CEC Continuing Education Center

DesignNews

Arduino Pro Primer

Day 3: Debugging Arduino Sketches

Sponsored by

DigiKey

informa markets

Webinar Logistics

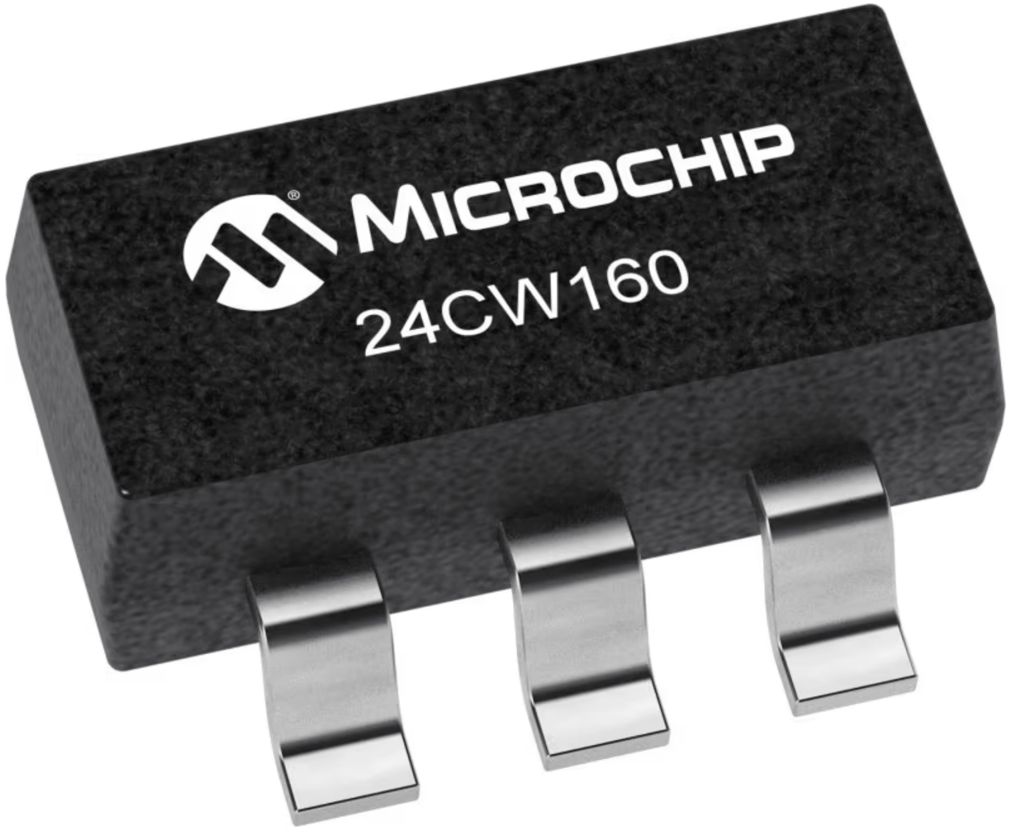
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

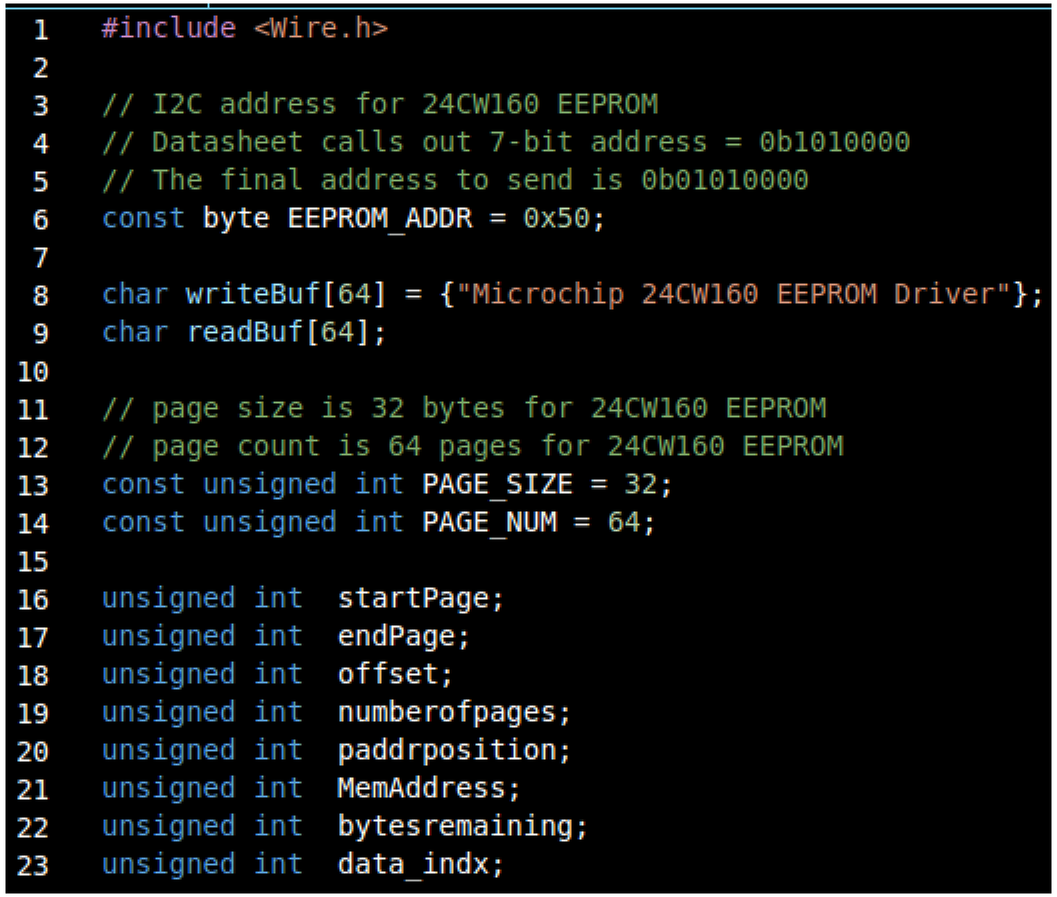
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

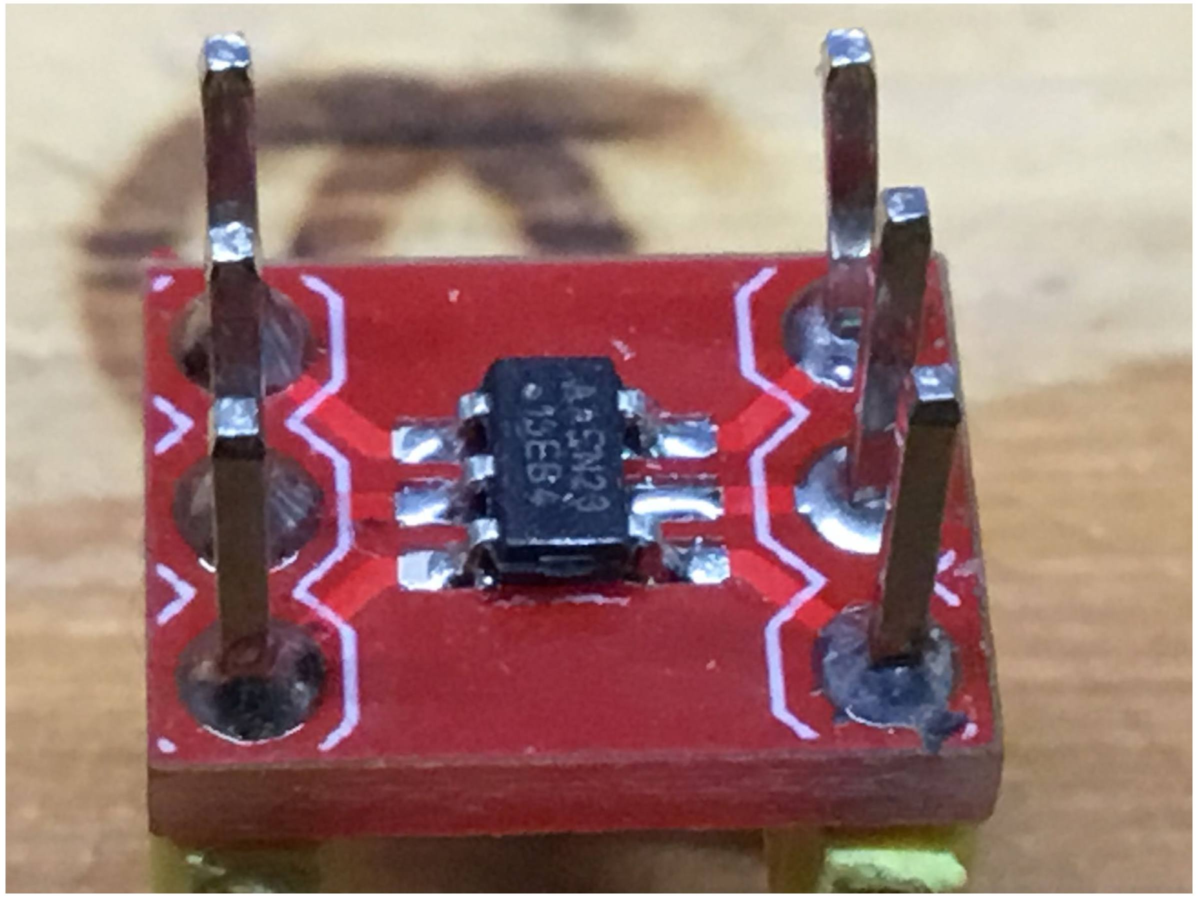
- Build a Microchip 24CW160 EEPROM Driver Sketch
- Debug the Microchip 24CW160 EEPROM Driver



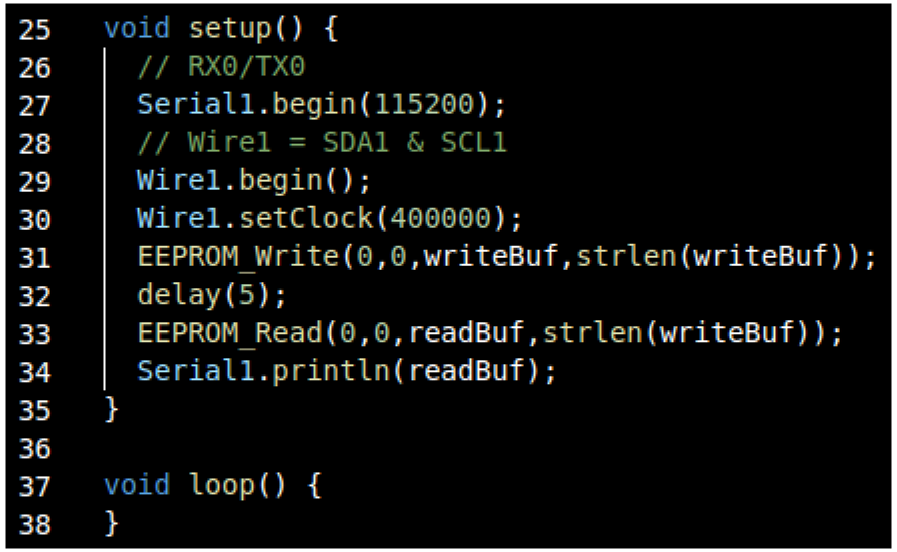
24CW160 Driver Variables and Constants



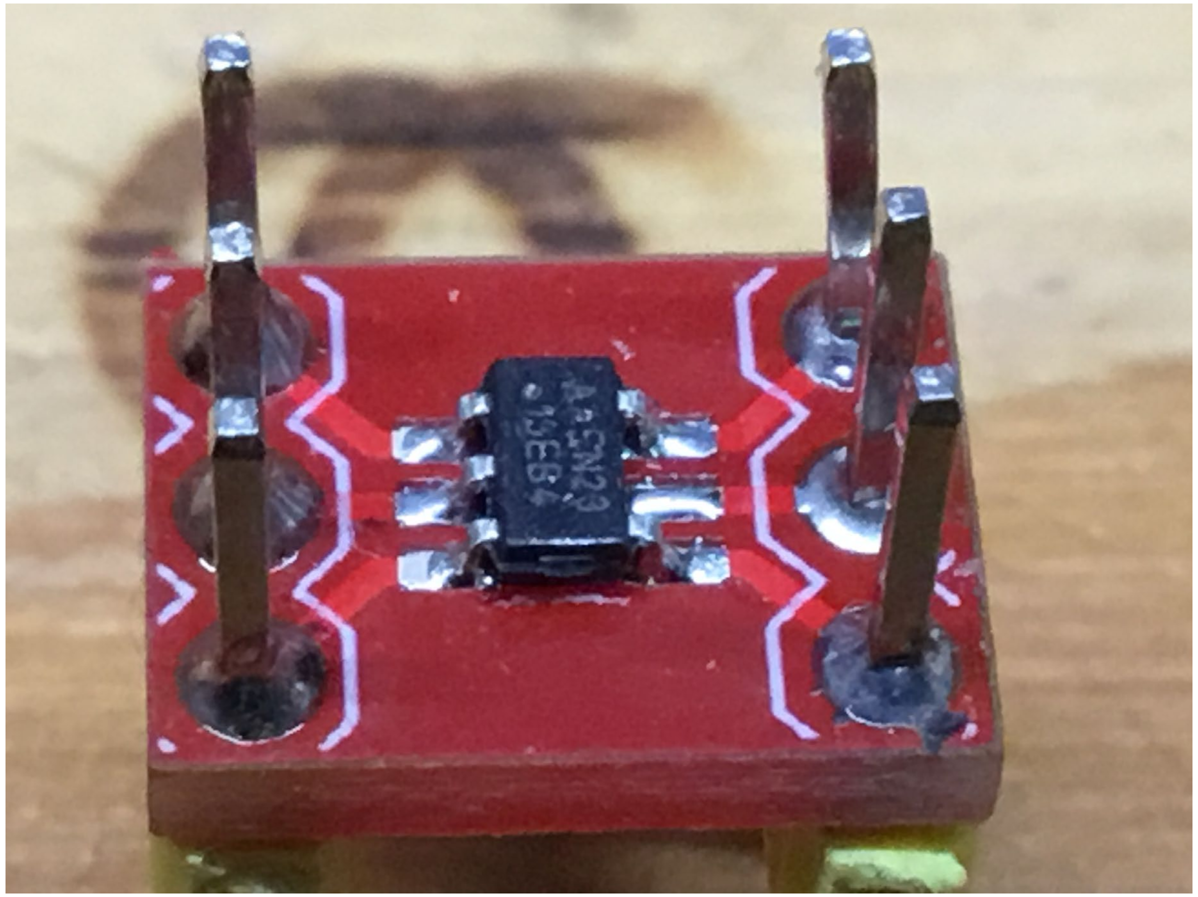
```
#include <Wire.h>

// I2C address for 24CW160 EEPROM
// Datasheet calls out 7-bit address = 0b1010000
// The final address to send is 0b01010000
const byte EEPROM_ADDR = 0x50;

char writeBuf[64] = {"Microchip 24CW160 EEPROM Driver"};
char readBuf[64];

// page size is 32 bytes for 24CW160 EEPROM
// page count is 64 pages for 24CW160 EEPROM
const unsigned int PAGE_SIZE = 32;
const unsigned int PAGE_NUM = 64;

unsigned int  startPage;
unsigned int  endPage;
unsigned int  offset;
unsigned int  numberofpages;
unsigned int  paddrposition;
unsigned int  MemAddress;
unsigned int  bytesremaining;
unsigned int  data_indx;
```



