Wire - slave_receiver, slave_sender

Example Code

```
// Wire Slave Receiver
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Receives data as an I2C/TWI slave device
// Refer to the "Wire Master Writer" example for use with this

// Created 29 March 2006

// This example code is in the public domain.


#include <Wire.h>

void setup() {
  Wire.begin(8);                // join i2c bus with address #8
  Wire.onReceive(receiveEvent); // register event
  Serial.begin(9600);           // start serial for output
}

void loop() {
  delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany) {
  while (1 < Wire.available()) { // loop through all but the last
    char c = Wire.read(); // receive byte as a character
    Serial.print(c);         // print the character
  }
  int x = Wire.read();    // receive byte as an integer
  Serial.println(x);         // print the integer
}
```

```
// Wire Slave Sender
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Sends data as an I2C/TWI slave device
// Refer to the "Wire Master Reader" example for use with this

// Created 29 March 2006
```

```
// This example code is in the public domain.


#include <Wire.h>

void setup() {
  Wire.begin(8);                // join i2c bus with address #8
  Wire.onRequest(requestEvent); // register event
}

void loop() {
  delay(100);
}

// function that executes whenever data is requested by master
// this function is registered as an event, see setup()
void requestEvent() {
  Wire.write("hello "); // respond with message of 6 bytes
  // as expected by master
}
```

Result

Examples compiled and uploaded successfully to the board.

