Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire Library

This library allows you to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21.

As a reference the table below shows where TWI pins are located on various Arduino boards.

Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

As of Arduino 1.0, the library inherits from the Stream functions, making it consistent with other read/write libraries. Because of this, send() and receive() have been replaced with read() and write().

Note

There are both 7- and 8-bit versions of I2C addresses. 7 bits identify the device, and the eighth bit determines if it's being written to or read from. The Wire library uses 7 bit addresses throughout. If you have a datasheet or sample code that uses 8 bit address, you'll want to drop the low bit (i.e. shift the value one bit to the right), yielding an address between 0 and 127. However the addresses from 0 to 7 are not used because are reserved so the first address that can be used is 8. Please note that a pull-up resistor is needed when connecting SDA/SCL pins. Please refer to the examples for more informations. MEGA 2560 board has pull-up resistors on pins 20 - 21 onboard.

The Wire library implementation uses a 32 byte buffer, therefore any communication should be within this limit. Exceeding bytes in a single transmission will just be dropped.

Functions

- begin (//www.arduino.cc/en/Reference/WireBegin)()
- requestFrom (//www.arduino.cc/en/Reference/WireRequestFrom)()
- beginTransmission (//www.arduino.cc/en/Reference/WireBeginTransmission)()
- endTransmission (//www.arduino.cc/en/Reference/WireEndTransmission)()
- write (//www.arduino.cc/en/Reference/WireWrite)()
- available (//www.arduino.cc/en/Reference/WireAvailable)()
- read (//www.arduino.cc/en/Reference/WireRead)()
- SetClock (//www.arduino.cc/en/Reference/WireSetClock)()
- onReceive (//www.arduino.cc/en/Reference/WireOnReceive)()
- onRequest (//www.arduino.cc/en/Reference/WireOnRequest)()

Examples

- Digital Potentiometer (/en/Tutorial/DigitalPotentiometer): Control an Analog Devices AD5171 Digital Potentiometer.
- Master Reader/Slave Writer (/en/Tutorial/MasterReader): Program two Arduino boards to communicate with one another in a Master Reader/Slave Sender configuration via the I2C.
- Master Writer/Slave receiver (/en/Tutorial/MasterWriter):Program two Arduino boards to communicate with one another in a Master Writer/Slave Receiver configuration via the I2C.
- SFR Ranger Reader (/en/Tutorial/SFRRangerReader): Read an ultra-sonic range finder interfaced via the I2C.
- Add SerCom (/en/Tutorial/SamdSercom): Adding mores Serial interfaces to SAMD microcontrollers.

See also

- Master Writer (http://arduino.cc/en/Tutorial/MasterWriter)
- Master Reader (http://arduino.cc/en/Tutorial/MasterReader)
- SFR Ranger Reader (http://arduino.cc/en/Tutorial/SFRRangerReader)
- Digital Potentiometer (http://arduino.cc/en/Tutorial/DigitalPotentiometer)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

The text of the Arduino reference is licensed under a Creative Commons Attribution-ShareAlike 3.0 License (http://creativecommons.org/licenses/by-sa/3.0/). Code samples in the reference are released into the public domain.

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.begin()

Wire.begin(address)

Description

Initiate the Wire library and join the I2C bus as a master or slave. This should normally be called only once.

Parameters

address: the 7-bit slave address (optional); if not specified, join the bus as a master.

Returns

None

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

The text of the Arduino reference is licensed under a Creative Commons Attribution-ShareAlike 3.0 License (http://creativecommons.org/licenses/by-sa/3.0/). Code samples in the reference are released into the public domain.

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.requestFrom()

Description

Used by the master to request bytes from a slave device. The bytes may then be retrieved with the available() (//www.arduino.cc/en/Reference/WireAvailable) and read() (//www.arduino.cc/en/Reference/WireRead) functions.

As of Arduino 1.0.1, requestFrom() accepts a boolean argument changing its behavior for compatibility with certain I2C devices.

If true, requestFrom() sends a stop message after the request, releasing the I2C bus.

If false, requestFrom() sends a restart message after the request. The bus will not be released, which prevents another master device from requesting between messages. This allows one master device to send multiple requests while in control.

The default value is true.

Syntax

Wire.requestFrom(address, quantity)
Wire.requestFrom(address, quantity, stop)

Parameters

address: the 7-bit address of the device to request bytes from

quantity: the number of bytes to request

stop : boolean. true will send a stop message after the request, releasing the bus. false will continually send a restart after the request, keeping the connection active.

Returns

byte : the number of bytes returned from the slave device

See Also

- Wire.available() (//www.arduino.cc/en/Reference/WireAvailable)
- Wire.read() (//www.arduino.cc/en/Reference/WireRead)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference   Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.endTransmission()

Description

Ends a transmission to a slave device that was begun by beginTransmission() (//www.arduino.cc/en/Reference/WireBeginTransmission) and transmits the bytes that were queued by write() (//www.arduino.cc/en/Reference/WireWrite).

As of Arduino 1.0.1, endTransmission() accepts a boolean argument changing its behavior for compatibility with certain I2C devices.

If true, endTransmission() sends a stop message after transmission, releasing the I2C bus.

If false, endTransmission() sends a restart message after transmission. The bus will not be released, which prevents another master device from transmitting between messages. This allows one master device to send multiple transmissions while in control.

The default value is true.

Syntax

Wire.endTransmission()
Wire.endTransmission(stop)

Parameters

stop : boolean. true will send a stop message, releasing the bus after transmission. false will send a restart, keeping the connection active.

Returns

byte, which indicates the status of the transmission:

Reference   Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.beginTransmission(address)

Description

Begin a transmission to the I2C slave device with the given address. Subsequently, queue bytes for transmission with the write() (//www.arduino.cc/en/Reference/WireWrite) function and transmit them by calling endTransmission() (//www.arduino.cc/en/Reference/WireEndTransmission).

