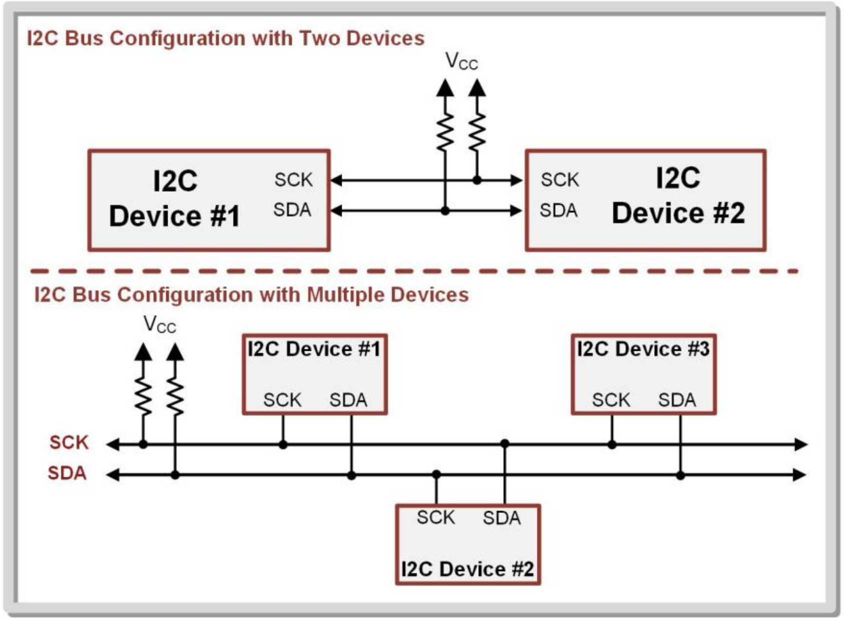
14.3.1 THE I2C PROTOCOL

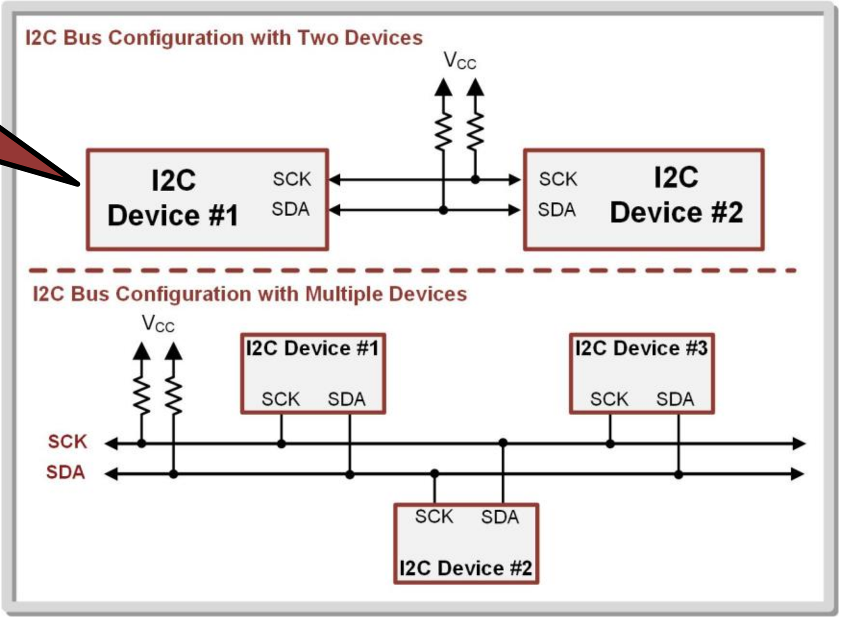
- The **Inter-Integrated Circuit (I2C)** standard is a serial interface implement with a two-wire link that can support multiple masters and multiple slaves.



14.3.1 THE I2C PROTOCOL

- The **Inter-Integrated Circuit (I2C)** standard is a serial interface implement with a two-wire link that can support multiple masters and multiple slaves.

Providing a clock allows higher data rates over UART.

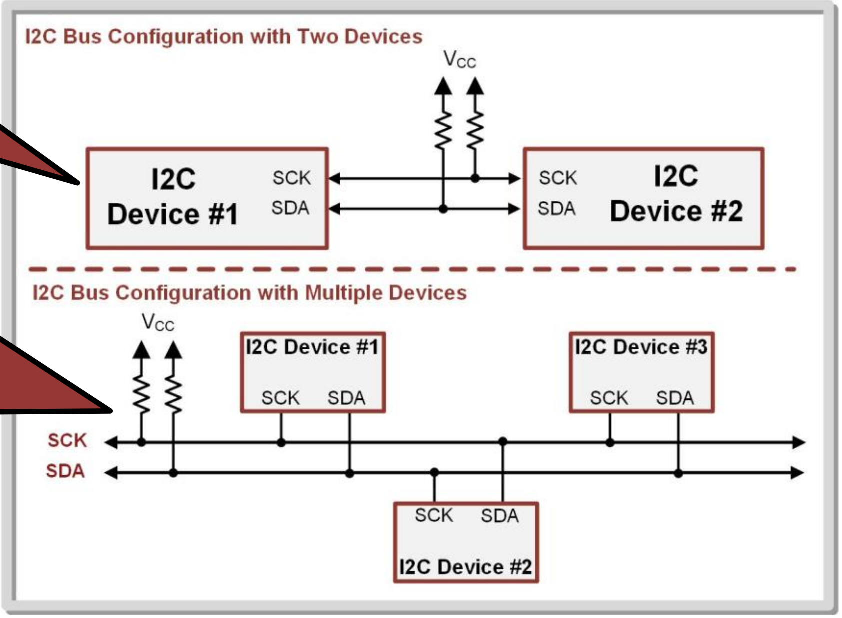


14.3.1 THE I2C PROTOCOL

- The **Inter-Integrated Circuit (I2C)** standard is a serial interface implement with a two-wire link that can support multiple masters and multiple slaves.

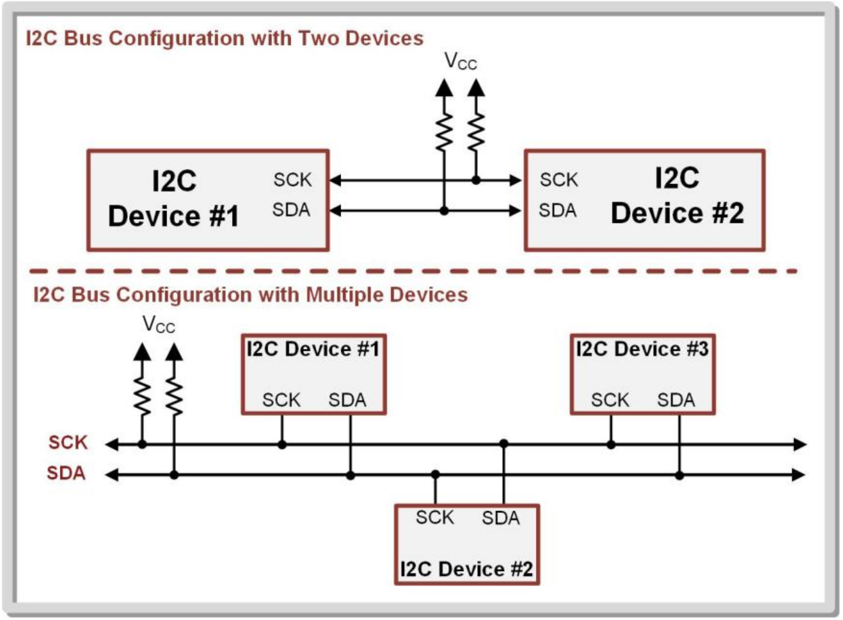
Providing a clock allows higher data rates over UART.

The two-wire interface allows devices to be added without increasing the number of lines like on SPI.



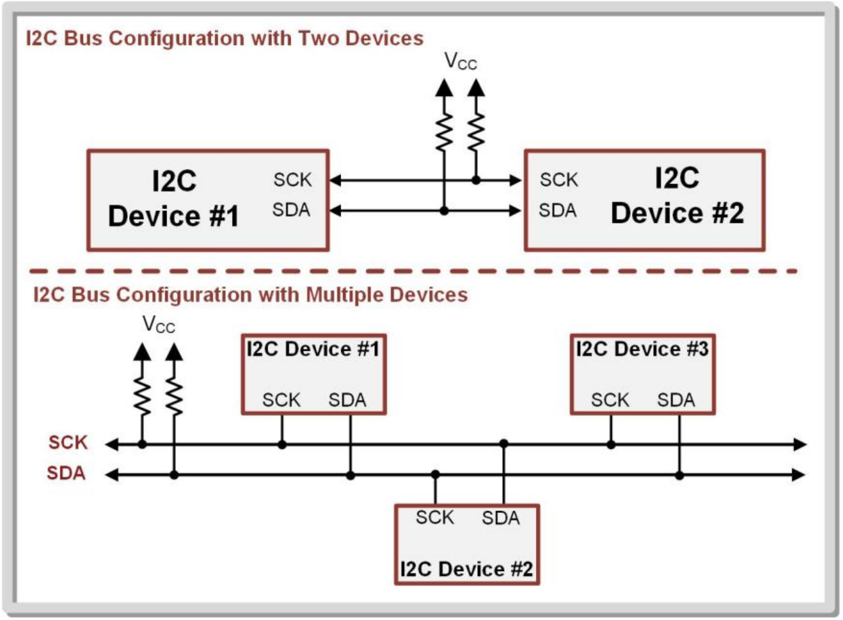
14.3.1 THE I2C PROTOCOL

- An I2C bus contains a clock line (SCK) and data line (SDA).