24CW160 Driver *setup()* and *loop()*

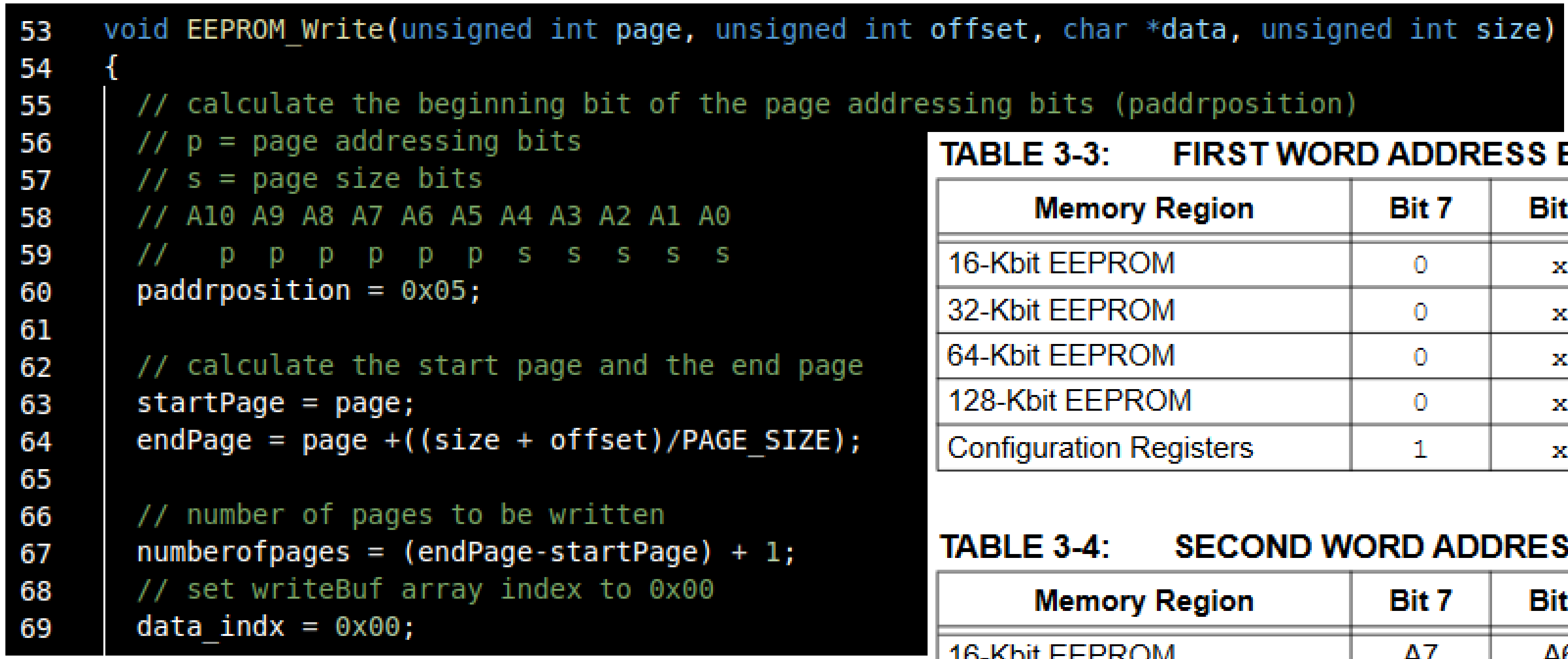


```
void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}
```



24CW160 Driver - EEPROM_Write Function

```
sssss = 0x00-0x1F bytes per page (maximum 32 bytes per page)
pppppp =  0x00-0x3F pages (maximum 40 pages)
```



```
void EEPROM_Write(unsigned int page, unsigned int offset, char *data, unsigned int size)
{
  // calculate the beginning bit of the page addressing bits (paddrposition)
  // p = page addressing bits
  // s = page size bits
  // A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
  //   p  p  p  p  p  p  s  s  s  s  s
  paddrposition = 0x05;

  // calculate the start page and the end page
  startPage = page;
  endPage = page +((size + offset)/PAGE_SIZE);

  // number of pages to be written
  numberofpages = (endPage-startPage) + 1;
  // set writeBuf array index to 0x00
  data_indx = 0x00;
```

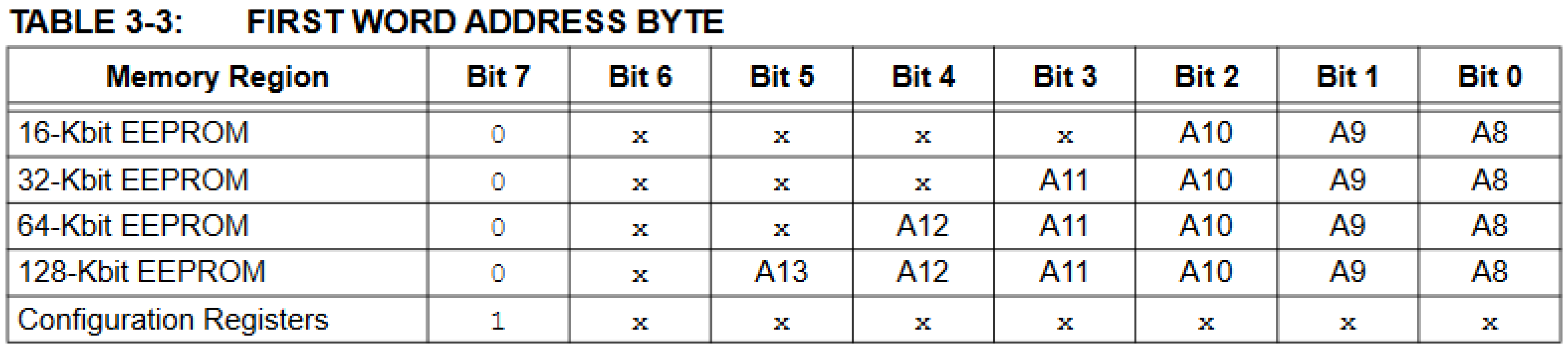


TABLE 3-3: FIRST WORD ADDRESS BYTE

Memory Region	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-Kbit EEPROM	0	x	x	x	x	A10	A9	A8
32-Kbit EEPROM	0	x	x	x	A11	A10	A9	A8
64-Kbit EEPROM	0	x	x	A12	A11	A10	A9	A8
128-Kbit EEPROM	0	x	A13	A12	A11	A10	A9	A8
Configuration Registers	1	x	x	x	x	x	x	x

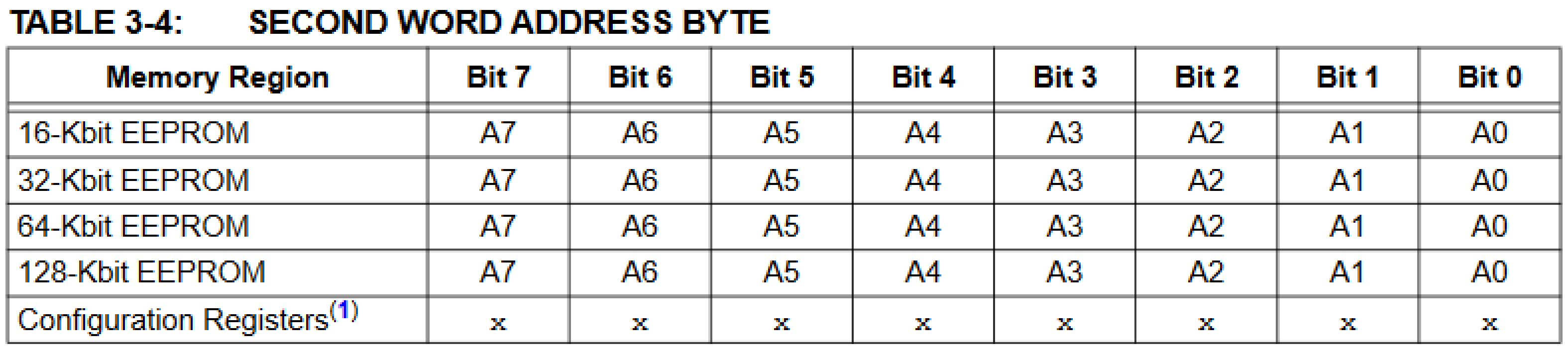
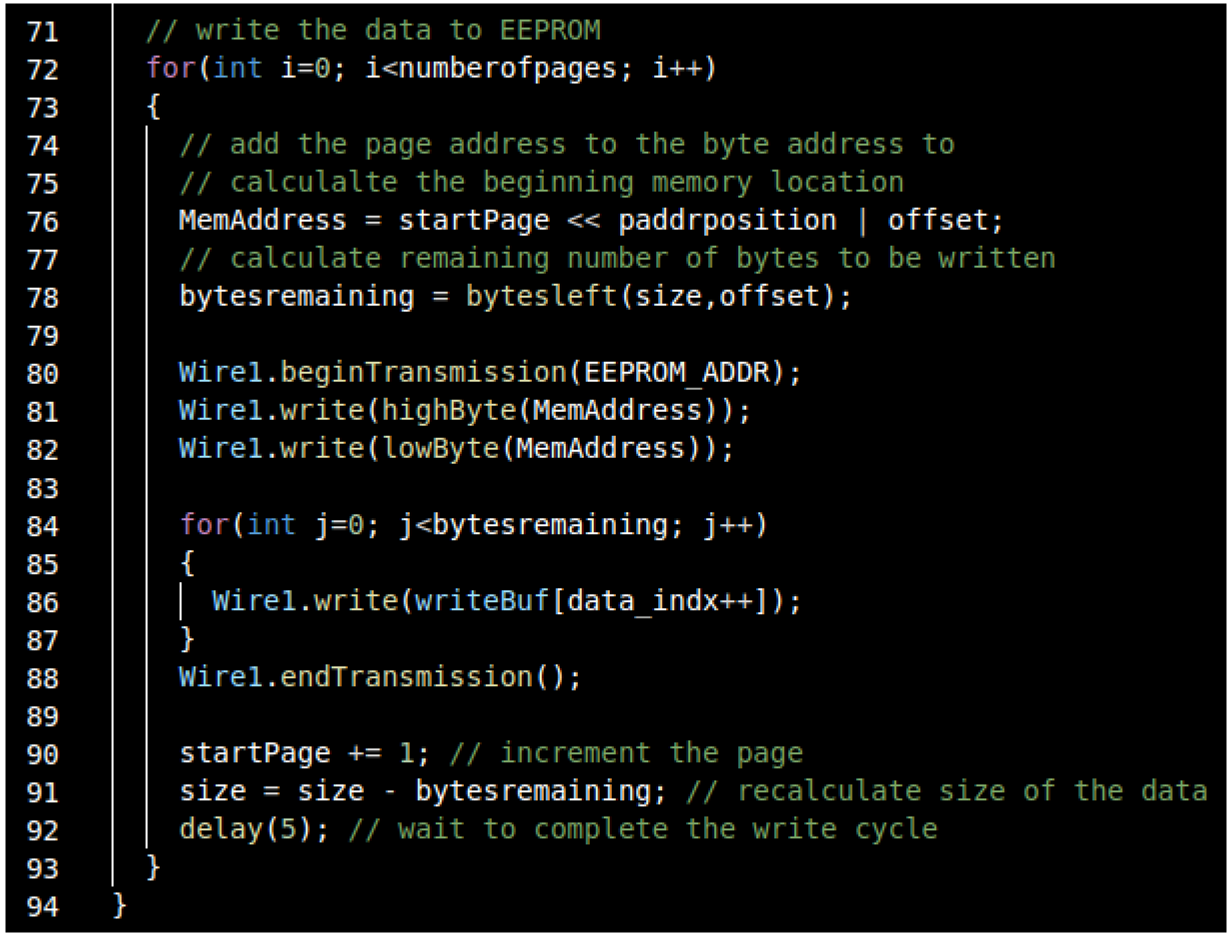


TABLE 3-4: SECOND WORD ADDRESS BYTE

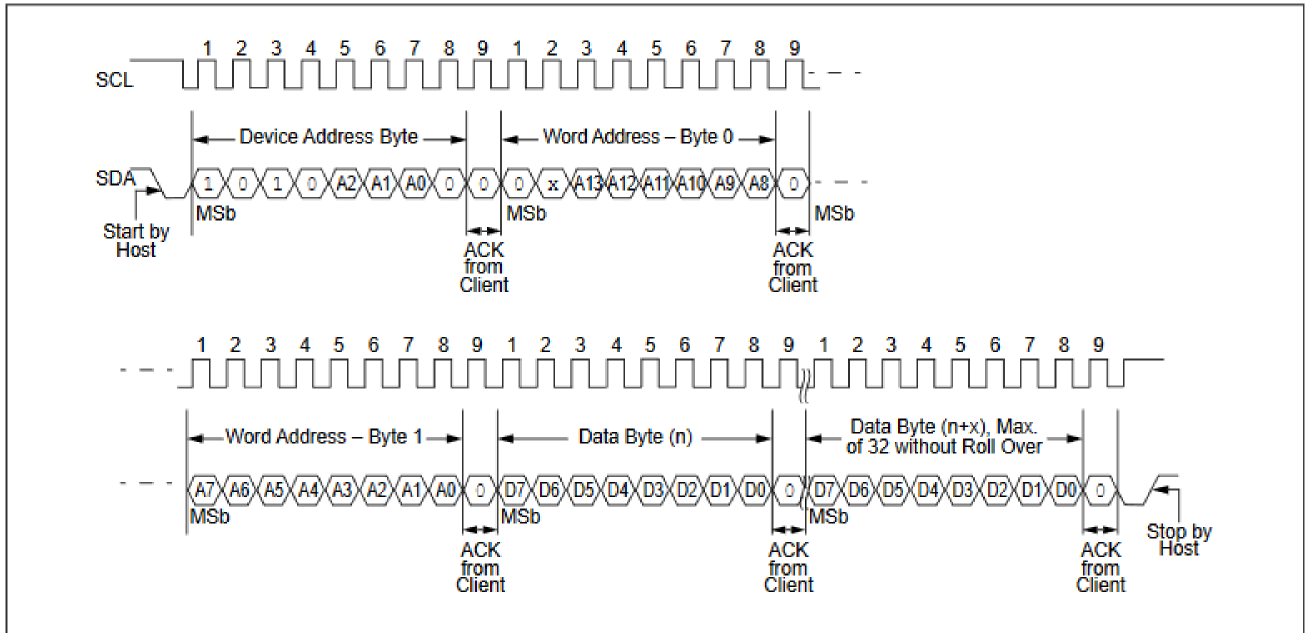
Memory Region	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16-Kbit EEPROM	A7	A6	A5	A4	A3	A2	A1	A0
32-Kbit EEPROM	A7	A6	A5	A4	A3	A2	A1	A0
64-Kbit EEPROM	A7	A6	A5	A4	A3	A2	A1	A0
128-Kbit EEPROM	A7	A6	A5	A4	A3	A2	A1	A0
Configuration Registers(1)	x	x	x	x	x	x	x	x

24CW160 Driver - EEPROM_Write Function



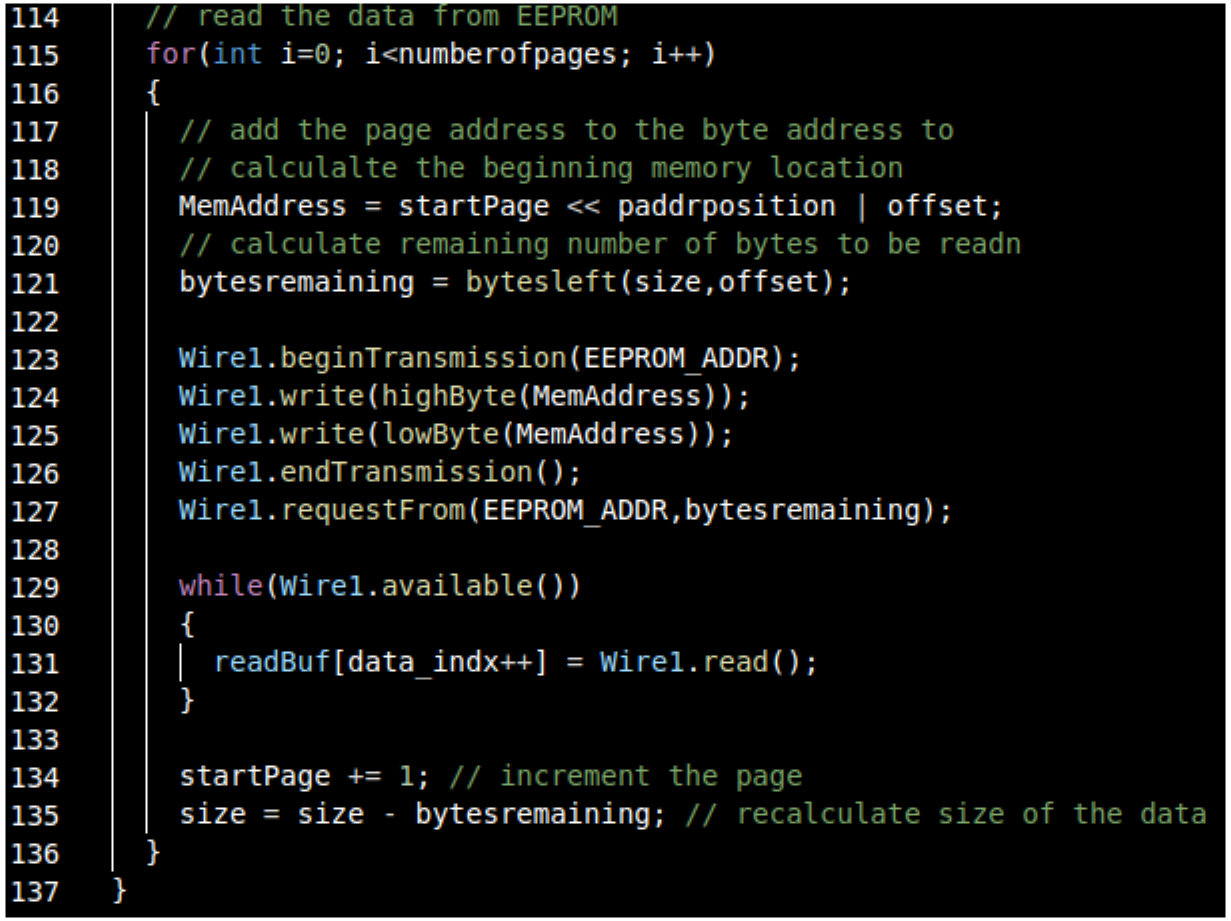
```
  // write the data to EEPROM
  for(int i=0; i<numberofpages; i++)
  {
    // add the page address to the byte address to
    // calculalte the beginning memory location
    MemAddress = startPage << paddrposition | offset;
    // calculate remaining number of bytes to be written
    bytesremaining = bytesleft(size,offset);

    Wire1.beginTransmission(EEPROM_ADDR);
    Wire1.write(highByte(MemAddress));
    Wire1.write(lowByte(MemAddress));

    for(int j=0; j<bytesremaining; j++)
    {
      Wire1.write(writeBuf[data_indx++]);
    }
    Wire1.endTransmission();

    startPage += 1; // increment the page
    size = size - bytesremaining; // recalculate size of the data
    delay(5); // wait to complete the write cycle
  }
}
```