Messages

```
Sketch uses 2220 bytes (1%) of program storage space. Maximum is 131072 bytes.
Global variables use 333 bytes (2%) of dynamic memory, leaving 16051 bytes for
local variables. Maximum is 16384 bytes.

avrdude: Version 6.3-20201216
         Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
         Copyright (c) 2007-2014 Joerg Wunsch

         System wide configuration file is
"C:\Users\ivanFernandez\AppData\Local\Arduino15\packages\Microchip\hardware\megaav
r\1.0.0/avrdude.conf"

         Using Port                    : usb
         Using Programmer              : curiosity_updi
avrdude: Found CMSIS-DAP compliant device, using EDBG protocol
         AVR Part                      : AVR128DA48
         Chip Erase delay              : 0 us
         PAGEL                         : P00
         BS2                           : P00
         RESET disposition             : dedicated
         RETRY pulse                   : SCK
```

```
          serial program mode           : yes
          parallel program mode         : yes
          Timeout                       : 0
          StabDelay                     : 0
          CmdexeDelay                   : 0
          SyncLoops                     : 0
          ByteDelay                     : 0
          PollIndex                     : 0
          PollValue                     : 0x00
          Memory Detail                 :

                                   Block Poll               Page
Polled
            Memory Type Mode Delay Size  Indx Paged  Size   Size #Pages MinW  MaxW
ReadBack
            ----------- ---- ----- ----- ---- ------ ------ ---- ------ ----- -----
---------
            signature      0     0     0    0 no          3    0      0     0     0
0x00 0x00
            prodsig        0     0     0    0 no        125  125      0     0     0
0x00 0x00
            fuses          0     0     0    0 no          9   16      0     0     0
0x00 0x00
            fuse0          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse1          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse2          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse4          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse5          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse6          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse7          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse8          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            lock           0     0     0    0 no          4    1      0     0     0
0x00 0x00
            data           0     0     0    0 no          0    0      0     0     0
0x00 0x00
            flash          0     0     0    0 no     131072  512      0     0     0
0x00 0x00
            eeprom         0     0     0    0 no        512   32      0     0     0
0x00 0x00

          Programmer Type : JTAGICE3_UPDI
          Description     : Microchip Curiosity in UPDI mode
          ICE hardware version: 0
          ICE firmware version: 1.17 (rel. 514)
          Serial number   : MCHP3280031800001901
          Vtarget         : 3.31 V
```

```
          JTAG clock megaAVR/program: 0 kHz
          JTAG clock megaAVR/debug:   0 kHz
          JTAG clock Xmega: 0 kHz
          PDI clock Xmega : 100 kHz

avrdude: Partial Family_ID returned: "    "
avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e9708 (probably avr128da48)
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be performed
         To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "0b11001001"
avrdude: writing fuse5 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse5 written
avrdude: verifying fuse5 memory against 0b11001001:
avrdude: load data fuse5 data from input file 0b11001001:
avrdude: input file 0b11001001 contains 1 bytes
avrdude: reading on-chip fuse5 data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse5 verified
avrdude: reading input file "0x00"
avrdude: writing fuse7 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse7 written
avrdude: verifying fuse7 memory against 0x00:
avrdude: load data fuse7 data from input file 0x00:
avrdude: input file 0x00 contains 1 bytes
avrdude: reading on-chip fuse7 data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse7 verified
avrdude: reading input file "0x00"
avrdude: writing fuse8 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse8 written
avrdude: verifying fuse8 memory against 0x00:
avrdude: load data fuse8 data from input file 0x00:
avrdude: input file 0x00 contains 1 bytes
avrdude: reading on-chip fuse8 data:
```

```
Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse8 verified
avrdude: reading input file
"C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_59380/slave_sender.ino.hex"
avrdude: writing flash (2220 bytes):

Writing | ################################################## | 100% 0.79s

avrdude: 2220 bytes of flash written
avrdude: verifying flash memory against
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_59380/slave_sender.ino.hex:
avrdude: load data flash data from input file
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_59380/slave_sender.ino.hex:
avrdude: input file
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_59380/slave_sender.ino.hex
contains 2220 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.43s

avrdude: verifying ...
avrdude: 2220 bytes of flash verified

avrdude done.  Thank you.
```

```
Sketch uses 3848 bytes (2%) of program storage space. Maximum is 131072 bytes.
Global variables use 632 bytes (3%) of dynamic memory, leaving 15752 bytes for
local variables. Maximum is 16384 bytes.

avrdude: Version 6.3-20201216
         Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
         Copyright (c) 2007-2014 Joerg Wunsch

         System wide configuration file is
"C:\Users\ivanFernandez\AppData\Local\Arduino15\packages\Microchip\hardware\megaav
r\1.0.0/avrdude.conf"

         Using Port                    : usb
         Using Programmer              : curiosity_updi
avrdude: Found CMSIS-DAP compliant device, using EDBG protocol
         AVR Part                      : AVR128DA48
         Chip Erase delay              : 0 us
         PAGEL                         : P00
         BS2                           : P00
         RESET disposition             : dedicated
         RETRY pulse                   : SCK
```

```
          serial program mode           : yes
          parallel program mode         : yes
          Timeout                       : 0
          StabDelay                     : 0
          CmdexeDelay                   : 0
          SyncLoops                     : 0
          ByteDelay                     : 0
          PollIndex                     : 0
          PollValue                     : 0x00
          Memory Detail                 :

                                   Block Poll               Page
Polled
            Memory Type Mode Delay Size  Indx Paged  Size   Size #Pages MinW  MaxW
ReadBack
            ----------- ---- ----- ----- ---- ------ ------ ---- ------ ----- -----
---------
            signature      0     0     0    0 no          3    0      0     0     0
0x00 0x00
            prodsig        0     0     0    0 no        125  125      0     0     0
0x00 0x00
            fuses          0     0     0    0 no          9   16      0     0     0
0x00 0x00
            fuse0          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse1          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse2          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse4          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse5          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse6          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse7          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            fuse8          0     0     0    0 no          1    0      0     0     0
0x00 0x00
            lock           0     0     0    0 no          4    1      0     0     0
0x00 0x00
            data           0     0     0    0 no          0    0      0     0     0
0x00 0x00
            flash          0     0     0    0 no     131072  512      0     0     0
0x00 0x00
            eeprom         0     0     0    0 no        512   32      0     0     0
0x00 0x00

          Programmer Type : JTAGICE3_UPDI
          Description     : Microchip Curiosity in UPDI mode
          ICE hardware version: 0
          ICE firmware version: 1.17 (rel. 514)
          Serial number   : MCHP3280031800001901
          Vtarget         : 3.31 V
```

```
         JTAG clock megaAVR/program: 0 kHz
         JTAG clock megaAVR/debug:   0 kHz
         JTAG clock Xmega: 0 kHz
         PDI clock Xmega : 100 kHz

avrdude: Partial Family_ID returned: "    "
avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e9708 (probably avr128da48)
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be performed
         To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "0b11001001"
avrdude: writing fuse5 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse5 written
avrdude: verifying fuse5 memory against 0b11001001:
avrdude: load data fuse5 data from input file 0b11001001:
avrdude: input file 0b11001001 contains 1 bytes
avrdude: reading on-chip fuse5 data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse5 verified
avrdude: reading input file "0x00"
avrdude: writing fuse7 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse7 written
avrdude: verifying fuse7 memory against 0x00:
avrdude: load data fuse7 data from input file 0x00:
avrdude: input file 0x00 contains 1 bytes
avrdude: reading on-chip fuse7 data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse7 verified
avrdude: reading input file "0x00"
avrdude: writing fuse8 (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of fuse8 written
avrdude: verifying fuse8 memory against 0x00:
avrdude: load data fuse8 data from input file 0x00:
avrdude: input file 0x00 contains 1 bytes
avrdude: reading on-chip fuse8 data:
```

```
Reading | ################################################## | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of fuse8 verified
avrdude: reading input file
"C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_282424/slave_receiver.ino.hex"
avrdude: writing flash (3848 bytes):

Writing | ################################################## | 100% 1.26s

avrdude: 3848 bytes of flash written
avrdude: verifying flash memory against
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_282424/slave_receiver.ino.hex:
avrdude: load data flash data from input file
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_282424/slave_receiver.ino.hex:
avrdude: input file
C:\Users\IVANFE~1\AppData\Local\Temp\arduino_build_282424/slave_receiver.ino.hex
contains 3848 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.69s

avrdude: verifying ...
avrdude: 3848 bytes of flash verified

avrdude done.  Thank you.
```

Notes

1. Each of the sketches compiled and uploaded successfully to the AVR128DA48 board. This concludes testing of the Wire examples within the Team 25 core.