Parameters

address: the 7-bit address of the device to transmit to

Returns

None

See Also

- Wire.write() (//www.arduino.cc/en/Reference/WireWrite)
- Wire.endTransmission() (//www.arduino.cc/en/Reference/WireEndTransmission)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

- 0:success
- 1:data too long to fit in transmit buffer
- 2:received NACK on transmit of address
- 3:received NACK on transmit of data
- 4:other error

See Also

- Wire.beginTransmission() (//www.arduino.cc/en/Reference/WireBeginTransmission)
- Wire.write() (//www.arduino.cc/en/Reference/WireWrite)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire (//www.arduino.cc/en/Reference/Wire)

write()

Description

Writes data from a slave device in response to a request from a master, or queues bytes for transmission from a master to slave device (in-between calls to beginTransmission() and endTransmission()).

Syntax

Wire.write(value)
Wire.write(string)
Wire.write(data, length)

Parameters

value: a value to send as a single byte

string: a string to send as a series of bytes

data: an array of data to send as bytes

length: the number of bytes to transmit

Returns

byte: write() will return the number of bytes written, though reading that number is optional

Example

```
#include <Wire.h>

byte val = 0;

void setup()
{
  Wire.begin(); // join i2c bus
}

void loop()
{
  Wire.beginTransmission(44); // transmit to device #44 (0x2c)
                              // device address is specified in datasheet
  Wire.write(val);            // sends value byte
  Wire.endTransmission();     // stop transmitting

  val++;        // increment value
  if(val == 64) // if reached 64th position (max)
  {
    val = 0;    // start over from lowest value
  }
  delay(500);
}
```

[Get Code] (//www.arduino.cc/en/Reference/WireWrite?action=sourceblock&num=1)

See also

- WireRead (//www.arduino.cc/en/Reference/WireRead)()
- WireBeginTransmission (//www.arduino.cc/en/Reference/WireBeginTransmission)()
- WireEndTransmission (//www.arduino.cc/en/Reference/WireEndTransmission)()
- Serial.write (//www.arduino.cc/en/Serial/Write)()

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.available()

Description

Returns the number of bytes available for retrieval with read() (//www.arduino.cc/en/Reference/WireRead). This should be called on a master device after a call to requestFrom() (//www.arduino.cc/en/Reference/WireRequestFrom) or on a slave inside the onReceive() (//www.arduino.cc/en/Reference/WireOnReceive) handler.

available() inherits from the Stream (//www.arduino.cc/en/Reference/Stream) utility class.

Parameters

None

Returns

The number of bytes available for reading.

See Also

- Wire.read() (//www.arduino.cc/en/Reference/WireRead)
- Stream.available (//www.arduino.cc/en/Reference/StreamAvailable)()

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire (//www.arduino.cc/en/Reference/Wire)

read()

Description

Reads a byte that was transmitted from a slave device to a master after a call to requestFrom() (//www.arduino.cc/en/Reference/WireRequestFrom) or was transmitted from a master to a slave. read() inherits from the Stream (//www.arduino.cc/en/Reference/Stream) utility class.

Syntax

Wire.read()

Parameters

Returns

The next byte received

Example

```
#include <Wire.h>

void setup()
{
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
}

void loop()
```

```
{
  Wire.requestFrom(2, 6);    // request 6 bytes from slave device #2

  while(Wire.available())    // slave may send less than requested
  {
    char c = Wire.read();    // receive a byte as character
    Serial.print(c);         // print the character
  }

  delay(500);
}
```

See also

- WireWrite (//www.arduino.cc/en/Reference/WireWrite)()
- WireAvailable (//www.arduino.cc/en/Reference/WireAvailable)()
- WireRequestFrom (//www.arduino.cc/en/Reference/WireRequestFrom)()
- Stream.read (//www.arduino.cc/en/Reference/StreamRead)()

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.setClock()

Description

This function modifies the clock frequency for I2C communication. I2C slave devices have no minimum working clock frequency, however 100KHz is usually the baseline.

Syntax

Wire.setClock(clockFrequency)

Parameters

clockFrequency: the value (in Hertz) of desired communication clock. Accepted values are 100000 (standard mode) and 400000 (fast mode). Some processors also support 10000 (low speed mode), 1000000 (fast mode plus) and 3400000 (high speed mode). Please refer to the specific processor documentation to make sure the desired mode is supported.

Returns

None

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.onReceive(handler)

Description

Registers a function to be called when a slave device receives a transmission from a master.

Parameters

handler: the function to be called when the slave receives data; this should take a single int parameter (the number of bytes read from the master) and return nothing, e.g.:
```
void myHandler(int numBytes)
```

Returns

None

See Also

- Wire.onRequest() (//www.arduino.cc/en/Reference/WireOnRequest)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share

Reference Language (//www.arduino.cc/en/Reference/HomePage) | Libraries (//www.arduino.cc/en/Reference/Libraries) | Comparison (//www.arduino.cc/en/Reference/Comparison) | Changes (//www.arduino.cc/en/Reference/Changes)

Wire.onRequest(handler)

Description

Register a function to be called when a master requests data from this slave device.

Parameters

handler: the function to be called, takes no parameters and returns nothing, e.g.:
```
void myHandler()
```

Returns

None

See Also

- Wire.onReceive() (//www.arduino.cc/en/Reference/WireOnReceive)

Reference Home (//www.arduino.cc/en/Reference/HomePage)

Corrections, suggestions, and new documentation should be posted to the Forum (http://arduino.cc/forum/index.php/board,23.0.html).

Share