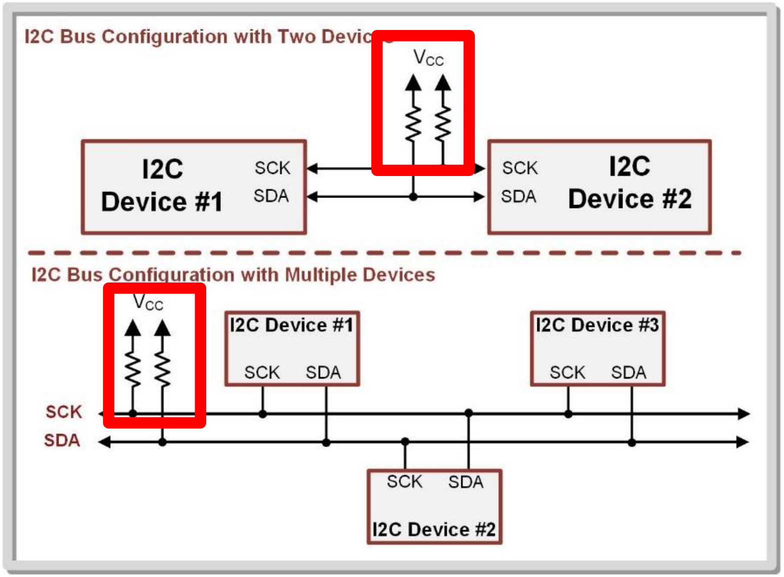
14.3.1 The I2C Protocol

- An I2C link is always **half-duplex**, meaning that all devices share the data line with only one device transmitting at any given time.



14.3.1 THE I2C PROTOCOL

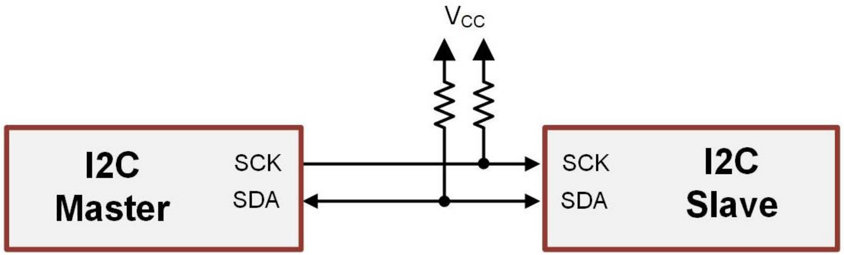
- An I2C bus **ALWAYS** needs external pull-up resistors on each of its lines.
- These can't be implemented on the MCU because the resistor function is only available when the port is configured as an INPUT.
- Since I2C is bidirectional, we never know wither the pin will be an input or output.



14.3.1 The I2C Protocol

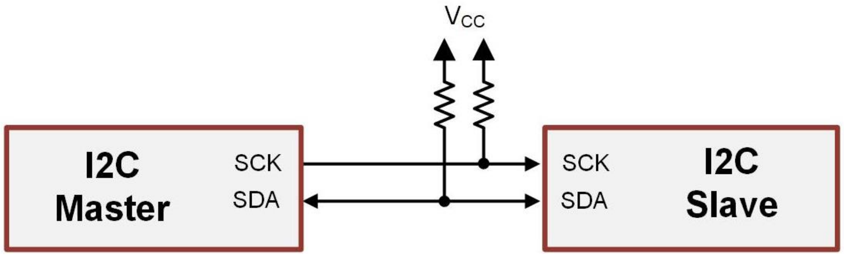
- **Master** – the device that initiates communication and controls the clock.

- Multiple masters are also supported on an I2C bus.



14.3.1 THE I2C PROTOCOL

- **Slave** – a device on the bus that is read or written to, but does not initiate transmission or provide a clock.
- **Slave address** – a unique and predetermined address for each slave on the bus.
- This address is used by the master to indicate which slave it wants to communicate with.

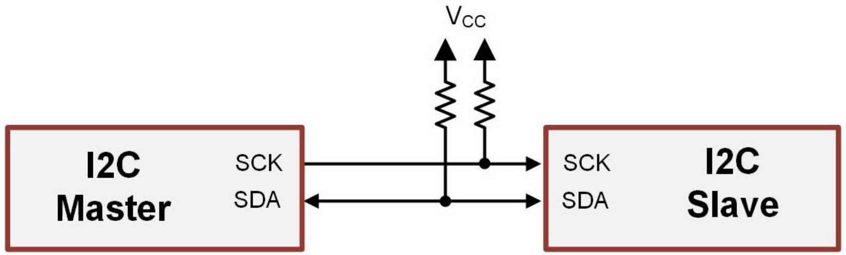


14.3.1 THE I2C PROTOCOL

- **Idle** – when both SDA and SCL are held high by the pull-up resistors and no I2C device is attempting to communicate.

- **Busy** – when devices are driving the bus.

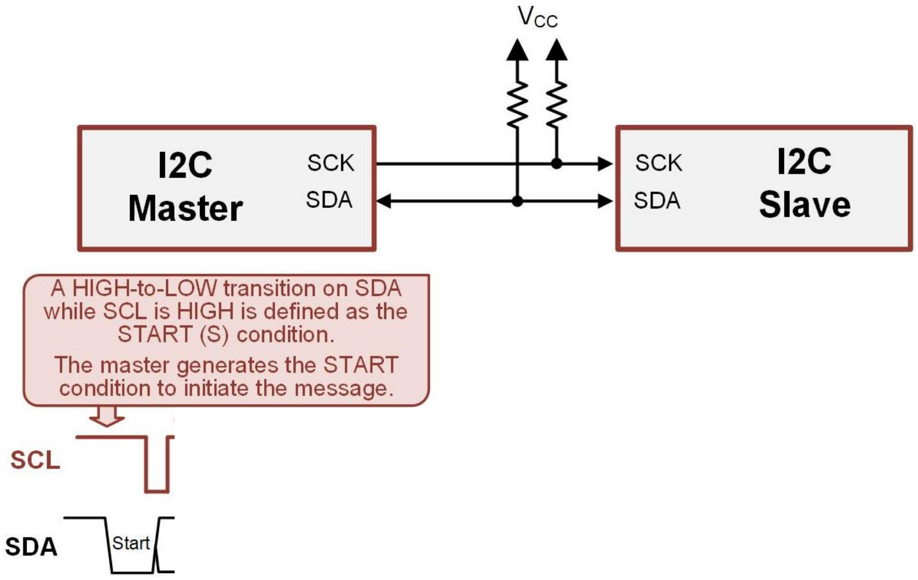
- **Messages** – how I2C information is transferred.



14.3.1 THE I2C PROTOCOL

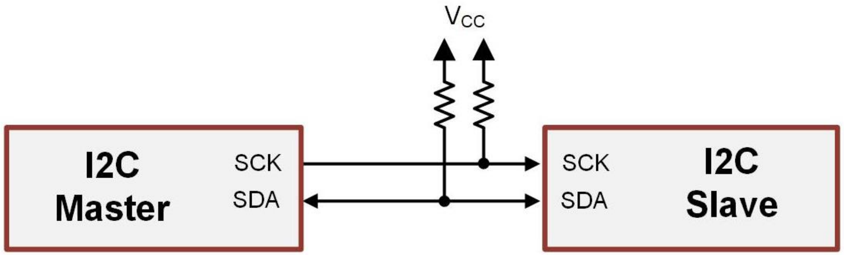
- A master initiates a new message by generating a START (S) condition by pulling SDA LOW while SCL is still HIGH.