FIGURE 6-2: PAGE WRITE



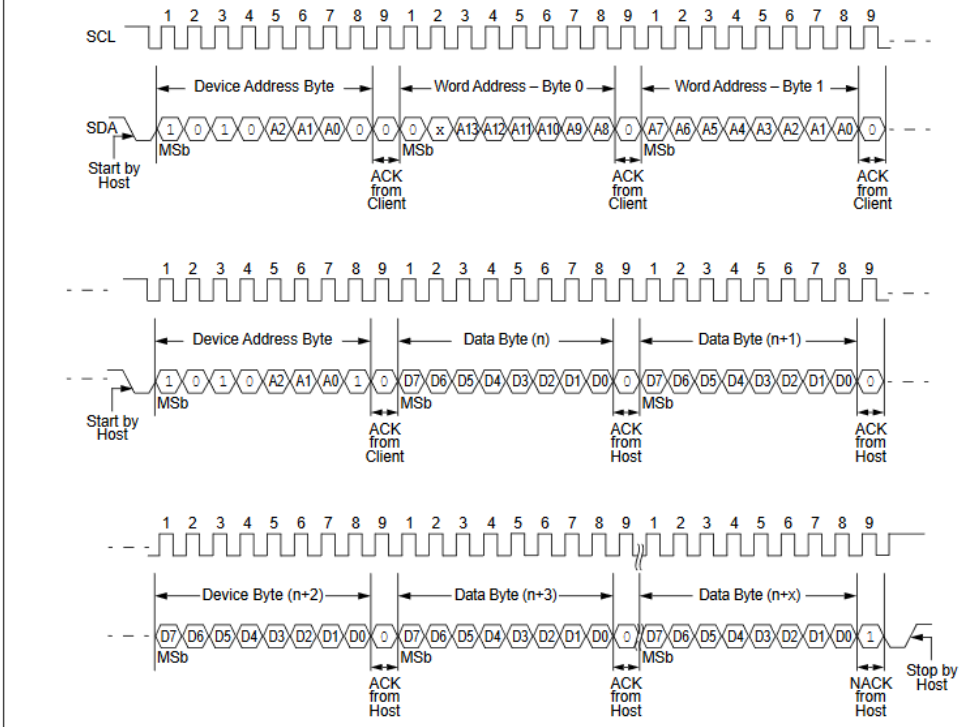
Note 1: The A13, A12 and A11 word address bits are “don’t care” bits on the 24CW16X.
2: The A13 and A12 word address bits are “don’t care” bits on the 24CW32X.
3: The A13 word address bit is a “don’t care” bit on the 24CW64X.

24CW160 Driver - EEPROM_Read Function



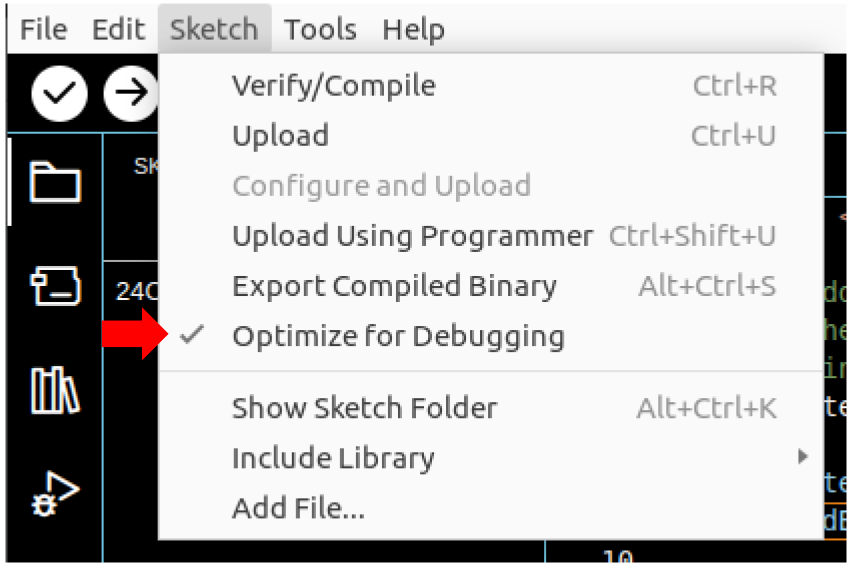
```
  // read the data from EEPROM
  for(int i=0; i<numberofpages; i++)
  {
    // add the page address to the byte address to
    // calculalte the beginning memory location
    MemAddress = startPage << paddrposition | offset;
    // calculate remaining number of bytes to be readn
    bytesremaining = bytesleft(size,offset);

    Wire1.beginTransmission(EEPROM_ADDR);
    Wire1.write(highByte(MemAddress));
    Wire1.write(lowByte(MemAddress));
    Wire1.endTransmission();
    Wire1.requestFrom(EEPROM_ADDR,bytesremaining);

    while(Wire1.available())
    {
      readBuf[data_indx++] = Wire1.read();
    }

    startPage += 1; // increment the page
    size = size - bytesremaining; // recalculate size of the data
  }
}
```

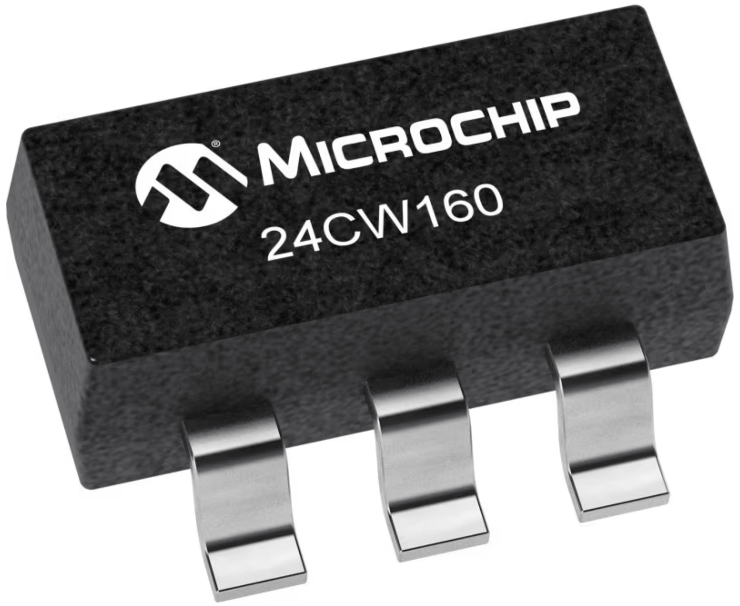
FIGURE 7-3: SEQUENTIAL READ

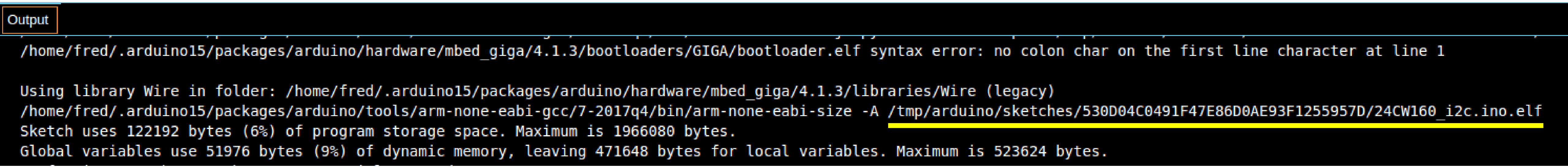


Note 1: The A13, A12 and A11 word address bits are “don’t care” bits on the 24CW16X.
2: The A13 and A12 word address bits are “don’t care” bits on the 24CW32X.
3: The A13 word address bit is a “don’t care” bit on the 24CW64X.

Verify the 24CW160 Driver


File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Configure and Upload
Upload Using Programmer Ctrl+Shift+U
Export Compiled Binary Alt+Ctrl+S
✓ Optimize for Debugging
Show Sketch Folder Alt+Ctrl+K
Include Library
Add File...

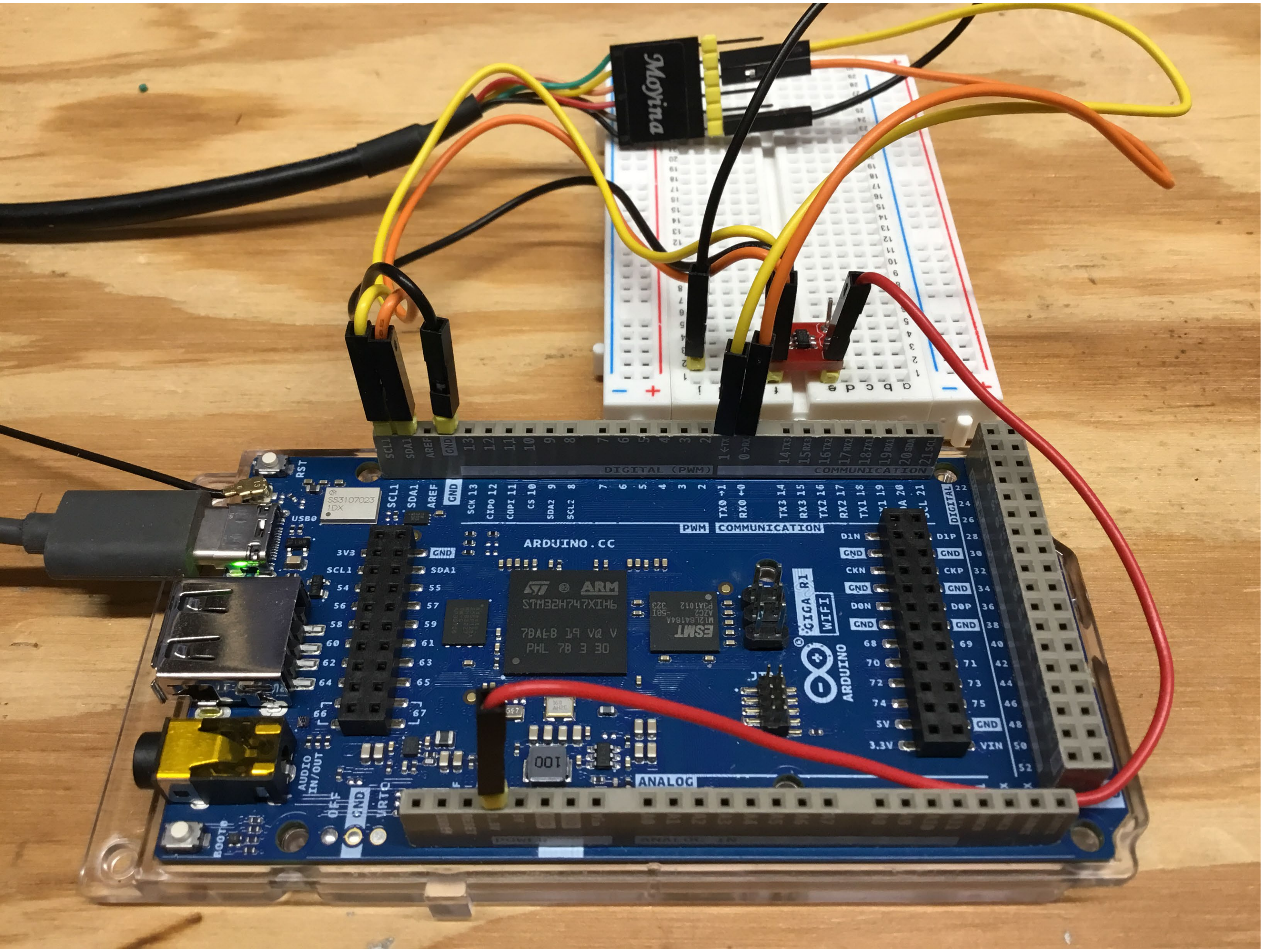


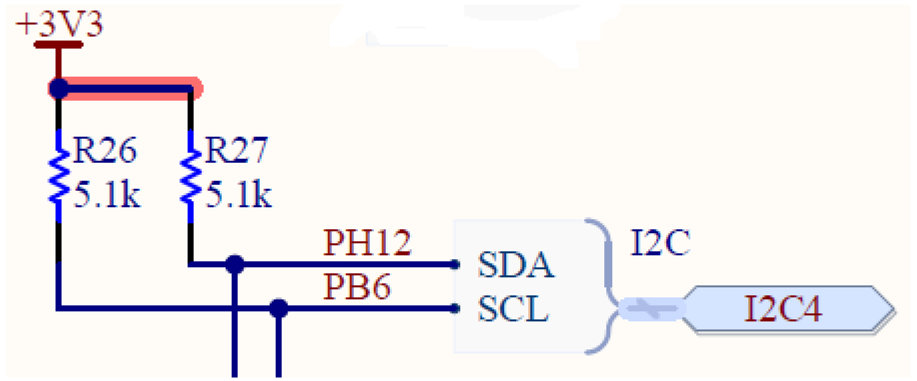


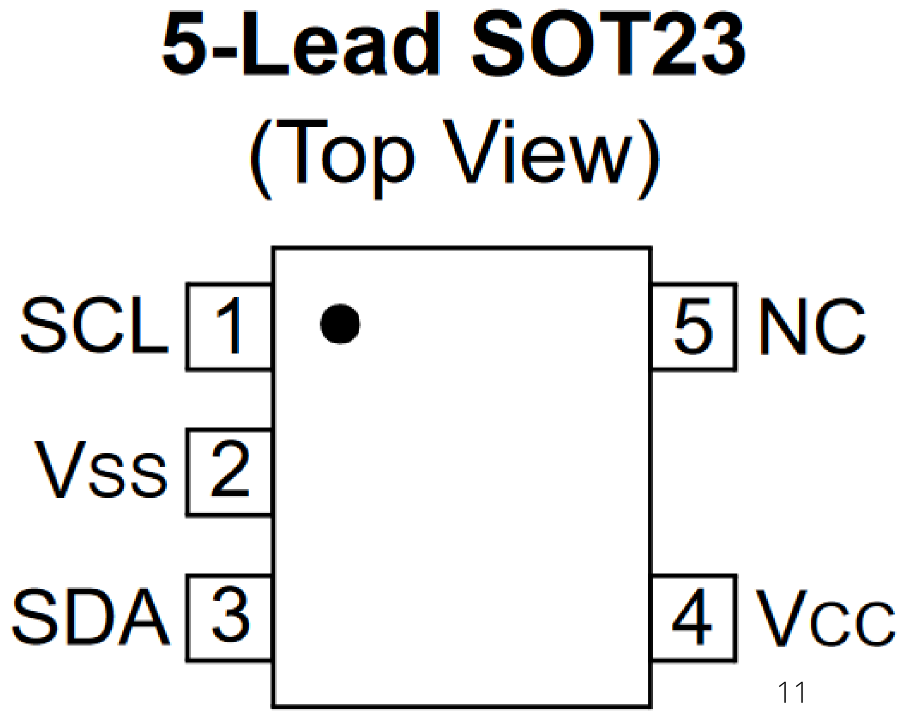
Output

```
/home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/bootloaders/GIGA/bootloader.elf syntax error: no colon char on the first line character at line 1

Using library Wire in folder: /home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/libraries/Wire (legacy)
/home/fred/.arduino15/packages/arduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-size -A /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf
Sketch uses 122192 bytes (6%) of program storage space. Maximum is 1966080 bytes.
Global variables use 51976 bytes (9%) of dynamic memory, leaving 471648 bytes for local variables. Maximum is 523624 bytes.
```

