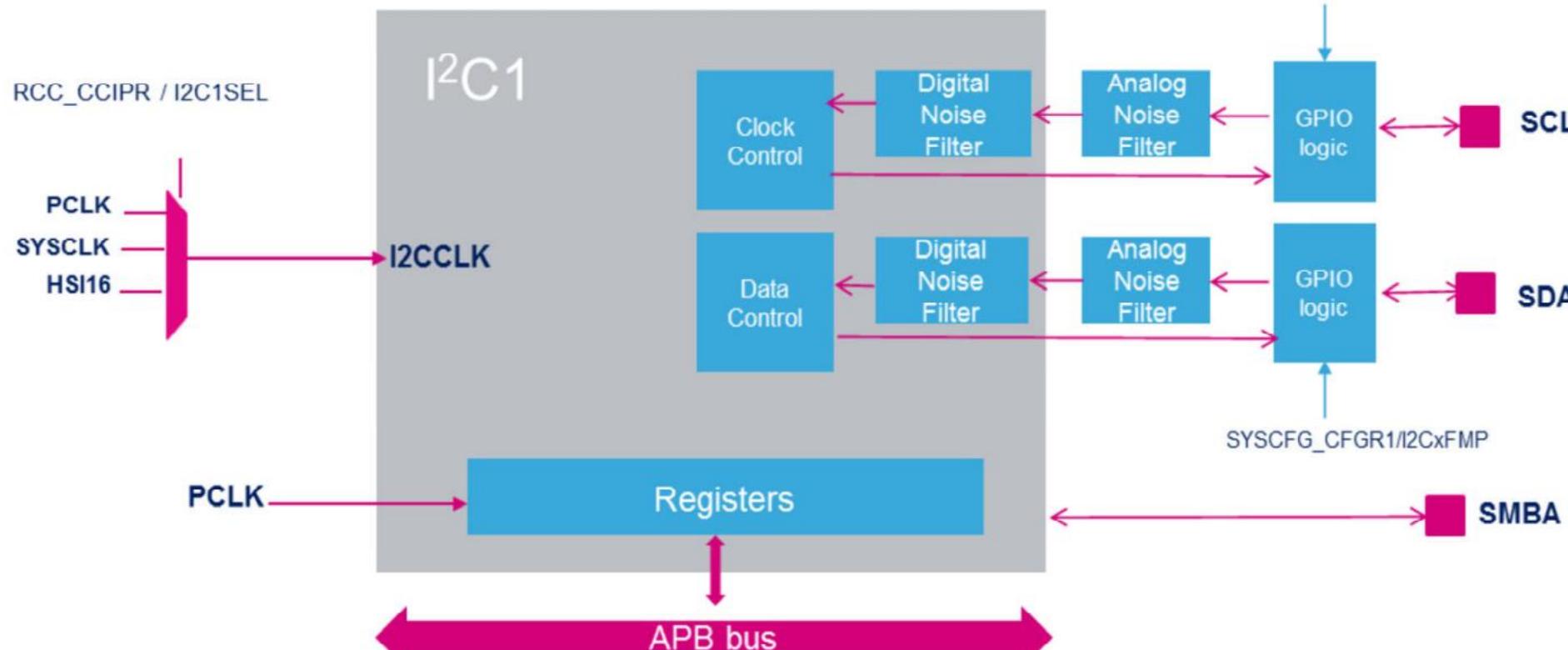
high-speed internal 16 MHz RC oscillator (2)

تدعم سرعتي نقل للبيانات على ناقل I2C :

Std.: 100kbps (1)

Fast: 400kbps (2)

المخطط الصندوقى لوحدة I2C في متحكمات STM32



أنماط العمل المختلفة لوحدة 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 ل التعامل 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);
```

حيث: **DevAddress**: عنوان الجهاز وهو مكون من 7 بت ويجب إزاحتة نحو اليسار قبل استدعاء الطرفية (لتحويل من 8 بت ل 7 بت)

pData: مؤشر الى **data buffer** 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 ل التعامل 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 ل التعامل 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 لـ كامـل الـ باـيـات المـوجـودـة ضـمـن الـ buffer وعـنـد الـ اـنـتـهـاء يـتم اـسـتـدـعـاء التـابـع التـالـي:

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

دوال مكتبة HAL المستخدمة ل التعامل مع وحدة I2C في وضع الـ 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 البيانات الموجودة ضمن الـ buffer وعند الانتهاء يتم استدعاء التابع التالي:

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

دوال مكتبة HAL المستخدمة ل التعامل مع وحدة I2C في وضع ال 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 لـ كامل البایتات 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 ل التعامل مع وحدة I2C في وضع ال 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 ال buffer وعند الانتهاء يتم استدعاء التابع التالي:

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

دوال مكتبة HAL المستخدمة للتتعامل 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 (نقط 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

: عنوان الذاكرة الداخلية ضمن الجهاز المرسل له :MemAddress

: حجم الذاكرة الداخلية :MemAddSize

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 للتتعامل مع وحدة I2C

لارسال البيانات (نط 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 ال DMA نستخدم الدالة `HAL_I2C_Mem_Write_DMA`

دوال مكتبة HAL المستخدمة ل التعامل مع وحدة I2C في وضع ال DMA

□ فحص حالة جهاز مرتبط على الناقل :I2C