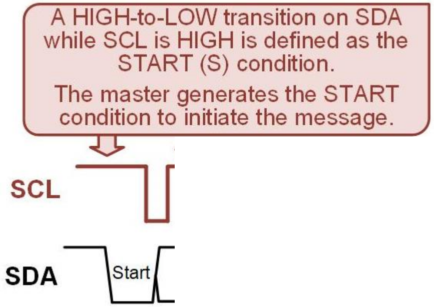
- As soon as the START condition is generated, the SCL will be pulled LOW and start pulsing to provide the clock for the message.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
SCL
SDA
Start

14.3.1 THE I2C PROTOCOL

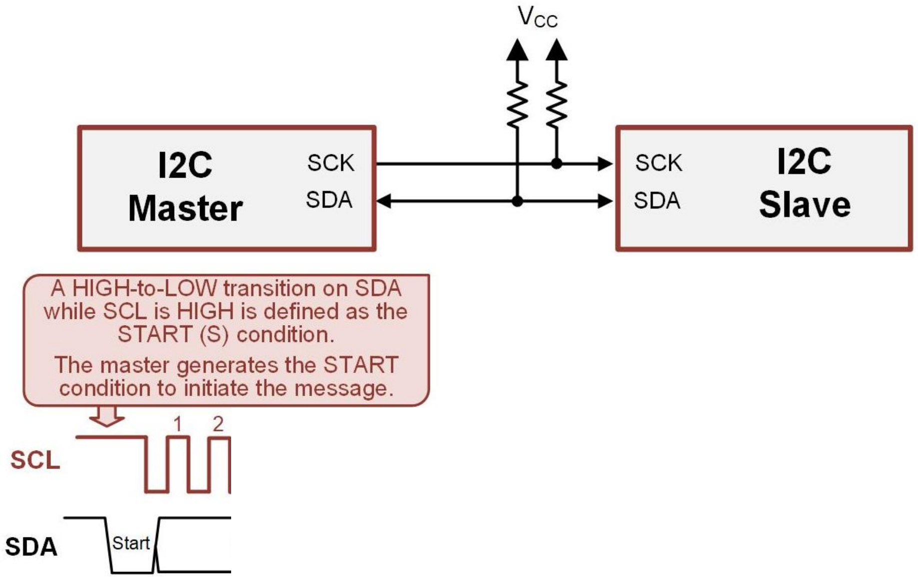
- The master is responsible for pulsing the clock.




A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
SCL
SDA
Start

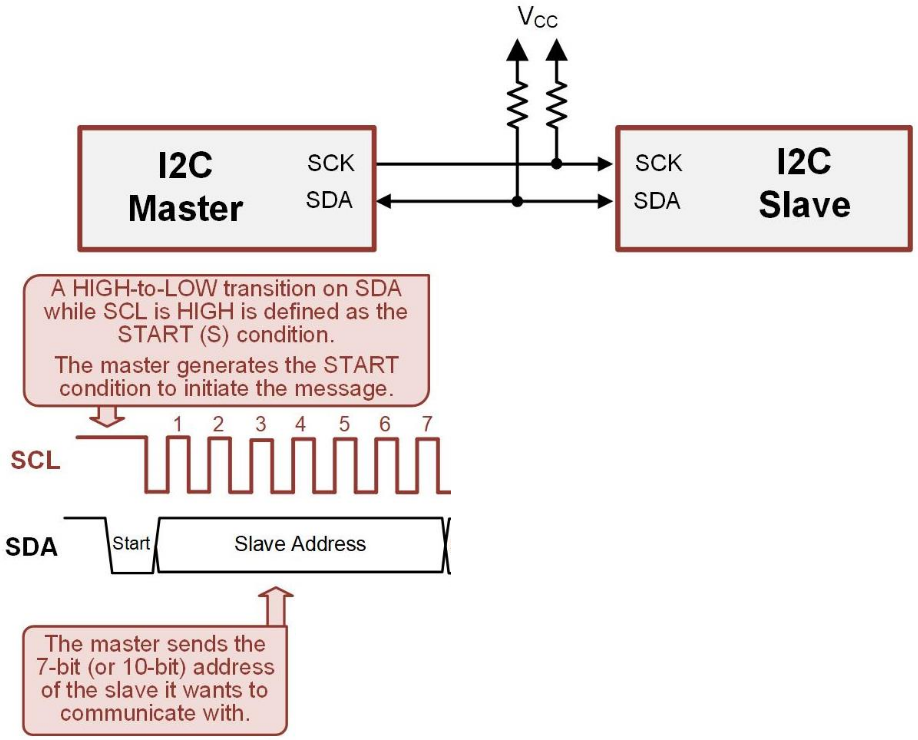
14.3.1 THE I2C PROTOCOL

- Each clock pulse within the I2C message is numbered by **periods**.
- Both the master and the slaves count the number of periods that have occurred since the message started in order to know when certain frames and signals should be present.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1
2
SCL
SDA
Start

14.3.1 THE I2C PROTOCOL

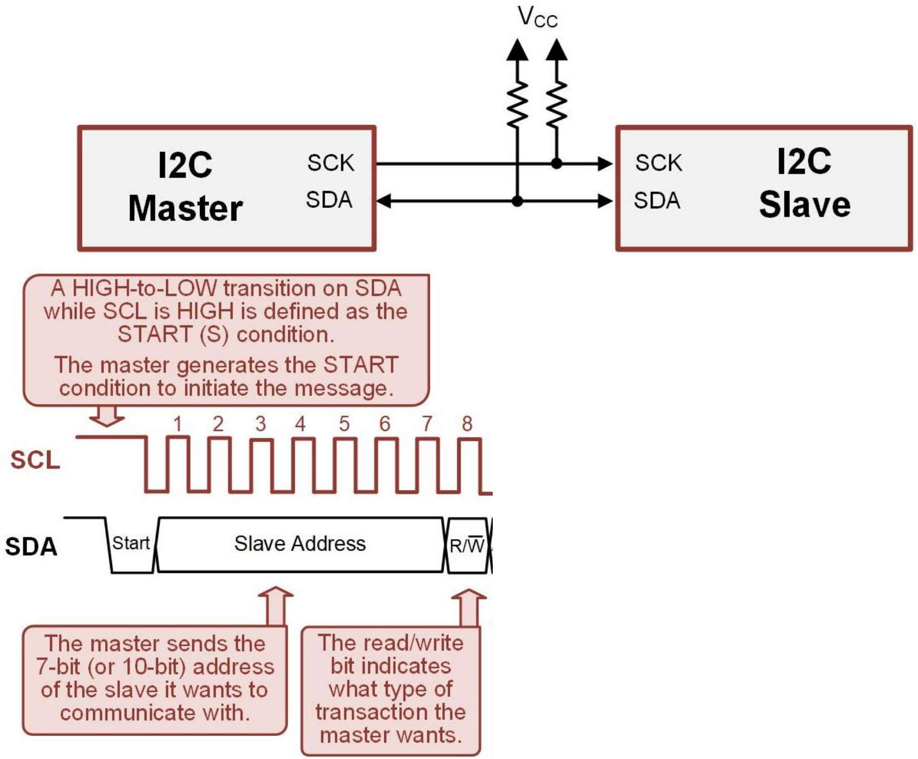
- After the master generates the START condition, it first sends the slave address that it wishes to communicate with.
- I2C slave addresses can either be 7-bit (default) or 10-bit.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7
SCL
SDA
Start
Slave Address
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.

14.3.1 THE I2C PROTOCOL

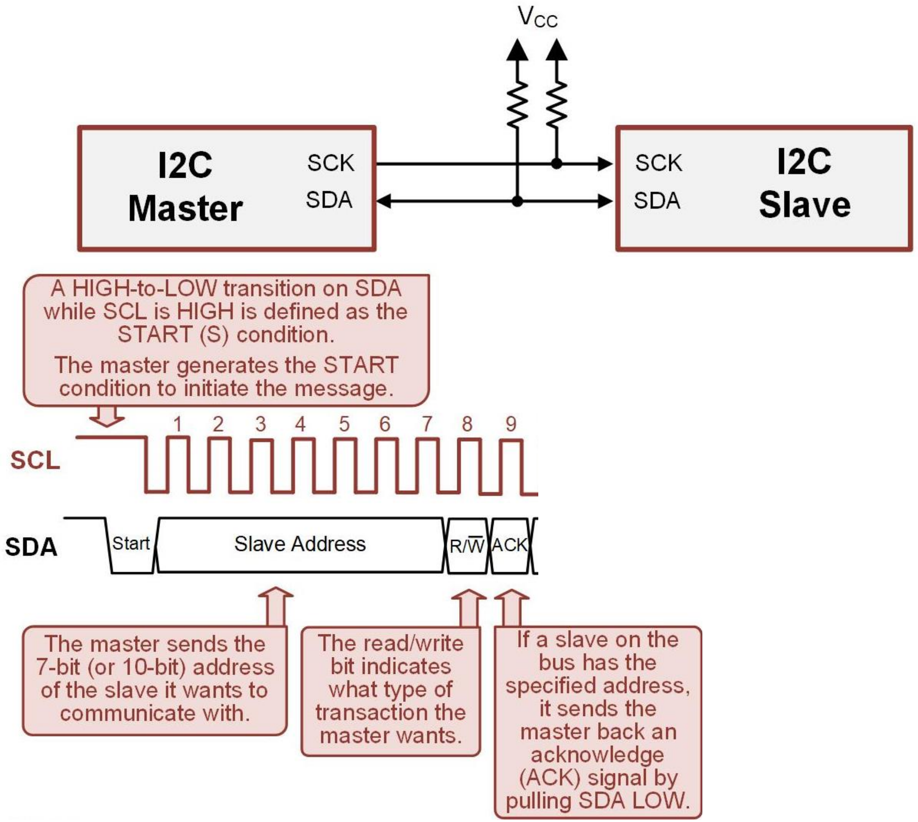
- The slave address is followed by the read/write signal indicating which type of transaction is being requested in the message.