Wire It All Up







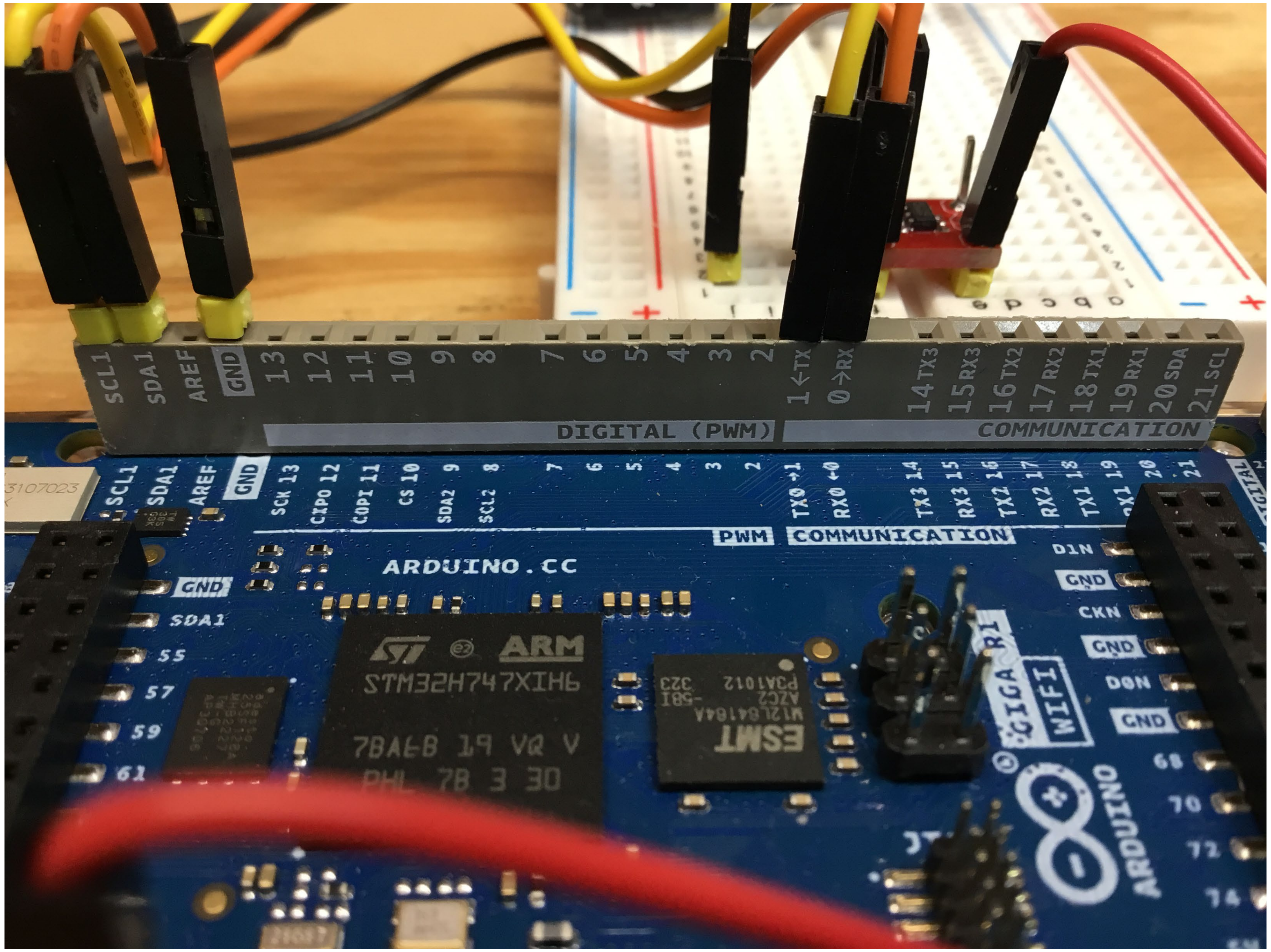
Continuing Education Center

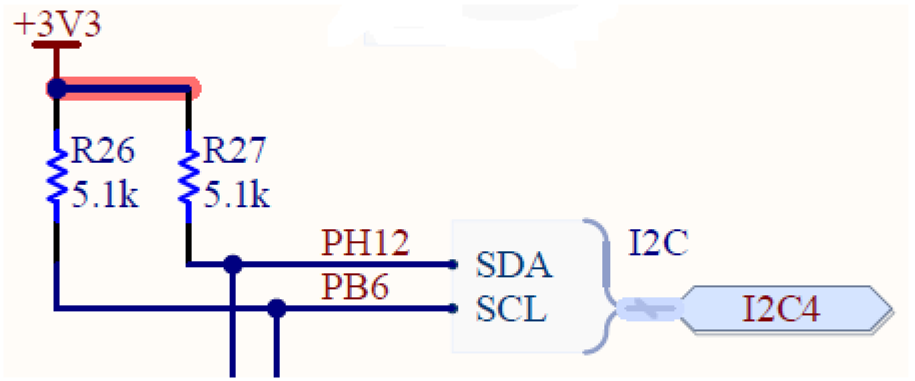
Arduino Pro Primer

Debugging Arduino Sketches

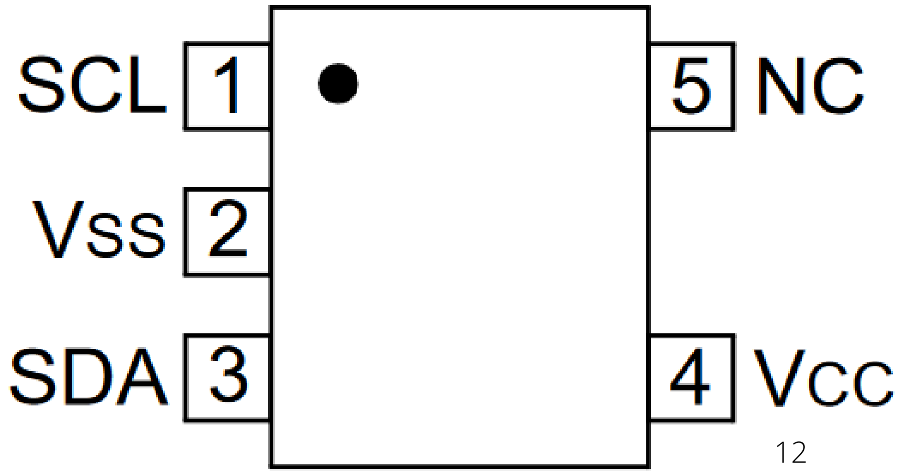
Debug the Microchip 24CW160 EEPROM Driver

Wire It All Up

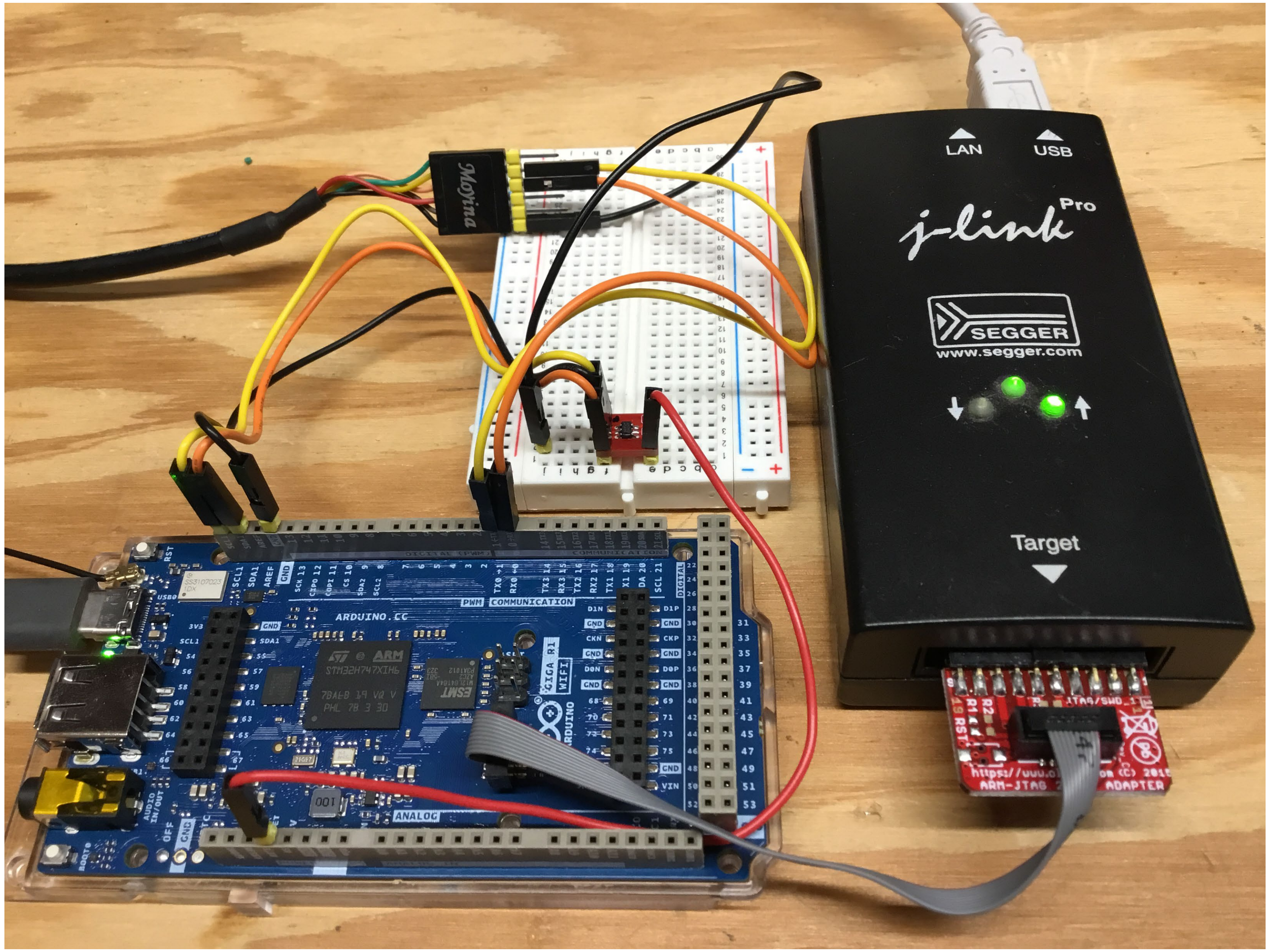




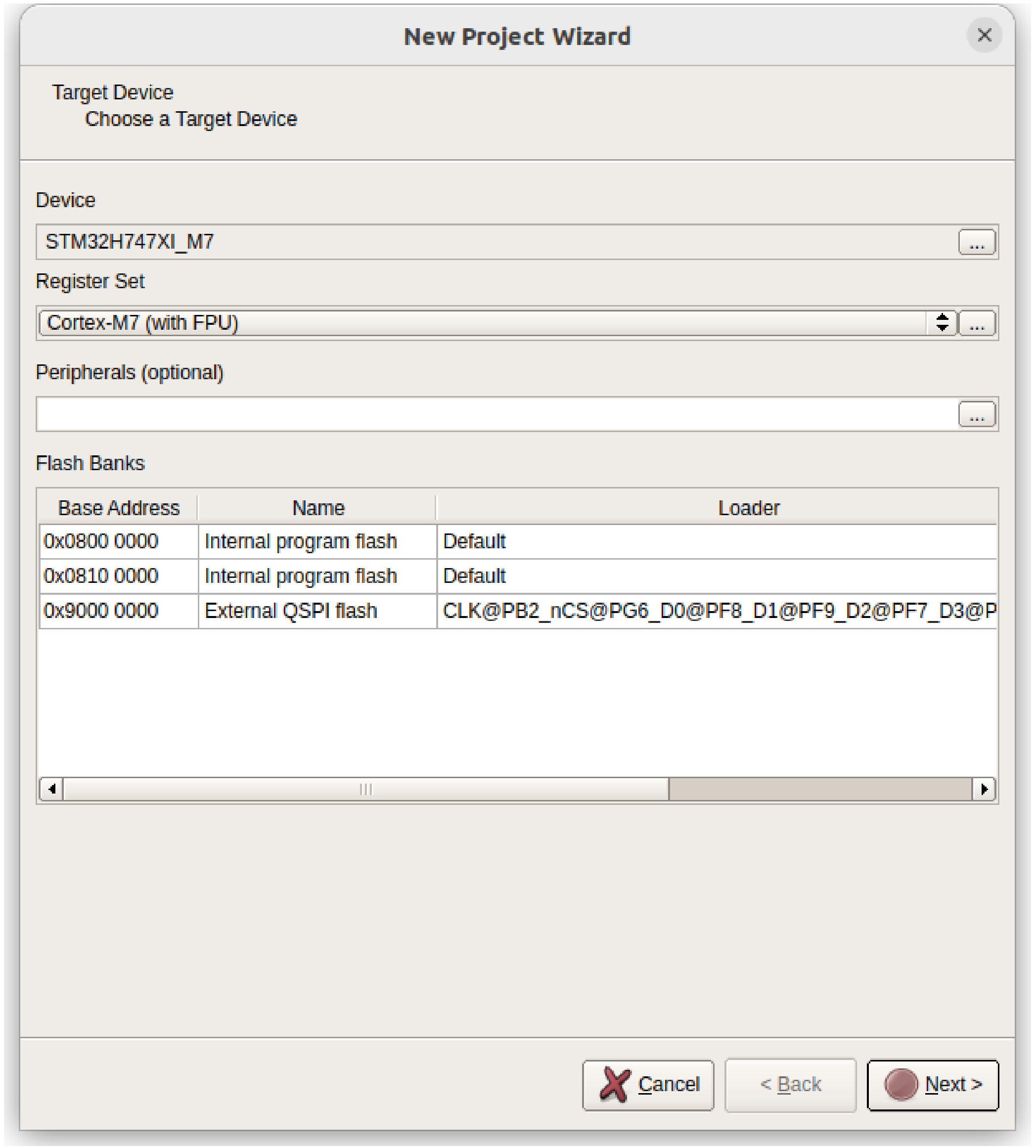
5-Lead SOT23
(Top View)



Wire It All Up – Attach the J-Link



Fire Up the Ozone Debugger



Target Device
Choose a Target Device

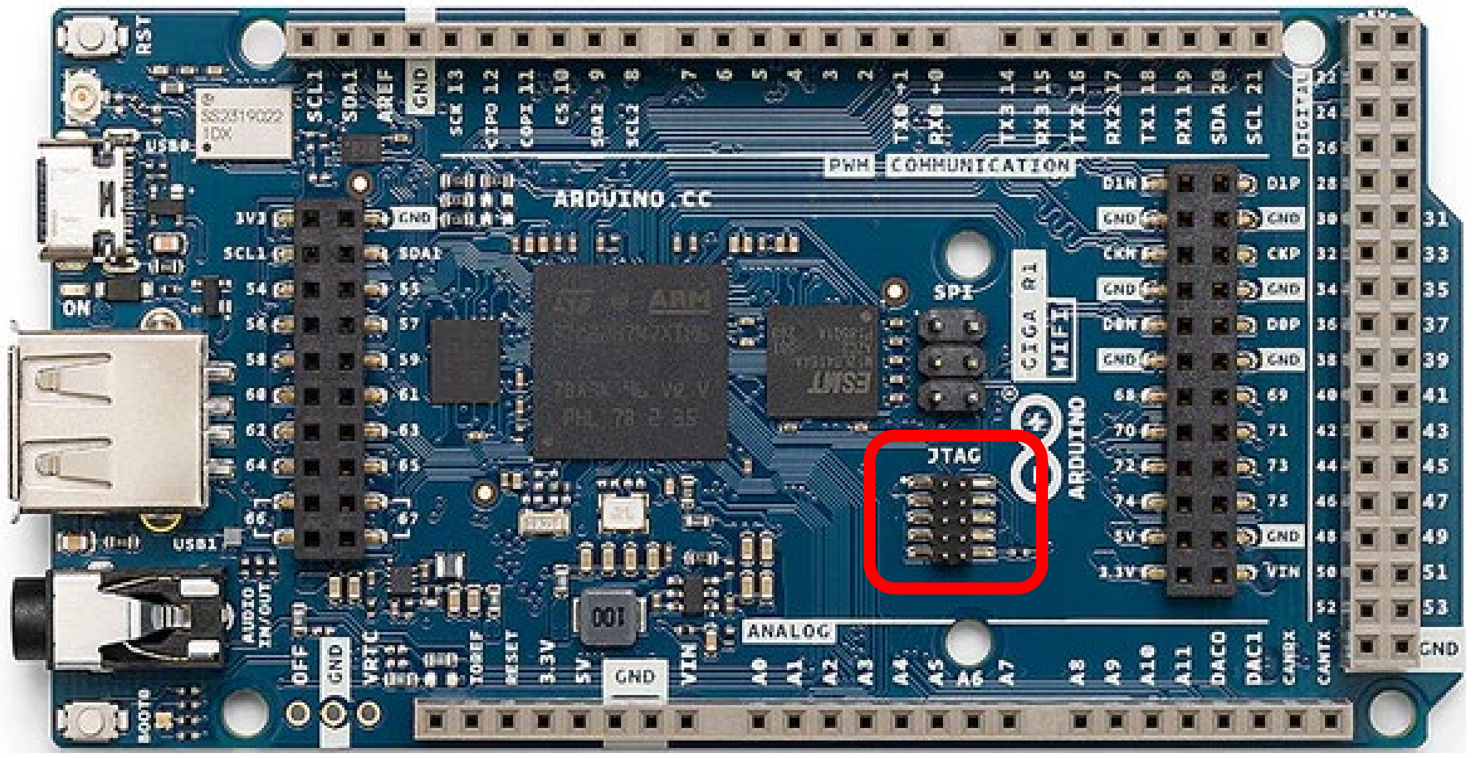
Device: STM32H747XI_M7

Register Set: Cortex-M7 (with FPU)