```
AL_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 بشكلٍ جيد أم لا.

ضبط إعدادات وحدة I2C في متحكمات STM32

- تفعيل وحدة I2C:

I2C1 Mode and Configuration

Mode

I2C

Disable

I2C

SMBus-Alert-mode

SMBus-two-wire-Interface

ضبط بارامترات وحدة I2C وتتضمن حالتين :

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

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Master Features

I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000

Slave Features

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 بaramترات وحدة I2C وتتضمن حالتين :

- (2) عندما تكون الوحدة تعمل بنمط Slave عذها يمكن ضبط كل من عدد بิตات العنونة وإعطاء الوحدة عنوان محدد بالإضافة إلى تفعيل أو تعطيل خاصية clock stretching والتي عند تفعيلها تقوم بالمحافظة على خط الـ SCL في المستوى المنخفض لمنع أي جهاز من إجراء أي عملية لحين تحرير خط الـ master

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)   

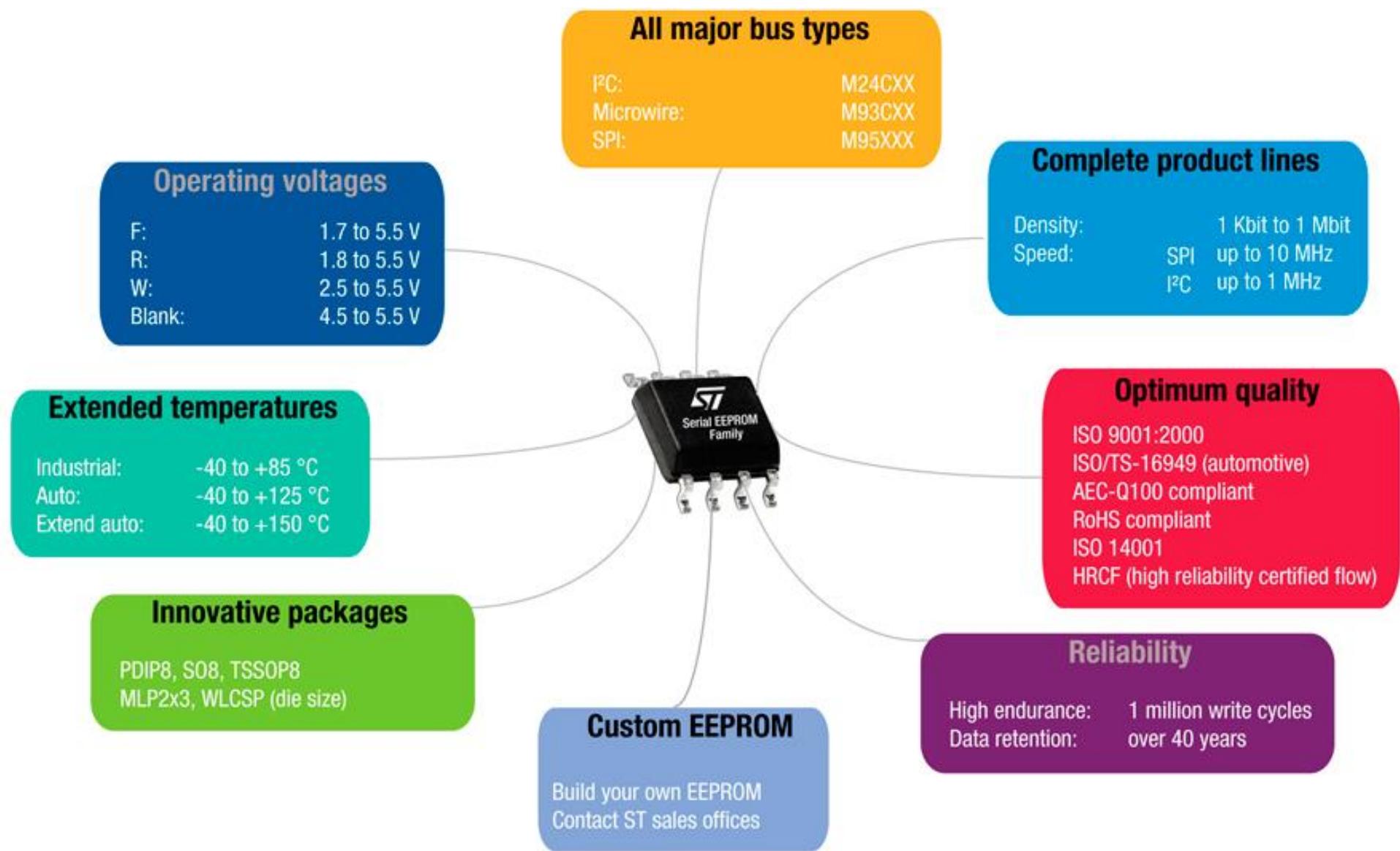
Master Features

I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000

Slave Features

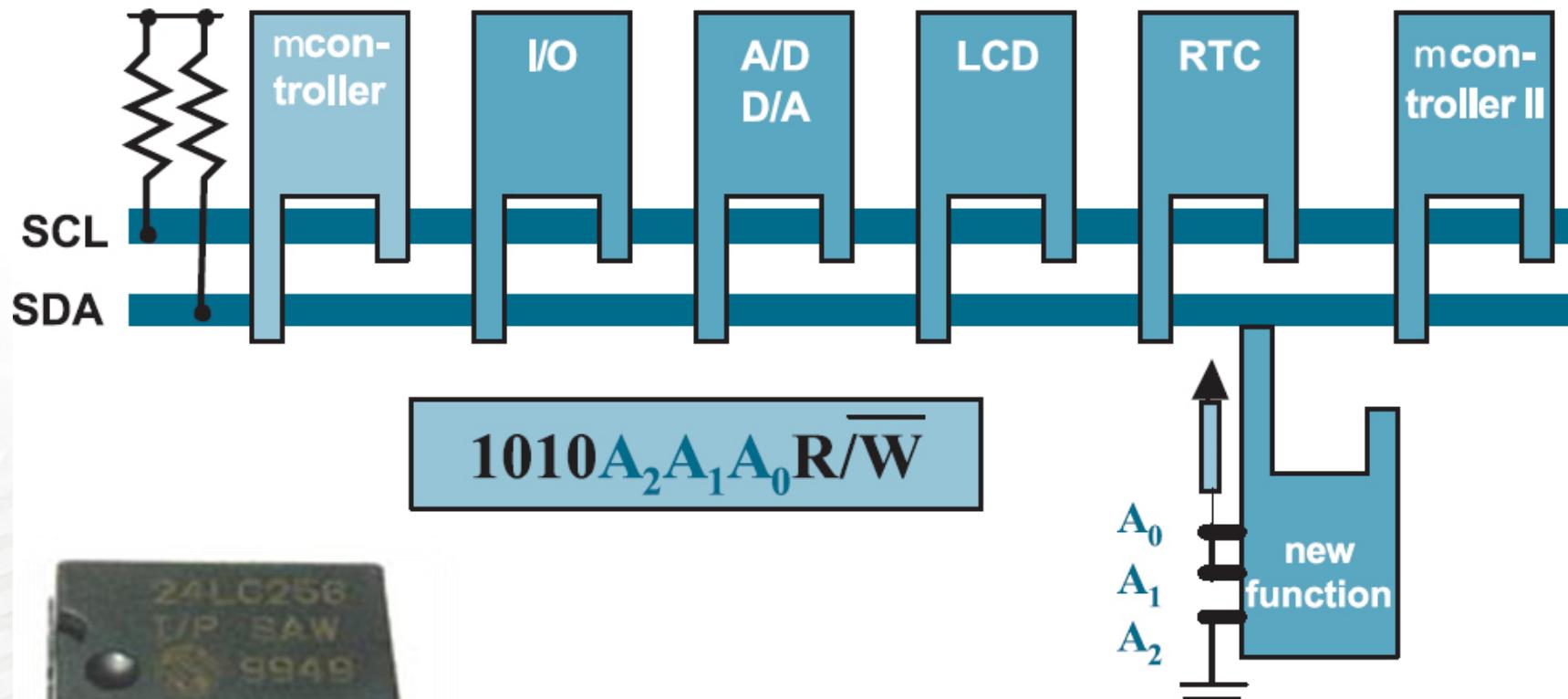
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

التطبيق العملي: ربط ذاكرة EEPROM مع المتحكم STM32G0B1CE



التطبيق العملي: ربط ذاكرة eeprom مع المتحكم

Unleash your Creativity!



AT24C02 Datasheet

Thank you for listening