- The START condition, slave address, and read/write signal constitute periods 1 → 8.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8
SCL
SDA
Start
Slave Address
R/$\overline{W}$
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.

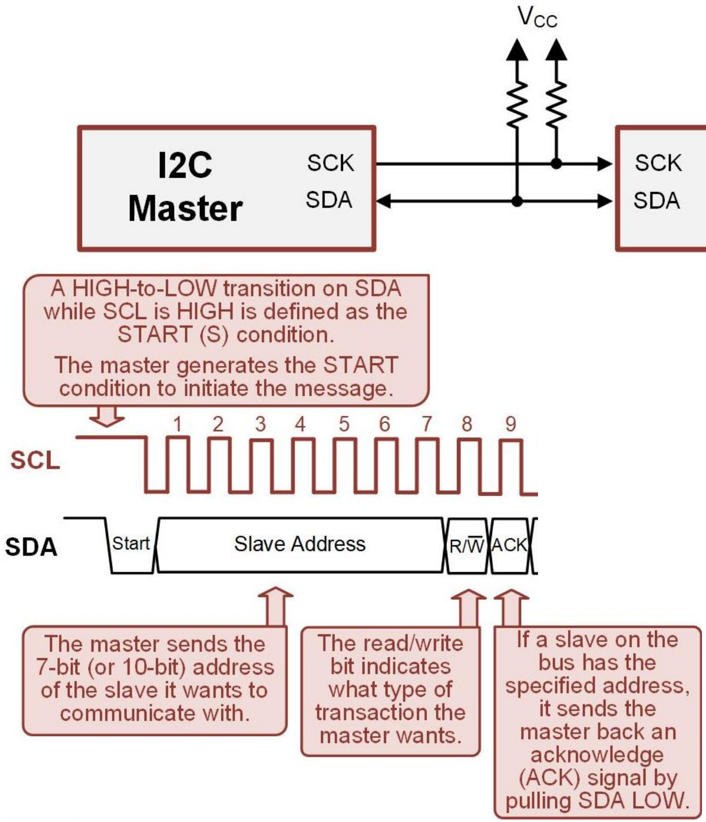
14.3.1 THE I2C PROTOCOL

- Period 9 of the message is reserved for the slave acknowledge (ACK) or no-acknowledge (NACK) signal.
- After the slave address and read/write signal are sent by the master, each slave on the bus checks whether it is being addressed.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8 9
SCL
SDA
Start
Slave Address
R/$\overline{W}$
ACK
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.

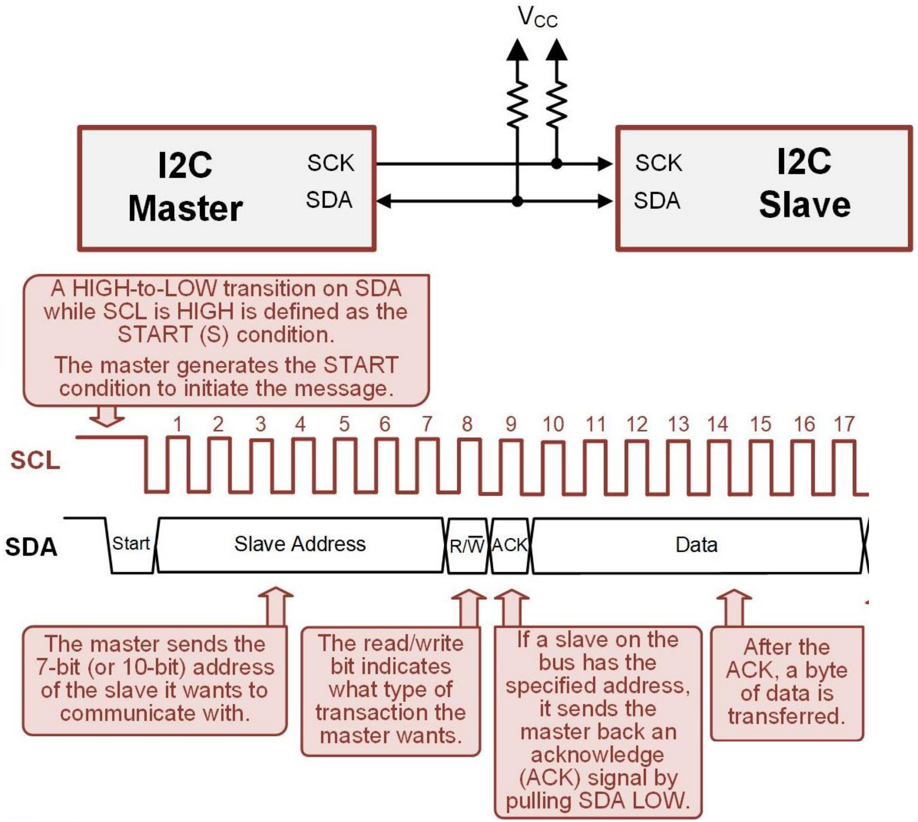
14.3.1 THE I2C PROTOCOL

- If a slave exists with the specified slave address, it will send an **ACK** signal back to the master by pulling SDA LOW.
- If no device exists with the specified slave address, no device will pull down SDA. This will resul in period 9 remaining HIGH an will be interpreted as a **NACK**.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8 9
SCL
SDA
Start
Slave Address
R/$\overline{W}$
ACK
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.

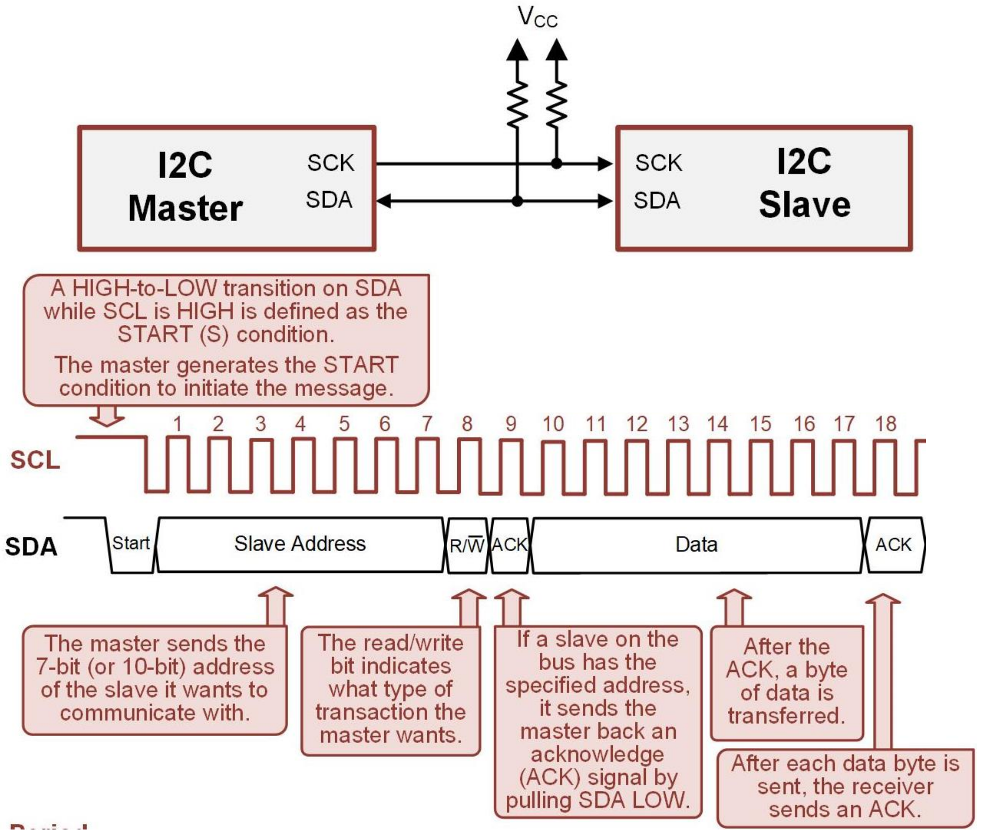
14.3.1 The I2C Protocol

- If the master sees the ACK signal, it knows a slave exists with the specified address and proceeds with the message.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
SCL
SDA
Start
Slave Address
R/$\overline{W}$
ACK
Data
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.
After the ACK, a byte of data is transferred.