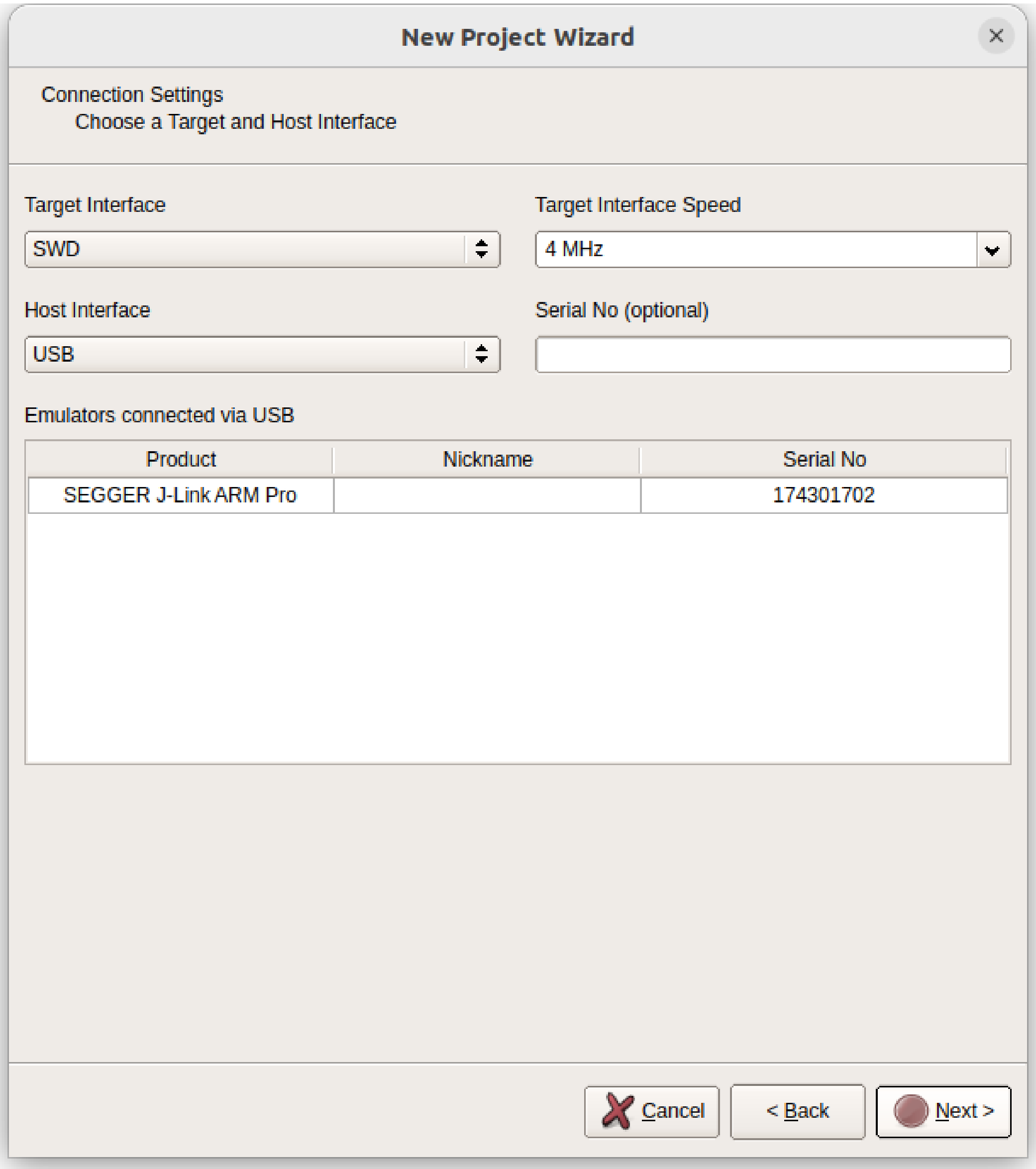
Peripherals (optional):

Flash Banks

Base Address	Name	Loader
0x0800 0000	Internal program flash	Default
0x0810 0000	Internal program flash	Default
0x9000 0000	External QSPI flash	CLK@PB2_nCS@PG6_D0@PF8_D1@PF9_D2@PF7_D3@P



Fire Up the Ozone Debugger

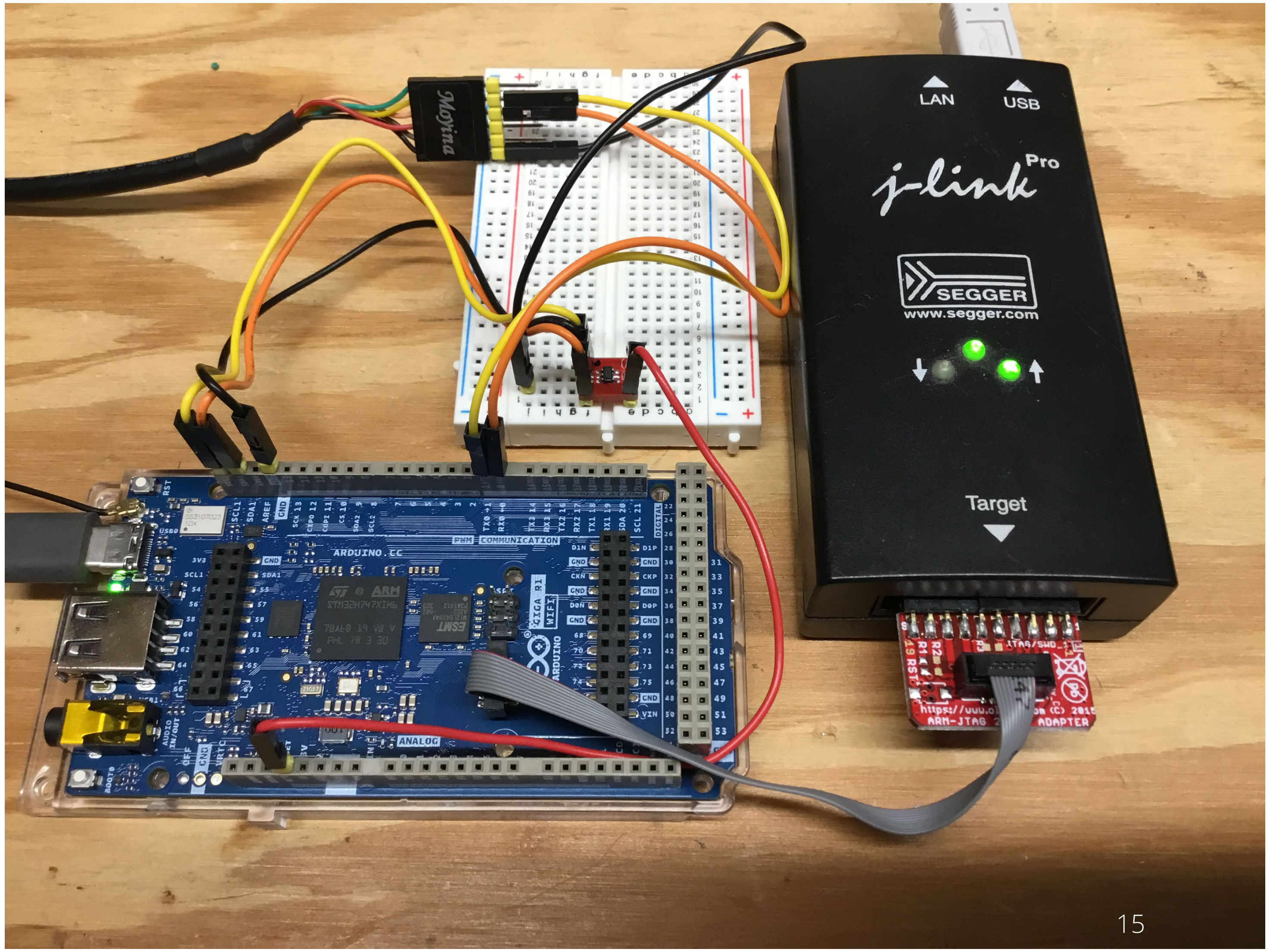

New Project Wizard

Connection Settings
Choose a Target and Host Interface

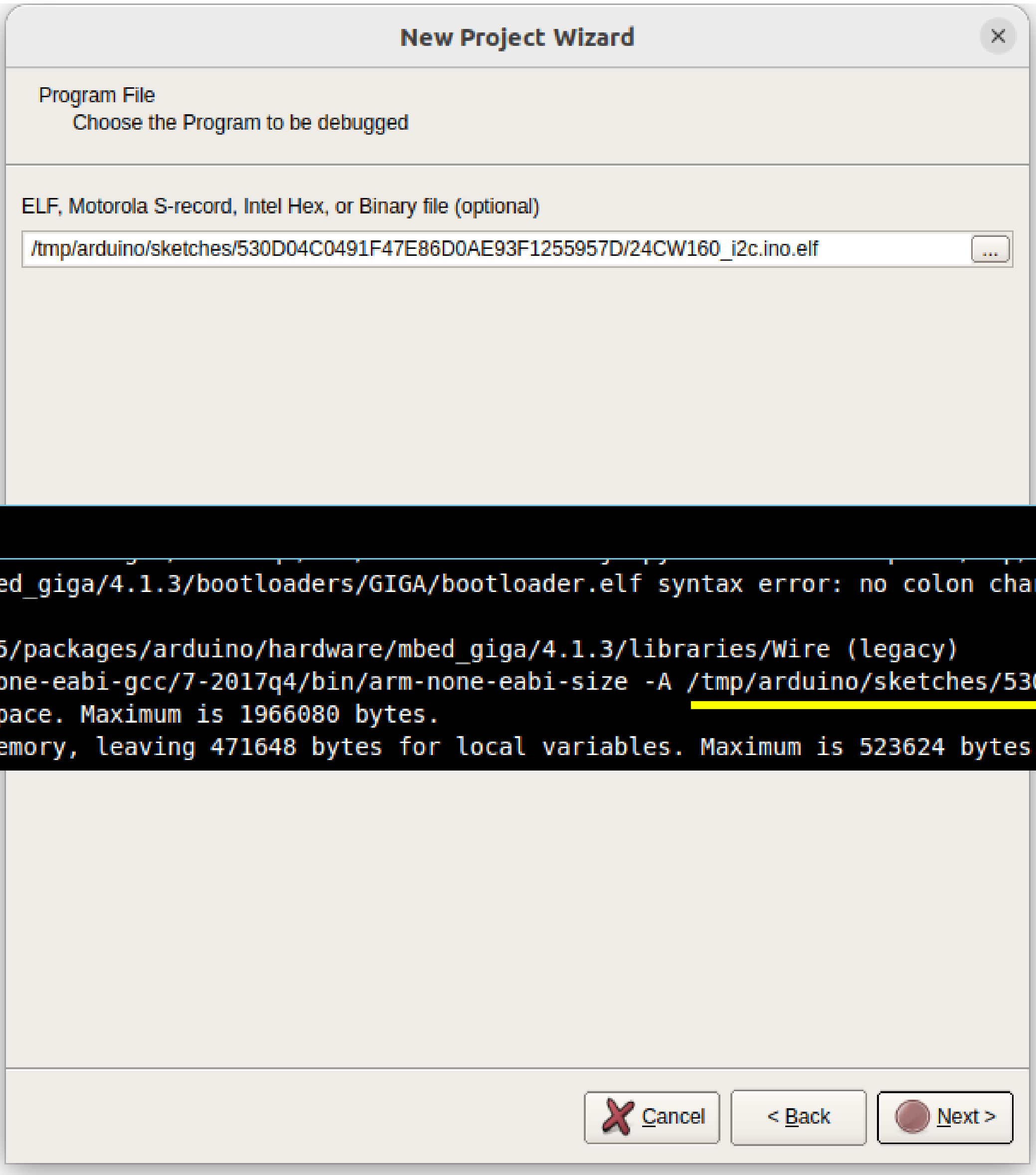
Target Interface: SWD
Target Interface Speed: 4 MHz
Host Interface: USB
Serial No (optional):

Emulators connected via USB

Product	Nickname	Serial No
SEGGER J-Link ARM Pro		174301702



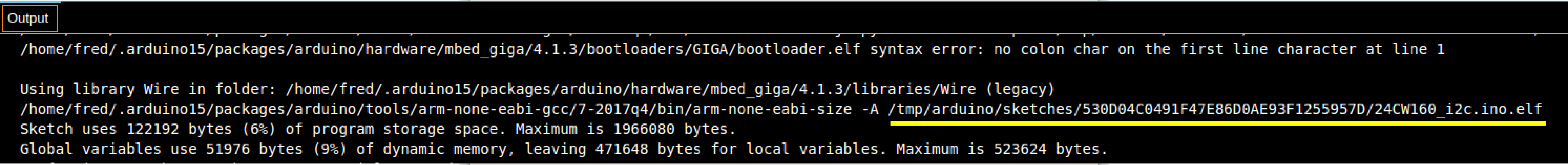
Fire Up the Ozone Debugger



Program File
Choose the Program to be debugged

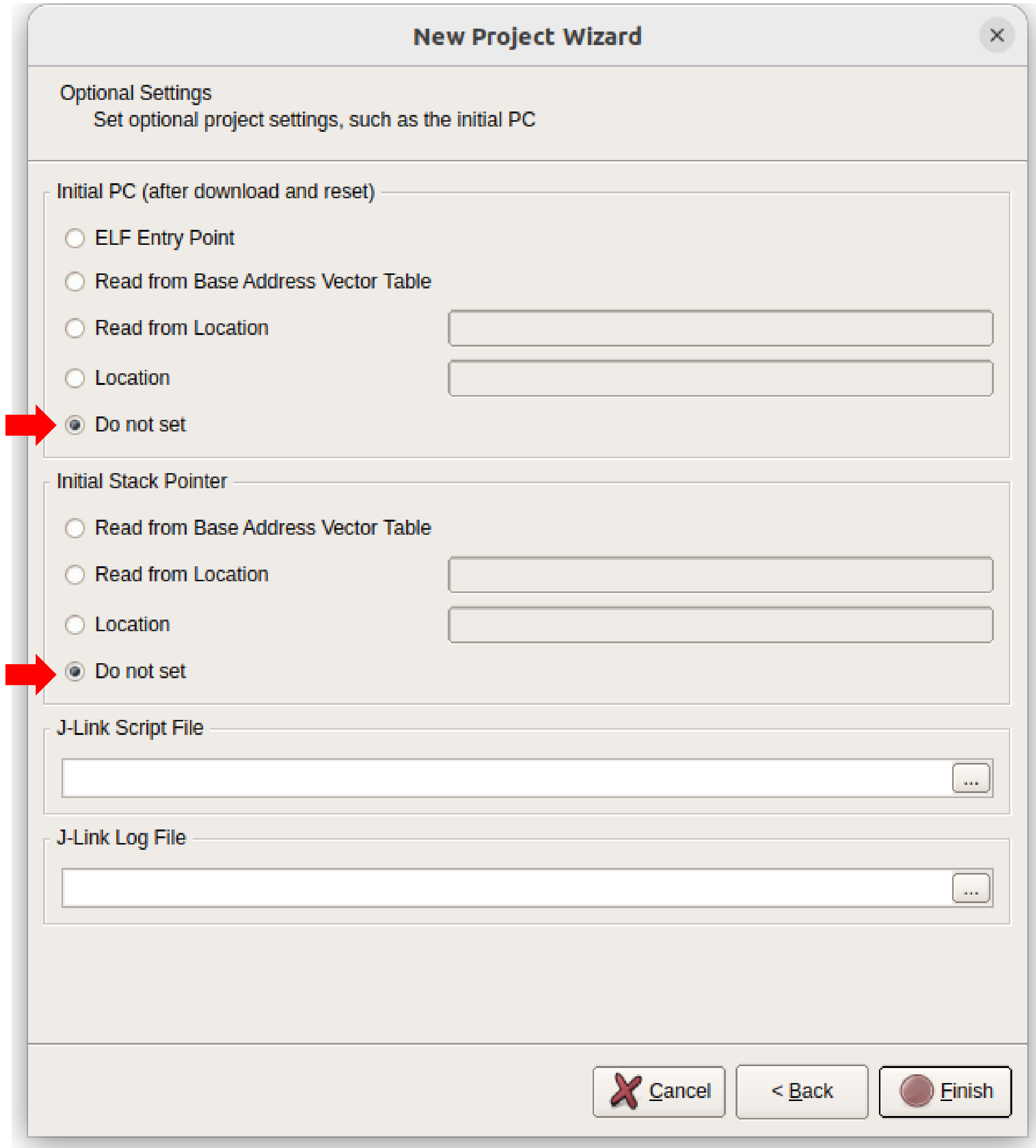
ELF, Motorola S-record, Intel Hex, or Binary file (optional)

/tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf



```
/home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/bootloaders/GIGA/bootloader.elf syntax error: no colon char on the first line character at line 1

Using library Wire in folder: /home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/libraries/Wire (legacy)
/home/fred/.arduino15/packages/arduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-size -A /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf
Sketch uses 122192 bytes (6%) of program storage space. Maximum is 1966080 bytes.
Global variables use 51976 bytes (9%) of dynamic memory, leaving 471648 bytes for local variables. Maximum is 523624 bytes.
```

Fire Up the Ozone Debugger



Optional Settings
Set optional project settings, such as the initial PC

Initial PC (after download and reset)
- ( ) ELF Entry Point
- ( ) Read from Base Address Vector Table
- ( ) Read from Location
- ( ) Location
- (●) Do not set

Initial Stack Pointer
- ( ) Read from Base Address Vector Table
- ( ) Read from Location
- ( ) Location
- (●) Do not set

J-Link Script File

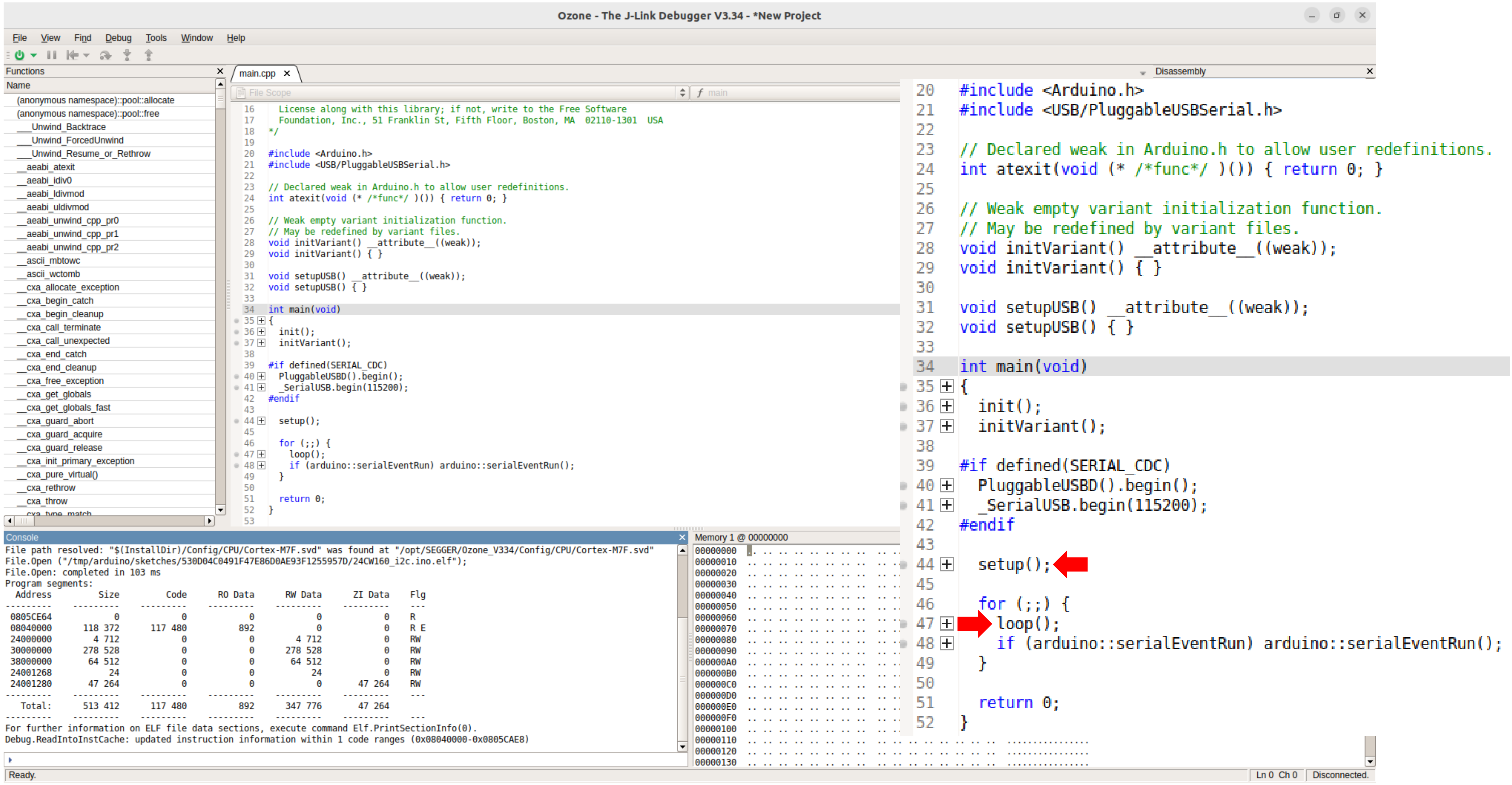
J-Link Log File

Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

A-HA!!



Functions:

```
(anonymous namespace)::pool::allocate
(anonymous namespace)::pool::free
___Unwind_Backtrace
___Unwind_ForcedUnwind
___Unwind_Resume_or_Rethrow
__aeabi_atexit
__aeabi_idiv0
__aeabi_ldivmod
__aeabi_uldivmod
__aeabi_unwind_cpp_pr0
__aeabi_unwind_cpp_pr1
__aeabi_unwind_cpp_pr2
__ascii_mbtowc
__ascii_wctomb
__cxa_allocate_exception
__cxa_begin_catch
__cxa_begin_cleanup
__cxa_call_terminate
__cxa_call_unexpected
__cxa_end_catch
__cxa_end_cleanup
__cxa_free_exception
__cxa_get_globals
__cxa_get_globals_fast
__cxa_guard_abort
__cxa_guard_acquire
__cxa_guard_release
__cxa_init_primary_exception
__cxa_pure_virtual()
__cxa_rethrow
__cxa_throw
__cxa_type_match
```

main.cpp:

```cpp
  License along with this library; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
*/

#include <Arduino.h>
#include <USB/PluggableUSBSerial.h>

// Declared weak in Arduino.h to allow user redefinitions.
int atexit(void (* /*func*/ )()) { return 0; }

// Weak empty variant initialization function.
// May be redefined by variant files.
void initVariant() __attribute__((weak));
void initVariant() { }

void setupUSB() __attribute__((weak));
void setupUSB() { }

int main(void)
{
  init();
  initVariant();

#if defined(SERIAL_CDC)
  PluggableUSBD().begin();
  _SerialUSB.begin(115200);
#endif

  setup();

  for (;;) {
    loop();
    if (arduino::serialEventRun) arduino::serialEventRun();
  }

  return 0;
}
```

```cpp
#include <Arduino.h>
#include <USB/PluggableUSBSerial.h>

// Declared weak in Arduino.h to allow user redefinitions.
int atexit(void (* /*func*/ )()) { return 0; }

// Weak empty variant initialization function.
// May be redefined by variant files.
void initVariant() __attribute__((weak));
void initVariant() { }

void setupUSB() __attribute__((weak));
void setupUSB() { }

int main(void)
{
  init();
  initVariant();

#if defined(SERIAL_CDC)
  PluggableUSBD().begin();
  _SerialUSB.begin(115200);
#endif

  setup();

  for (;;) {
    loop();
    if (arduino::serialEventRun) arduino::serialEventRun();
  }

  return 0;
}
```

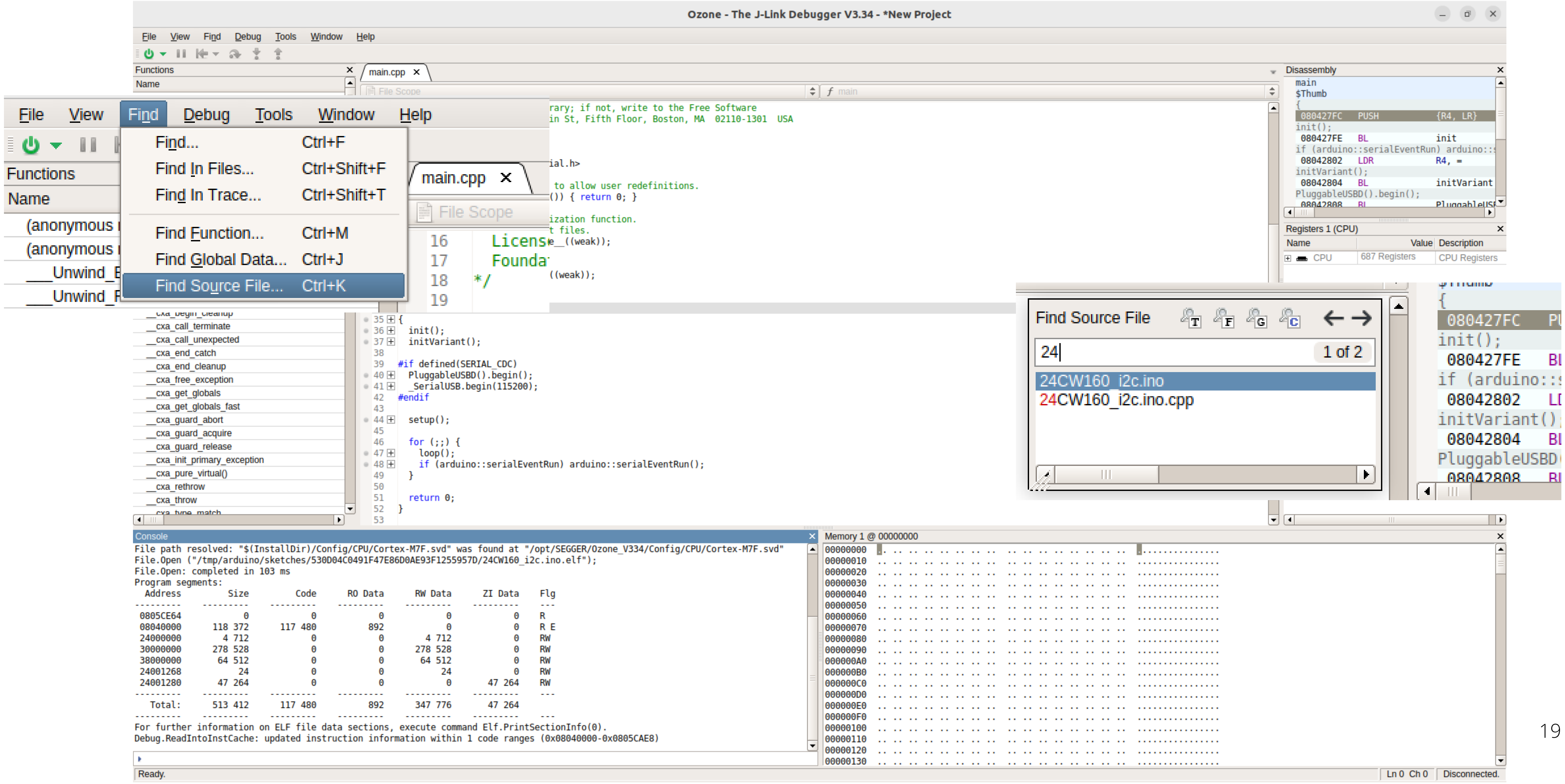
Console:

```
File path resolved: "$(InstallDir)/Config/CPU/Cortex-M7F.svd" was found at "/opt/SEGGER/Ozone_V334/Config/CPU/Cortex-M7F.svd"
File.Open ("/tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf");
File.Open: completed in 103 ms
Program segments:
  Address        Size        Code     RO Data     RW Data     ZI Data    Flg
 ---------   ---------   ---------   ---------   ---------   ---------   ---
  0805CE64           0           0           0           0           0    R
  08040000     118 372     117 480         892           0           0    R E
  24000000       4 712           0           0       4 712           0    RW
  30000000     278 528           0           0     278 528           0    RW
  38000000      64 512           0           0      64 512           0    RW
  24001268          24           0           0          24           0    RW
  24001280      47 264           0           0           0      47 264    RW
 ---------   ---------   ---------   ---------   ---------   ---------   ---
    Total:     513 412     117 480         892     347 776      47 264
 ---------   ---------   ---------   ---------   ---------   ---------   ---
For further information on ELF file data sections, execute command Elf.PrintSectionInfo(0).
Debug.ReadIntoInstCache: updated instruction information within 1 code ranges (0x08040000-0x0805CAE8)
```

Memory 1 @ 00000000:

```
00000000  .. .. .. .. .. .. .. ..  .. ..
00000010  .. .. .. .. .. .. .. ..  .. ..
00000020  .. .. .. .. .. .. .. ..  .. ..
00000030  .. .. .. .. .. .. .. ..  .. ..
00000040  .. .. .. .. .. .. .. ..  .. ..
00000050  .. .. .. .. .. .. .. ..  .. ..
00000060  .. .. .. .. .. .. .. ..  .. ..
00000070  .. .. .. .. .. .. .. ..  .. ..
00000080  .. .. .. .. .. .. .. ..  .. ..
00000090  .. .. .. .. .. .. .. ..  .. ..
000000A0  .. .. .. .. .. .. .. ..  .. ..
000000B0  .. .. .. .. .. .. .. ..  .. ..
000000C0  .. .. .. .. .. .. .. ..  .. ..
000000D0  .. .. .. .. .. .. .. ..  .. ..
000000E0  .. .. .. .. .. .. .. ..  .. ..
000000F0  .. .. .. .. .. .. .. ..  .. ..
00000100  .. .. .. .. .. .. .. ..  .. ..
00000110  .. .. .. .. .. .. .. ..  .. .. .. .. .. .. .. ..  ................
00000120  .. .. .. .. .. .. .. ..  .. .. .. .. .. .. .. ..  ................
00000130  .. .. .. .. .. .. .. ..  .. .. .. .. .. .. .. ..  ................
```

Find the Sketch Source



```
{
  init();
  initVariant();

#if defined(SERIAL_CDC)
  PluggableUSBD().begin();
  _SerialUSB.begin(115200);
#endif

  setup();

  for (;;) {
    loop();
    if (arduino::serialEventRun) arduino::serialEventRun();
  }

  return 0;
}
```