14.3.1 The I2C Protocol

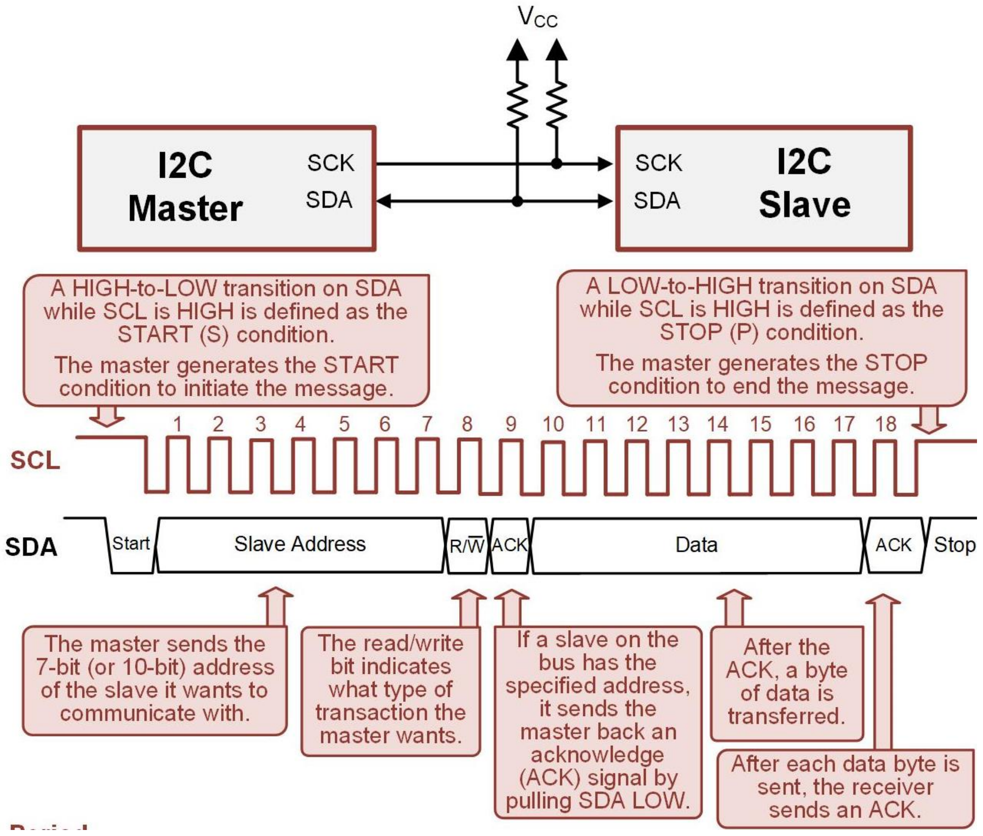
- After each byte is sent, the receiving device sends an ACK signal indicating that it successfully received the data.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
SCL
SDA
Start
Slave Address
R/$\overline{W}$
ACK
Data
ACK
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.
After the ACK, a byte of data is transferred.
After each data byte is sent, the receiver sends an ACK.

14.3.1 The I2C Protocol

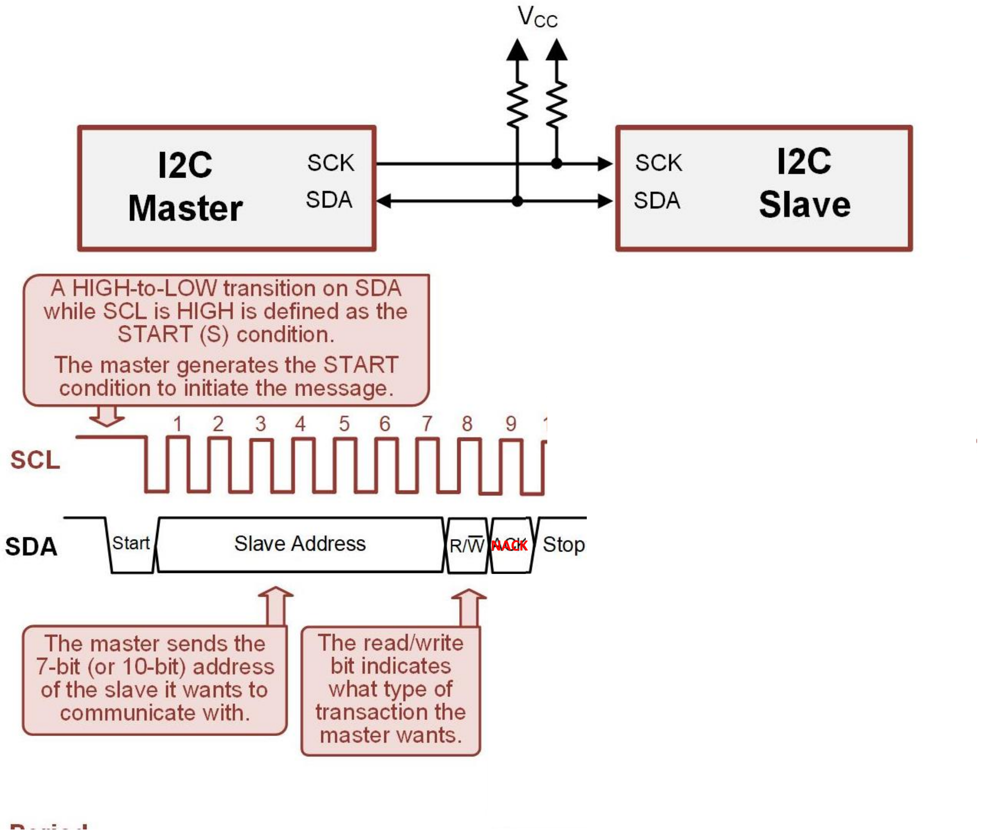
- A STOP condition occurs when there is a LOW-to-HIGH transition on SDA while SCL is HIGH.

- Once, SDA goes HIGH, SCL also remains HIGH indicating that the bus is idle again.

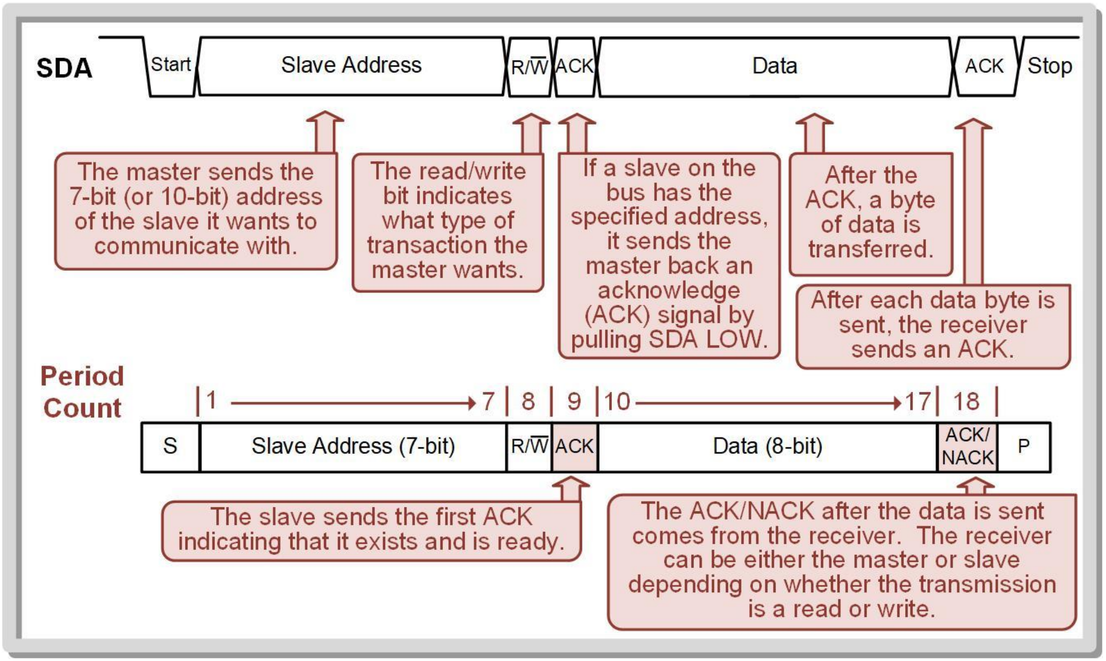

Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
A LOW-to-HIGH transition on SDA while SCL is HIGH is defined as the STOP (P) condition.
The master generates the STOP condition to end the message.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
SCL
SDA
Start
Slave Address
R/W̄
ACK
Data
ACK
Stop
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.
After the ACK, a byte of data is transferred.
After each data byte is sent, the receiver sends an ACK.

14.3.1 THE I2C PROTOCOL

- A NACK in period 9 tells the master that no slave exists with the specified address.
- The master then generates a STOP condition and ends the message.


Vcc
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
A HIGH-to-LOW transition on SDA while SCL is HIGH is defined as the START (S) condition.
The master generates the START condition to initiate the message.
1 2 3 4 5 6 7 8 9
SCL
SDA
Start
Slave Address
R/W
NACK
Stop
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.