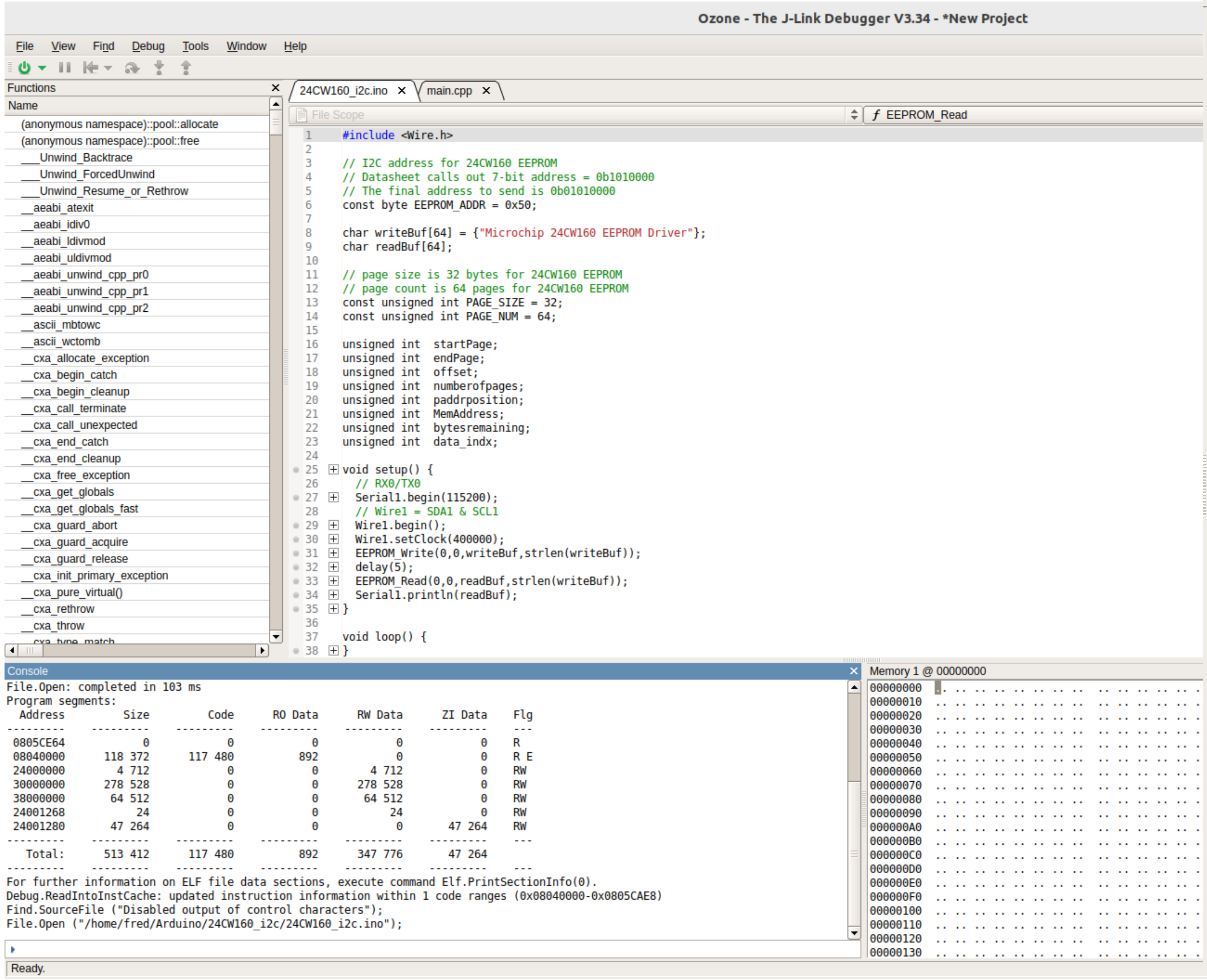
```
File path resolved: "$(InstallDir)/Config/CPU/Cortex-M7F.svd" was found at "/opt/SEGGER/Ozone_V334/Config/CPU/Cortex-M7F.svd"
File.Open ("/tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf");
File.Open: completed in 103 ms
Program segments:
  Address       Size       Code    RO Data    RW Data    ZI Data   Flg
---------  ---------  ---------  ---------  ---------  ---------   ---
 0805CE64          0          0          0          0          0   R
 08040000    118 372    117 480        892          0          0   R E
 24000000      4 712          0          0      4 712          0   RW
 30000000    278 528          0          0    278 528          0   RW
 38000000     64 512          0          0     64 512          0   RW
 24001268         24          0          0         24          0   RW
 24001280     47 264          0          0          0     47 264   RW
---------  ---------  ---------  ---------  ---------  ---------   ---
   Total:    513 412    117 480        892    347 776     47 264
---------  ---------  ---------  ---------  ---------  ---------   ---
For further information on ELF file data sections, execute command Elf.PrintSectionInfo(0).
Debug.ReadIntoInstCache: updated instruction information within 1 code ranges (0x08040000-0x0805CAE8)
```

Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

Display the Sketch Source


Ozone - The J-Link Debugger V3.34 - *New Project

File View Find Debug Tools Window Help

Functions

Name

(anonymous namespace)::pool::allocate
(anonymous namespace)::pool::free
___Unwind_Backtrace
___Unwind_ForcedUnwind
___Unwind_Resume_or_Rethrow
__aeabi_atexit
__aeabi_idiv0
__aeabi_ldivmod
__aeabi_uldivmod
__aeabi_unwind_cpp_pr0
__aeabi_unwind_cpp_pr1
__aeabi_unwind_cpp_pr2
__ascii_mbtowc
__ascii_wctomb
__cxa_allocate_exception
__cxa_begin_catch
__cxa_begin_cleanup
__cxa_call_terminate
__cxa_call_unexpected
__cxa_end_catch
__cxa_end_cleanup
__cxa_free_exception
__cxa_get_globals
__cxa_get_globals_fast
__cxa_guard_abort
__cxa_guard_acquire
__cxa_guard_release
__cxa_init_primary_exception
__cxa_pure_virtual()
__cxa_rethrow
__cxa_throw

24CW160_i2c.ino ✕ main.cpp ✕

File Scope ƒ EEPROM_Read

```
#include <Wire.h>

// I2C address for 24CW160 EEPROM
// Datasheet calls out 7-bit address = 0b1010000
// The final address to send is 0b01010000
const byte EEPROM_ADDR = 0x50;

char writeBuf[64] = {"Microchip 24CW160 EEPROM Driver"};
char readBuf[64];

// page size is 32 bytes for 24CW160 EEPROM
// page count is 64 pages for 24CW160 EEPROM
const unsigned int PAGE_SIZE = 32;
const unsigned int PAGE_NUM = 64;

unsigned int  startPage;
unsigned int  endPage;
unsigned int  offset;
unsigned int  numberofpages;
unsigned int  paddrposition;
unsigned int  MemAddress;
unsigned int  bytesremaining;
unsigned int  data_indx;

void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}
```

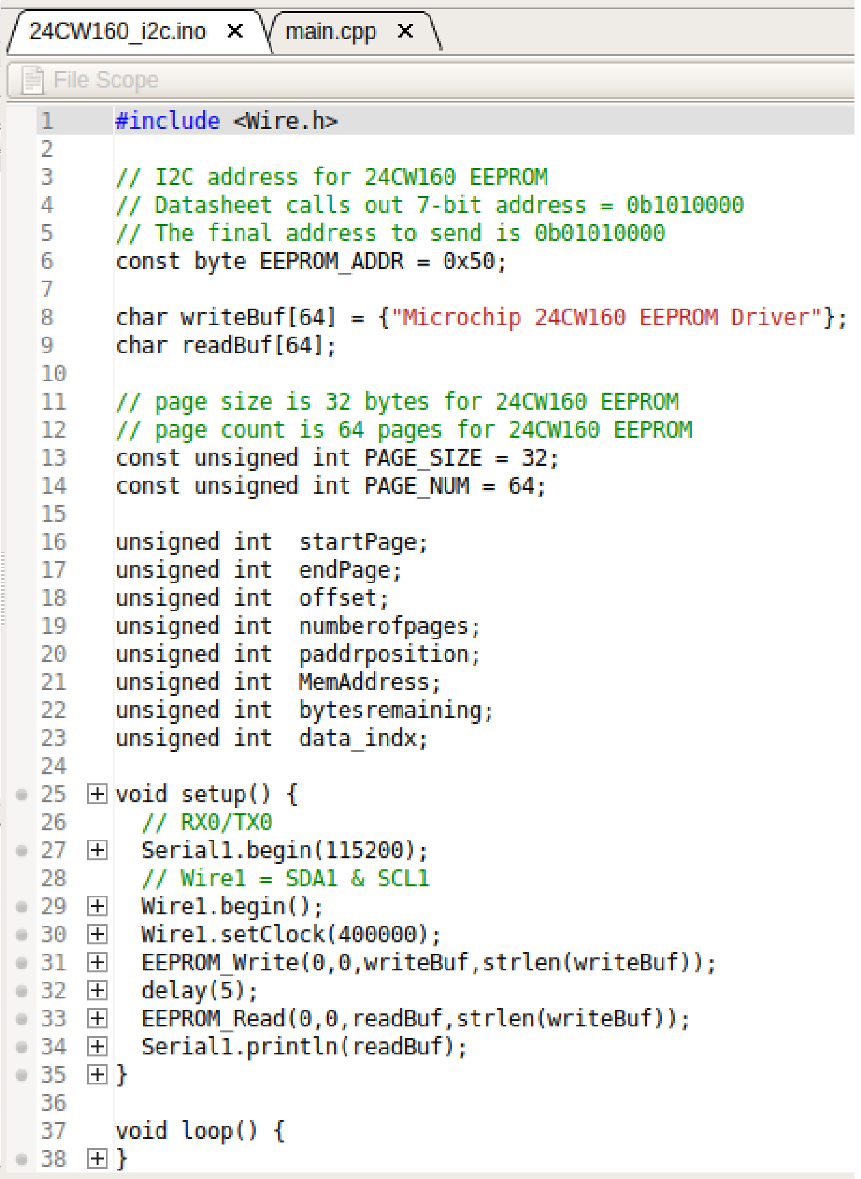
Console

```
File.Open: completed in 103 ms
Program segments:
  Address         Size         Code       RO Data       RW Data       ZI Data    Flg
---------    ---------    ---------     ---------     ---------     ---------    ---
 0805CE64            0            0             0             0             0    R
 08040000      118 372      117 480           892             0             0    R E
 24000000        4 712            0             0         4 712             0    RW
 30000000      278 528            0             0       278 528             0    RW
 38000000       64 512            0             0        64 512             0    RW
 24001268           24            0             0            24             0    RW
 24001280       47 264            0             0             0        47 264    RW
---------    ---------    ---------     ---------     ---------     ---------    ---
   Total:      513 412      117 480           892       347 776        47 264
---------    ---------    ---------     ---------     ---------     ---------    ---
For further information on ELF file data sections, execute command Elf.PrintSectionInfo(0).
Debug.ReadIntoInstCache: updated instruction information within 1 code ranges (0x08040000-0x0805CAE8)
Find.SourceFile ("Disabled output of control characters");
File.Open ("/home/fred/Arduino/24CW160_i2c/24CW160_i2c.ino");
```

Memory 1 @ 00000000

```
00000000
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100
00000110
00000120
00000130
```

Ready.


24CW160_i2c.ino ✕ main.cpp ✕

File Scope

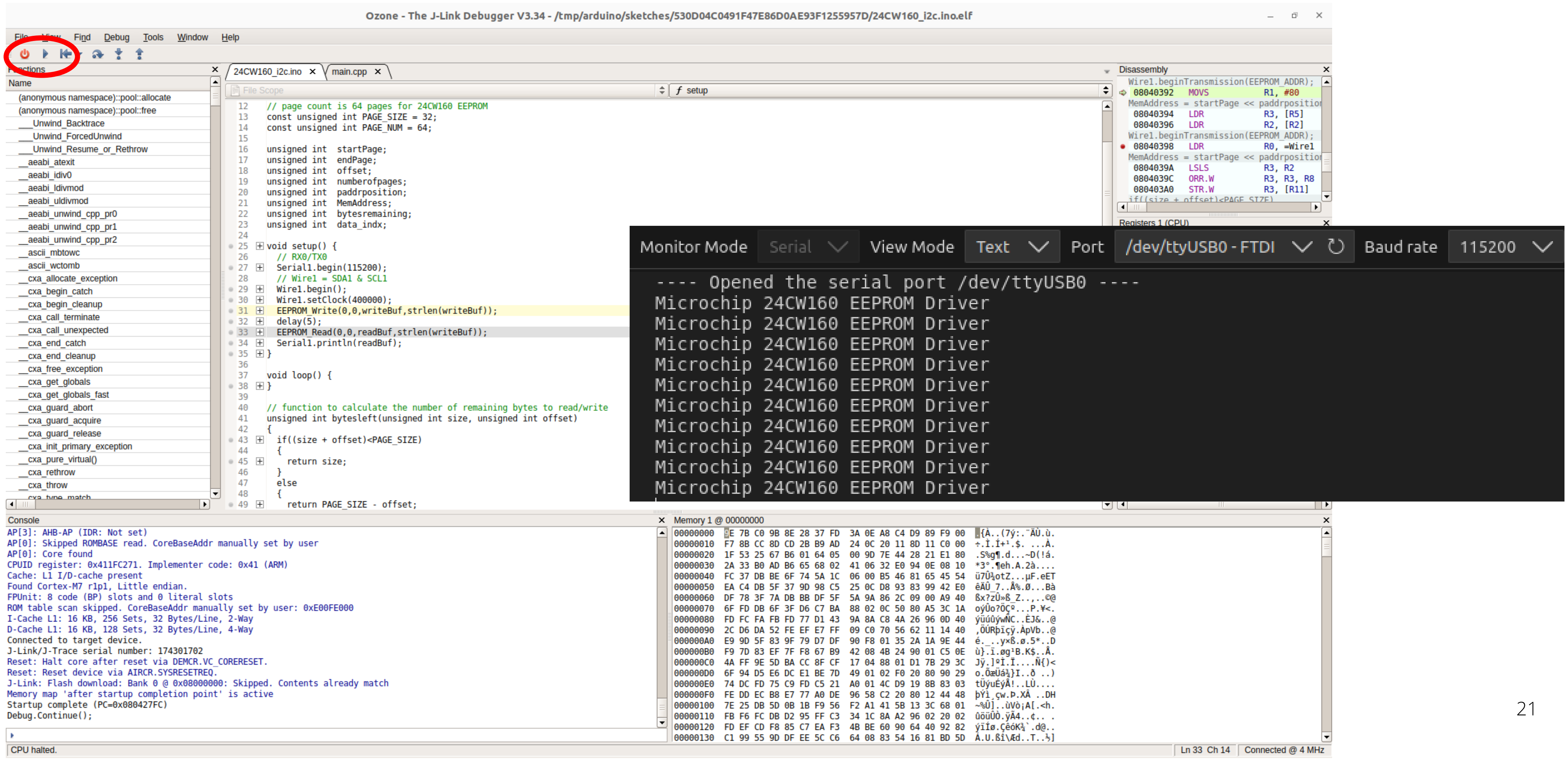
```
#include <Wire.h>

// I2C address for 24CW160 EEPROM
// Datasheet calls out 7-bit address = 0b1010000
// The final address to send is 0b01010000
const byte EEPROM_ADDR = 0x50;

char writeBuf[64] = {"Microchip 24CW160 EEPROM Driver"};
char readBuf[64];

// page size is 32 bytes for 24CW160 EEPROM
// page count is 64 pages for 24CW160 EEPROM
const unsigned int PAGE_SIZE = 32;
const unsigned int PAGE_NUM = 64;

unsigned int  startPage;
unsigned int  endPage;
unsigned int  offset;
unsigned int  numberofpages;
unsigned int  paddrposition;
unsigned int  MemAddress;
unsigned int  bytesremaining;
unsigned int  data_indx;

void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}
```

Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

It Works!