14.3.1 THE I2C PROTOCOL


SDA
Start
Slave Address
R/$\overline{W}$
ACK
Data
ACK
Stop
The master sends the 7-bit (or 10-bit) address of the slave it wants to communicate with.
The read/write bit indicates what type of transaction the master wants.
If a slave on the bus has the specified address, it sends the master back an acknowledge (ACK) signal by pulling SDA LOW.
After the ACK, a byte of data is transferred.
After each data byte is sent, the receiver sends an ACK.
Period Count
1
7
8
9
10
17
18
S
Slave Address (7-bit)
R/$\overline{W}$
ACK
Data (8-bit)
ACK/ NACK
P
The slave sends the first ACK indicating that it exists and is ready.
The ACK/NACK after the data is sent comes from the receiver. The receiver can be either the master or slave depending on whether the transmission is a read or write.

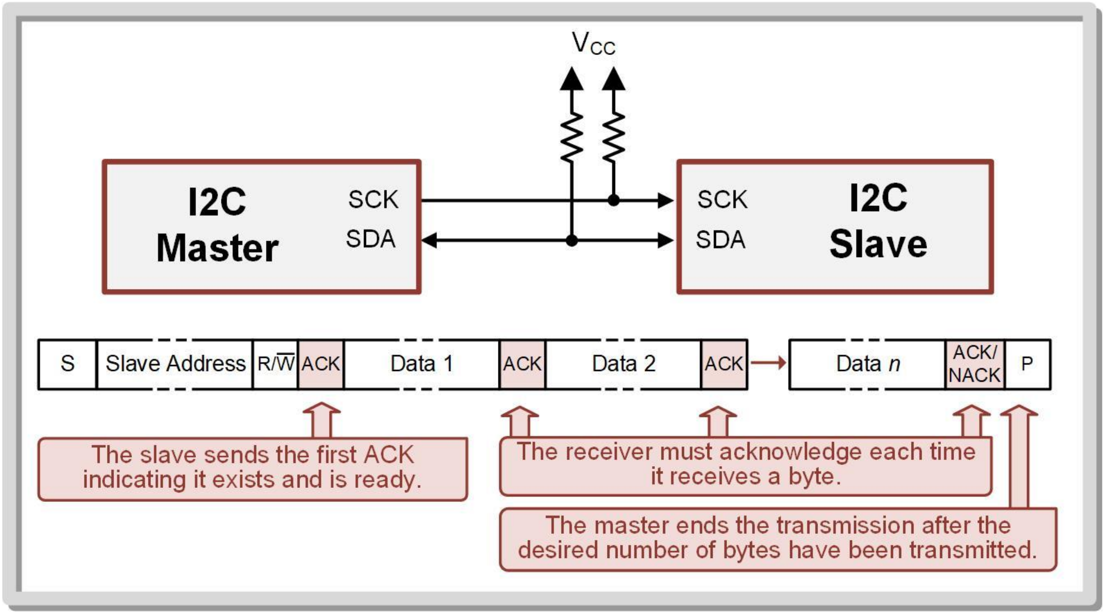
14.3.1 THE I2C PROTOCOL

- When the master is **writing** to a slave, the master sends the 8-bits of data and the slave produces the ACK/NACK signal.
- When the master is **reading** from a slave, the slave sends the 8-bits of data and the master produces the ACK/NACK signal.
- After the data has been sent and acknowledged, the master can end the message by generating the STOP condition anytime.

Image Courtesy of https://neodem.wp.horizon.ac.uk/

14.3.1 THE I2C PROTOCOL

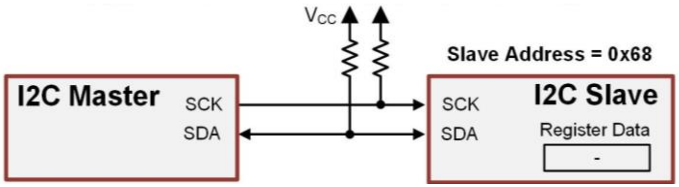
- The Master can send multiple data bytes in a single message.


VCC
I2C Master
SCK
SDA
SCK
SDA
I2C Slave
S
Slave Address
R/W̄
ACK
Data 1
ACK
Data 2
ACK
Data n
ACK/ NACK
P
The slave sends the first ACK indicating it exists and is ready.
The receiver must acknowledge each time it receives a byte.
The master ends the transmission after the desired number of bytes have been transmitted.

14.3.1 The I2C Protocol

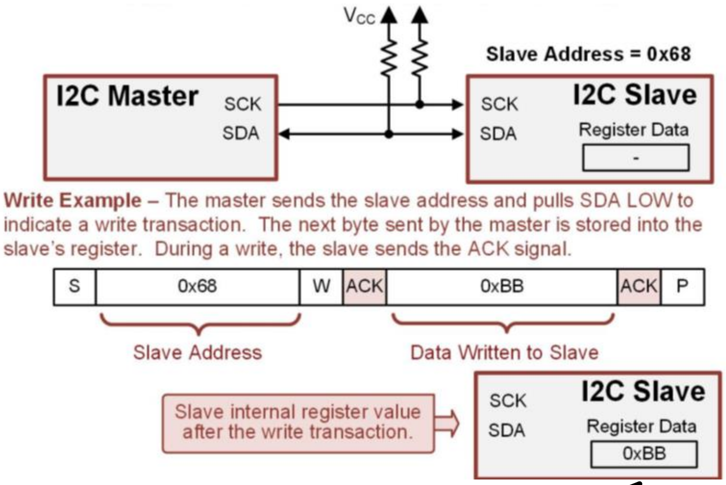
- I2C devices contain individual registers that hold their information.

- In the simplest case, an I2C device contains only a single register.



14.3.1 The I2C Protocol

- **Writing** to a slave with a single register.
  - The master initiates the message by sending the S bit.
  - The master provides the slave address and a WRITE signal.
  - The slave ACKs to indicate it is ready.
  - The master writes the data.
  - The slave sends an **ACK** that it got the data.
  - The master ends the message by sending the P bit.



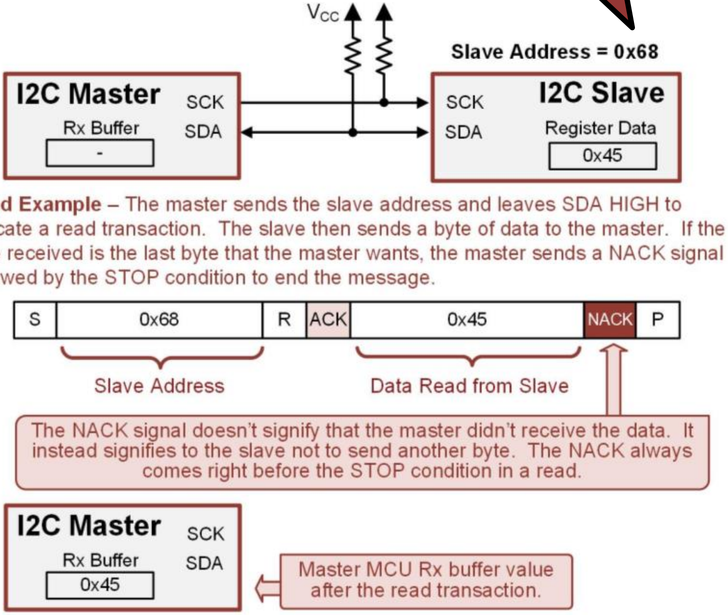
Write Example – The master sends the slave address and pulls SDA LOW to indicate a write transaction. The next byte sent by the master is stored into the slave's register. During a write, the slave sends the ACK signal.

No register address is needed.

14.3.1 THE I2C PROTOCOL

- **Reading** from a slave with a single register.
  - The master initiates the message by sending the S bit.
  - The master provides the slave address and a WRITE signal.
  - The slave ACKs to indicate it is ready.
  - The slave sends the data.
  - The master sends a **NACK** to stop the slave from sending more data.
  - The master ends the message by sending the P bit.

No register address is needed.

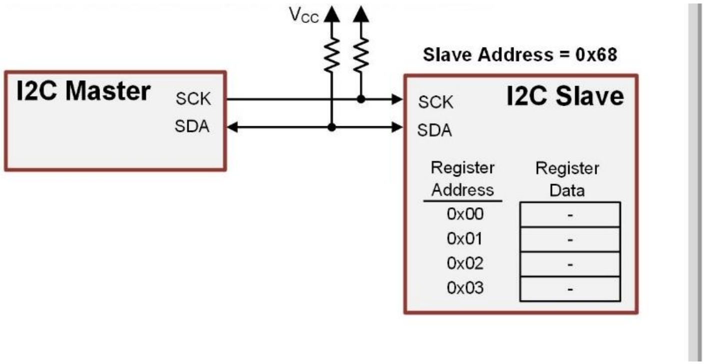


Vcc
Slave Address = 0x68
I2C Master
SCK
SDA
Rx Buffer
-
I2C Slave
Register Data
0x45
S
0x68
R
ACK
0x45
NACK
P
Slave Address
Data Read from Slave
The NACK signal doesn't signify that the master didn't receive the data. It instead signifies to the slave not to send another byte. The NACK always comes right before the STOP condition in a read.
I2C Master
SCK
SDA
Rx Buffer
0x45
Master MCU Rx buffer value after the read transaction.

Read Example – The master sends the slave address and leaves SDA HIGH to indicate a read transaction. The slave then sends a byte of data to the master. If the byte received is the last byte that the master wants, the master sends a NACK signal followed by the STOP condition to end the message.

14.3.1 THE I2C PROTOCOL

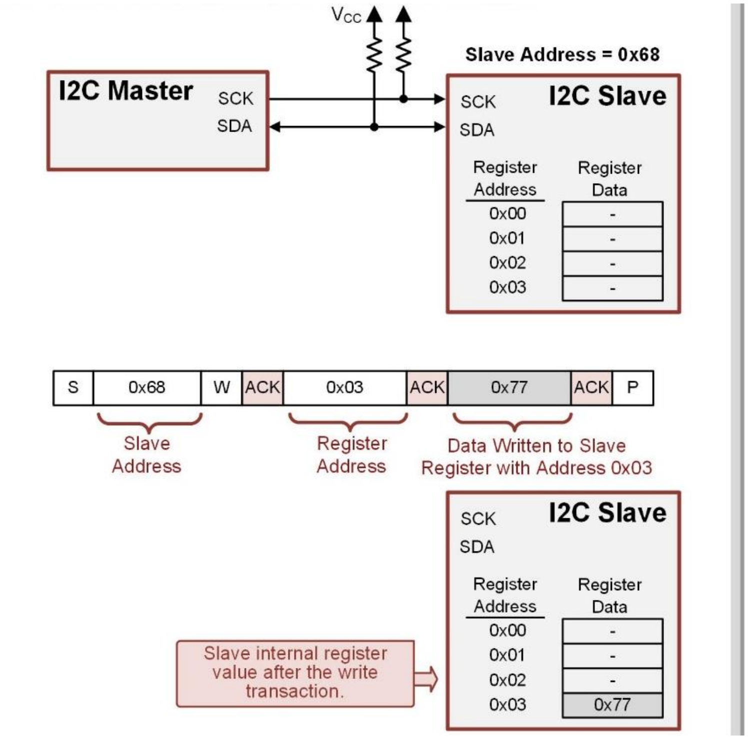
- When an I2C device has multiple registers, each is assigned a **register address**.
- To access a specific register in the slave, the master needs to provide the register address in the message.



14.3.1 The I2C Protocol

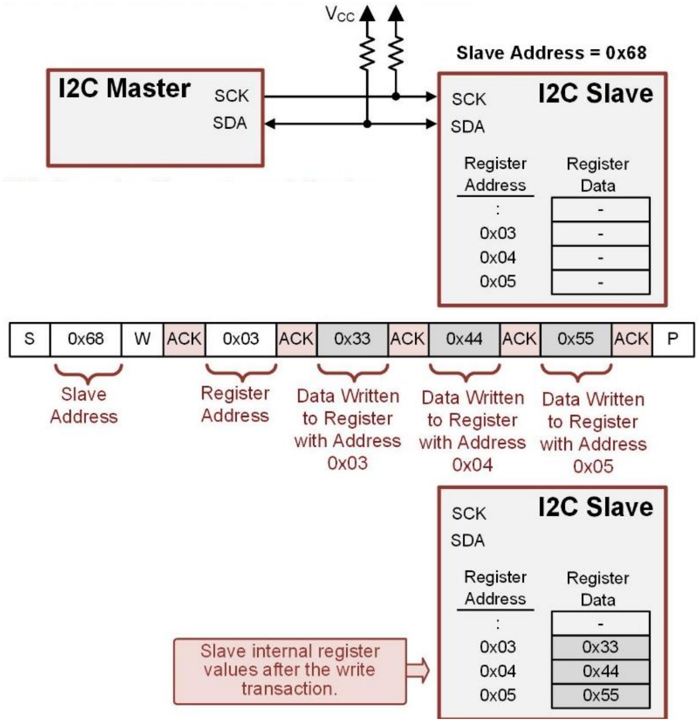
- For a **write** transaction, the master generates an I2C message that first provides the slave address, then provides the register address to access, then provides the data to be written.

- After each frame, the slave sends an ACK signal.



14.3.1 THE I2C PROTOCOL

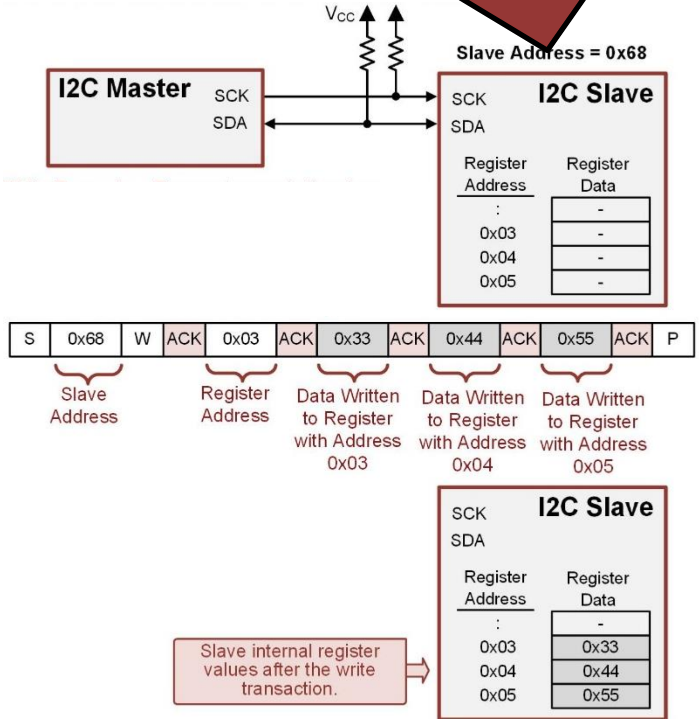
- The master can also **write** a block of data to registers.
- The slave automatically increments the starting address after each byte of data is written.
- The master still sends the slave address, the write signal, and the starting register address.
- The next byte of data that is sent goes into the first register address location.


Vcc
Slave Address = 0x68
I2C Master
SCK
SDA
I2C Slave
SCK
SDA
Register Address
Register Data
:
0x03
0x04
0x05
-
S
0x68
W
ACK
0x03
0x33
0x44
0x55
P
Slave Address
Register Address
Data Written to Register with Address 0x03
Data Written to Register with Address 0x04
Data Written to Register with Address 0x05
Slave internal register values after the write transaction.

14.3.1 THE I2C PROTOCOL

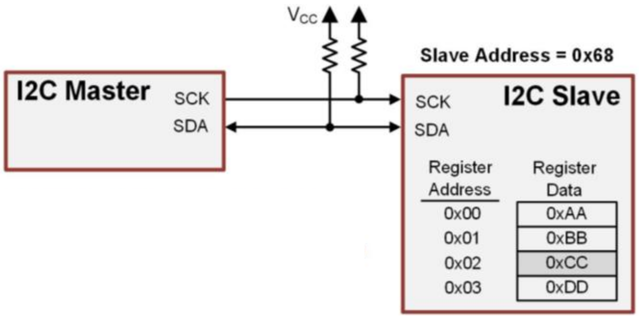
- The slave will continue to increment its register address until it sees the STOP condition generated by the master.

Check the slave's datasheet for details!!!


Vcc
Slave Address = 0x68
I2C Master
SCK
SDA
I2C Slave
SCK
SDA
Register Address
Register Data
:
0x03
0x04
0x05
-
S
0x68
W
ACK
0x03
0x33
0x44
0x55
P
Slave Address
Register Address
Data Written to Register with Address 0x03
Data Written to Register with Address 0x04
Data Written to Register with Address 0x05
Slave internal register values after the write transaction.
0x33
0x44
0x55

14.3.1 THE I2C PROTOCOL

- When **reading** from a device that contains multiple registers, two message are needed.

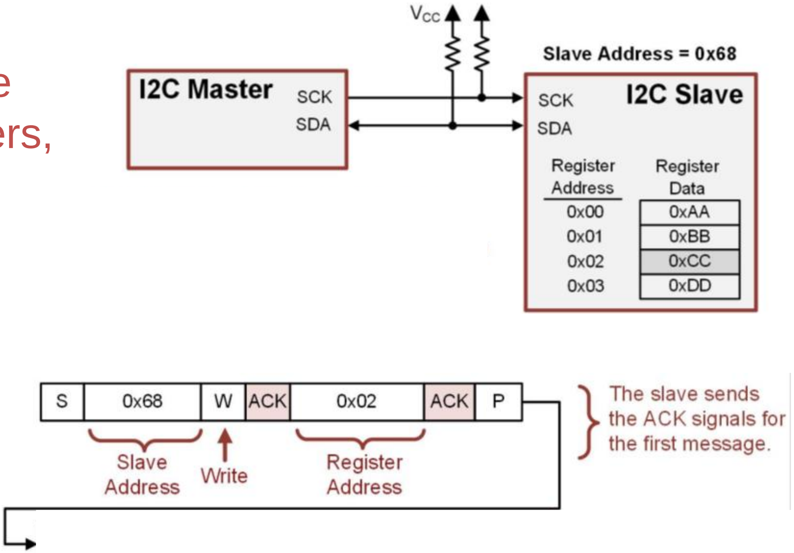


14.3.1 The I2C Protocol

- When **reading** from a device that contains multiple registers, two message are needed.

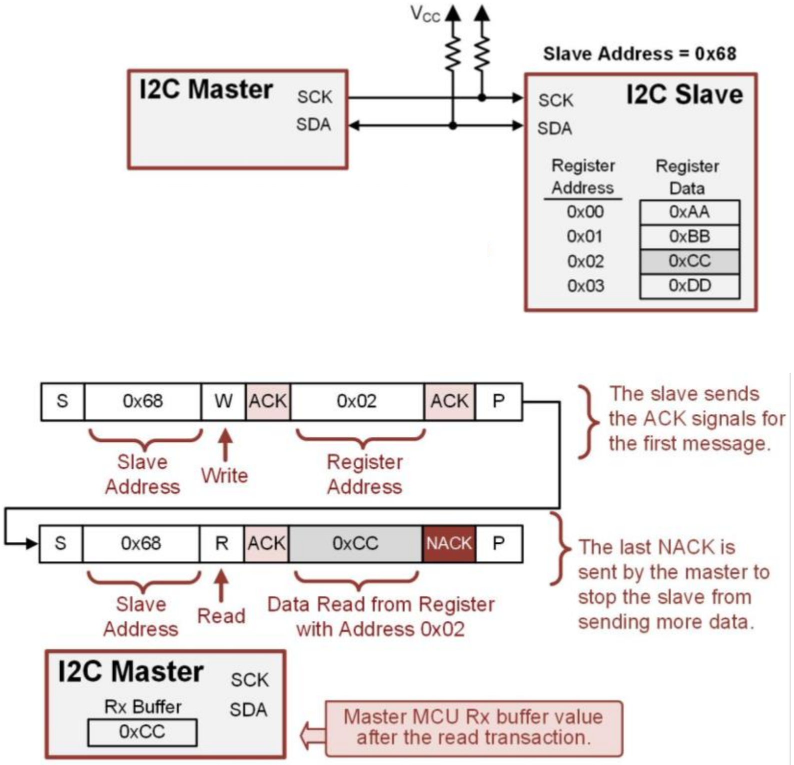
- The first message sets the register address within the slave that will be read from using a write transaction.

- The first message puts the slave into a mode where it's expecting a second message that will read data from the register address that was just sent.



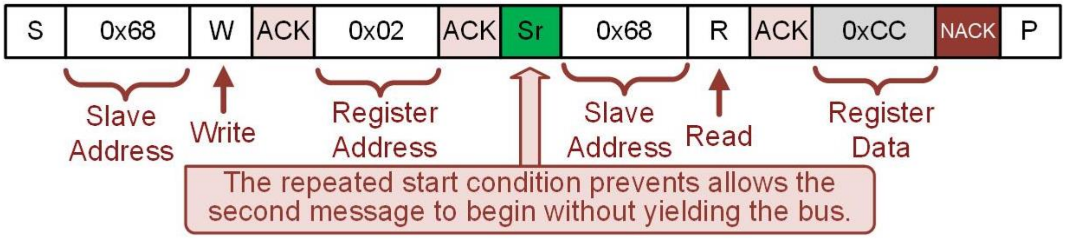
14.3.1 THE I2C PROTOCOL

- The second message performs a read transaction that retrieves the data from the register address sent in the first message.



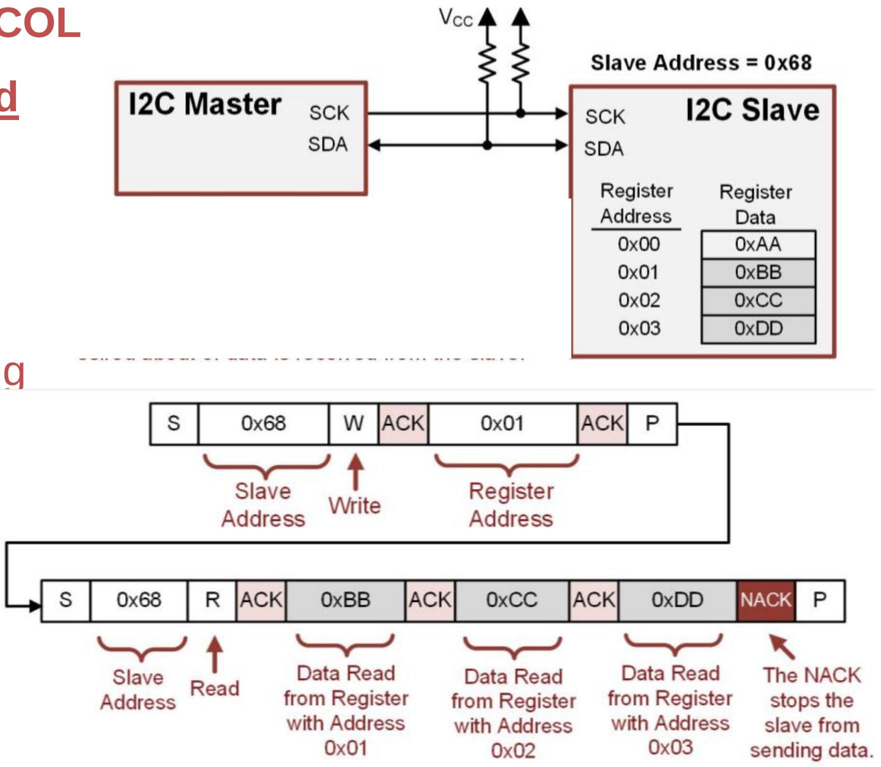
14.3.1 THE I2C PROTOCOL

- An alternative approach can also be used where a second START (Sr) condition is generated before the STOP condition of the first message.
- This initiates the second read transaction immediately without giving up control of the bus.



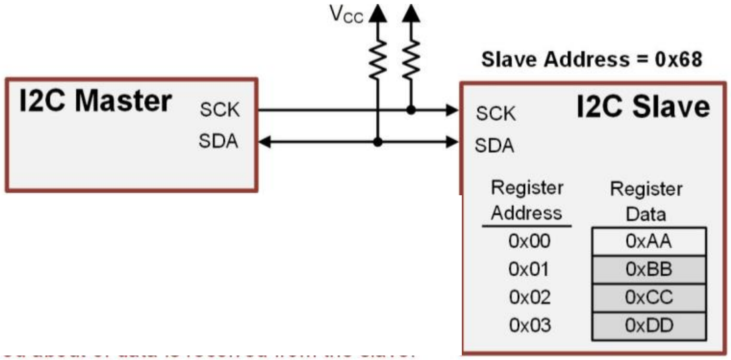
14.3.1 THE I2C PROTOCOL

- The master can also <u>**read**</u> blocks of data from a slave.
- In this situation, the master still sends two messages, the first setting the register address and the second retrieving the data.
- In the second message, the master continually sends SCL pulses to rea as many bytes as it want prior to generating the STOP condition.

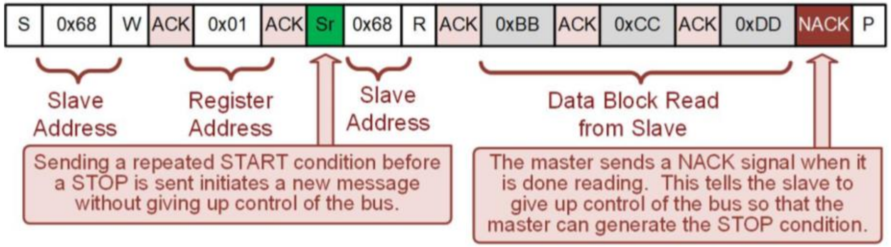


14.3.1 THE I2C PROTOCOL

- A repeated Start bit (Sr) can also be used to read blocks of information.



Register Address	Register Data
0x00	0xAA
0x01	0xBB
0x02	0xCC
0x03	0xDD



Sending a repeated START condition before a STOP is sent initiates a new message without giving up control of the bus.

The master sends a NACK signal when it is done reading. This tells the slave to give up control of the bus so that the master can generate the STOP condition.