Ozone - The J-Link Debugger V3.34 - /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf

```
(anonymous namespace)::pool::allocate
(anonymous namespace)::pool::free
__Unwind_Backtrace
__Unwind_ForcedUnwind
__Unwind_Resume_or_Rethrow
__aeabi_atexit
__aeabi_idiv0
__aeabi_ldivmod
__aeabi_uldivmod
__aeabi_unwind_cpp_pr0
__aeabi_unwind_cpp_pr1
__aeabi_unwind_cpp_pr2
__ascii_mbtowc
__ascii_wctomb
__cxa_allocate_exception
__cxa_begin_catch
__cxa_begin_cleanup
__cxa_call_terminate
__cxa_call_unexpected
__cxa_end_catch
__cxa_end_cleanup
__cxa_free_exception
__cxa_get_globals
__cxa_get_globals_fast
__cxa_guard_abort
__cxa_guard_acquire
__cxa_guard_release
__cxa_init_primary_exception
__cxa_pure_virtual()
__cxa_rethrow
__cxa_throw
```

```
// page count is 64 pages for 24CW160 EEPROM
const unsigned int PAGE_SIZE = 32;
const unsigned int PAGE_NUM = 64;

unsigned int  startPage;
unsigned int  endPage;
unsigned int  offset;
unsigned int  numberofpages;
unsigned int  paddrposition;
unsigned int  MemAddress;
unsigned int  bytesremaining;
unsigned int  data_indx;

void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}

// function to calculate the number of remaining bytes to read/write
unsigned int bytesleft(unsigned int size, unsigned int offset)
{
  if((size + offset)<PAGE_SIZE)
  {
    return size;
  }
  else
  {
    return PAGE_SIZE - offset;
```

Disassembly

```
Wire1.beginTransmission(EEPROM_ADDR);
 08040392   MOVS      R1, #80
MemAddress = startPage << paddrposition
 08040394   LDR       R3, [R5]
 08040396   LDR       R2, [R2]
Wire1.beginTransmission(EEPROM_ADDR);
 08040398   LDR       R0, =Wire1
MemAddress = startPage << paddrposition
 0804039A   LSLS      R3, R2
 0804039C   ORR.W     R3, R3, R8
 080403A0   STR.W     R3, [R11]
```

Monitor Mode: Serial | View Mode: Text | Port: /dev/ttyUSB0 - FTDI | Baud rate: 115200

```
---- Opened the serial port /dev/ttyUSB0 ----
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
Microchip 24CW160 EEPROM Driver
```

Console

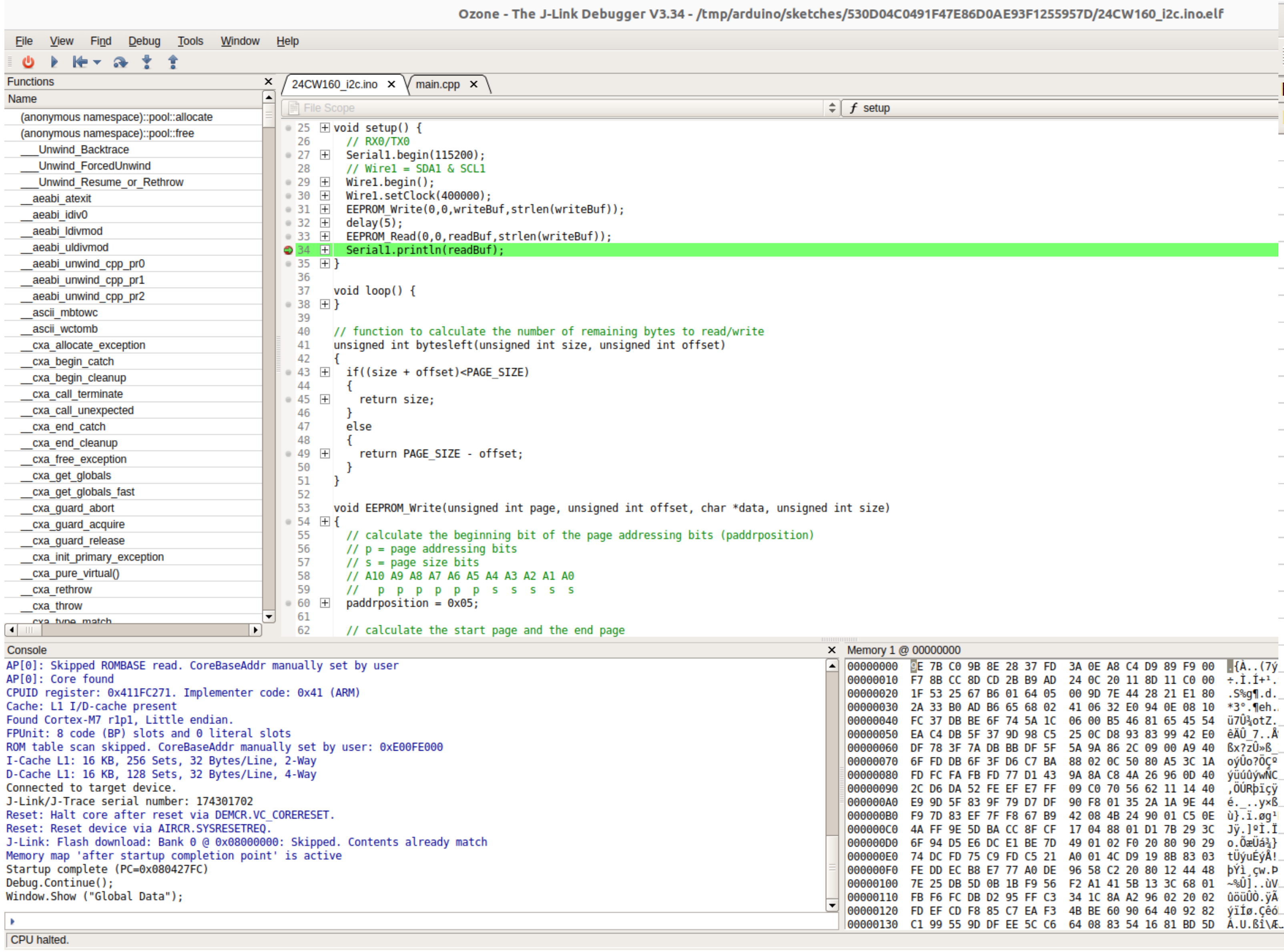
```
AP[3]: AHB-AP (IDR: Not set)
AP[0]: Skipped ROMBASE read. CoreBaseAddr manually set by user
AP[0]: Core found
CPUID register: 0x411FC271. Implementer code: 0x41 (ARM)
Cache: L1 I/D-cache present
Found Cortex-M7 r1p1, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
ROM table scan skipped. CoreBaseAddr manually set by user: 0xE00FE000
I-Cache L1: 16 KB, 256 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 16 KB, 128 Sets, 32 Bytes/Line, 4-Way
Connected to target device.
J-Link/J-Trace serial number: 174301702
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link: Flash download: Bank 0 @ 0x08000000: Skipped. Contents already match
Memory map 'after startup completion point' is active
Startup complete (PC=0x080427FC)
Debug.Continue();
```

Memory 1 @ 00000000

```
00000000  9E 7B C0 9B 8E 28 37 FD  3A 0E A8 C4 D9 89 F9 00  .{À..(7ý:.¨ÄÙ.ù.
00000010  F7 8B CC 8D CD 2B B9 AD  24 0C 20 11 8D 11 C0 00  ÷.Ì.Í+¹.$. ...À.
00000020  1F 53 25 67 B6 01 64 05  00 9D 7E 44 28 21 E1 80  .S%g¶.d...~D(!á.
00000030  2A 33 B0 AD B6 65 68 02  41 06 32 E0 94 0E 08 10  *3°.¶eh.A.2à....
00000040  FC 37 DB BE 6F 74 5A 1C  06 00 B5 46 81 65 45 54  ü7Û¾otZ...µF.eET
00000050  EA C4 DB 5F 37 9D 98 C5  25 0C D8 93 83 99 42 E0  êÄÛ_7..Å%.Ø...Bà
00000060  DF 78 3F 7A DB BB DF 5F  5A 9A 86 2C 09 00 A9 40  ßx?zÛ»ß_Z..,..©@
00000070  6F FD DB 6F 3F D6 C7 BA  88 02 0C 50 80 A5 3C 1A  oýÛo?ÖÇº...P.¥<.
00000080  FD FC FA FB FD 77 D1 43  9A 8A C8 4A 26 96 0D 40  ýüúûýwÑC..ÈJ&..@
00000090  2C D6 DA 52 FE EF E7 FF  09 C0 70 56 62 11 14 40  ,ÖÚRþïçÿ.ÀpVb..@
000000A0  E9 9D 5F 83 9F 79 D7 DF  90 F8 01 35 2A 1A 9E 44  é._..y×ß.ø.5*..D
000000B0  F9 7D 83 EF 7F F8 67 B9  42 08 4B 24 90 01 C5 0E  ù}.ï.øg¹B.K$..Å.
000000C0  4A FF 9E 5D BA CC 8F CF  17 04 88 01 D1 7B 29 3C  Jÿ.]ºÌ.Ï....Ñ{)<
000000D0  6F 94 D5 E6 DC E1 BE 7D  49 01 02 F0 20 80 90 29  o.ÕæÜá¾}I..ð ..)
000000E0  74 DC FD 75 C9 FD C5 21  A0 01 4C D9 19 8B 83 03  tÜýuÉýÅ!..LÙ....
000000F0  FE DD EC B8 E7 77 A0 DE  96 58 C2 20 80 12 44 48  þÝì¸çw.Þ.XÂ ..DH
00000100  7E 25 DB 5D 0B 1B F9 56  F2 A1 41 5B 13 3C 68 01  ~%Û]..ùVò¡A[.<h.
00000110  FB F6 FC DB D2 95 FF C3  34 1C 8A A2 96 02 20 02  ûöüÛÒ.ÿÃ4..¢.. .
00000120  FD EF CD F8 85 C7 EA F3  4B BE 60 90 64 40 92 82  ýïÍø.ÇêóK¾`.d@..
00000130  C1 99 55 9D DF EE 5C C6  64 08 83 54 16 81 BD 5D  Á.U.ßî\Æd..T..½]
```

CPU halted. | Ln 33 Ch 14 | Connected @ 4 MHz

Debug Views

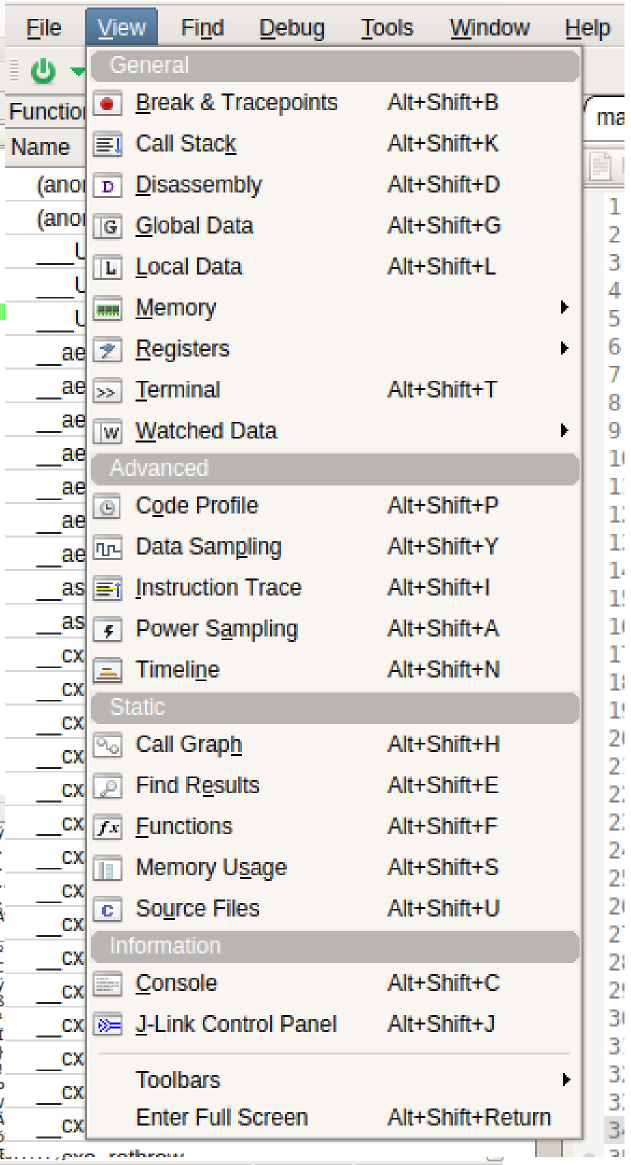

Ozone - The J-Link Debugger V3.34 - /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf
File View Find Debug Tools Window Help
Functions
Name
24CW160_i2c.ino
main.cpp
File Scope
setup
Console
Memory 1 @ 00000000
CPU halted.
Ln 34 Ch 28
Connected @ 4 MHz

```
(anonymous namespace)::pool::allocate
(anonymous namespace)::pool::free
___Unwind_Backtrace
___Unwind_ForcedUnwind
___Unwind_Resume_or_Rethrow
__aeabi_atexit
__aeabi_idiv0
__aeabi_ldivmod
__aeabi_uldivmod
__aeabi_unwind_cpp_pr0
__aeabi_unwind_cpp_pr1
__aeabi_unwind_cpp_pr2
__ascii_mbtowc
__ascii_wctomb
__cxa_allocate_exception
__cxa_begin_catch
__cxa_begin_cleanup
__cxa_call_terminate
__cxa_call_unexpected
__cxa_end_catch
__cxa_end_cleanup
__cxa_free_exception
__cxa_get_globals
__cxa_get_globals_fast
__cxa_guard_abort
__cxa_guard_acquire
__cxa_guard_release
__cxa_init_primary_exception
__cxa_pure_virtual()
__cxa_rethrow
__cxa_throw
__cxa_type_match
```

```
void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}

// function to calculate the number of remaining bytes to read/write
unsigned int bytesleft(unsigned int size, unsigned int offset)
{
  if((size + offset)<PAGE_SIZE)
  {
    return size;
  }
  else
  {
    return PAGE_SIZE - offset;
  }
}

void EEPROM_Write(unsigned int page, unsigned int offset, char *data, unsigned int size)
{
  // calculate the beginning bit of the page addressing bits (paddrposition)
  // p = page addressing bits
  // s = page size bits
  // A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
  //   p  p  p  p  p  p  s  s  s  s  s
  paddrposition = 0x05;

  // calculate the start page and the end page
```

```
AP[0]: Skipped ROMBASE read. CoreBaseAddr manually set by user
AP[0]: Core found
CPUID register: 0x411FC271. Implementer code: 0x41 (ARM)
Cache: L1 I/D-cache present
Found Cortex-M7 r1p1, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
ROM table scan skipped. CoreBaseAddr manually set by user: 0xE00FE000
I-Cache L1: 16 KB, 256 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 16 KB, 128 Sets, 32 Bytes/Line, 4-Way
Connected to target device.
J-Link/J-Trace serial number: 174301702
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link: Flash download: Bank 0 @ 0x08000000: Skipped. Contents already match
Memory map 'after startup completion point' is active
Startup complete (PC=0x080427FC)
Debug.Continue();
Window.Show ("Global Data");
```

```
00000000  9E 7B C0 9B 8E 28 37 FD  3A 0E A8 C4 D9 89 F9 00  .{À..(7ý
00000010  F7 8B CC 8D CD 2B B9 AD  24 0C 20 11 8D 11 C0 00  ÷.Ì.Í+¹.
00000020  1F 53 25 67 B6 01 64 05  00 9D 7E 44 28 21 E1 80  .S%g¶.d.
00000030  2A 33 B0 AD B6 65 68 02  41 06 32 E0 94 0E 08 10  *3°.¶eh.
00000040  FC 37 DB BE 6F 74 5A 1C  06 00 B5 46 81 65 45 54  ü7Û¾otZ.
00000050  EA C4 DB 5F 37 9D 98 C5  25 0C D8 93 83 99 42 E0  êÄÛ_7..Å
00000060  DF 78 3F 7A DB BB DF 5F  5A 9A 86 2C 09 00 A9 40  ßx?zÛ»ß_
00000070  6F FD DB 6F 3F D6 C7 BA  88 02 0C 50 80 A5 3C 1A  oýÛo?ÖÇº
00000080  FD FC FA FB FD 77 D1 43  9A 8A C8 4A 26 96 0D 40  ýüúûýwÑC
00000090  2C D6 DA 52 FE EF E7 FF  09 C0 70 56 62 11 14 40  ,ÖÚRþïçÿ
000000A0  E9 9D 5F 83 9F 79 D7 DF  90 F8 01 35 2A 1A 9E 44  é._..y×ß
000000B0  F9 7D 83 EF 7F F8 67 B9  42 08 4B 24 90 01 C5 0E  ù}.ï.øg¹
000000C0  4A FF 9E 5D BA CC 8F CF  17 04 88 01 D1 7B 29 3C  Jÿ.]ºÌ.Ï
000000D0  6F 94 D5 E6 DC E1 BE 7D  49 01 02 F0 20 80 90 29  o.ÕæÜá¾}
000000E0  74 DC FD 75 C9 FD C5 21  A0 01 4C D9 19 8B 83 03  tÜýuÉýÅ!
000000F0  FE DD EC B8 E7 77 A0 DE  96 58 C2 20 80 12 44 48  þÝì¸çw.Þ
00000100  7E 25 DB 5D 0B 1B F9 56  F2 A1 41 5B 13 3C 68 01  ~%Û]..ùV
00000110  FB F6 FC DB D2 95 FF C3  34 1C 8A A2 96 02 20 02  ûöüÛÒ.ÿÃ
00000120  FD EF CD F8 85 C7 EA F3  4B BE 60 90 64 40 92 82  ýïÍø.Çêó
00000130  C1 99 55 9D DF EE 5C C6  64 08 83 54 16 81 BD 5D  Á.U.ßî\Æ
```

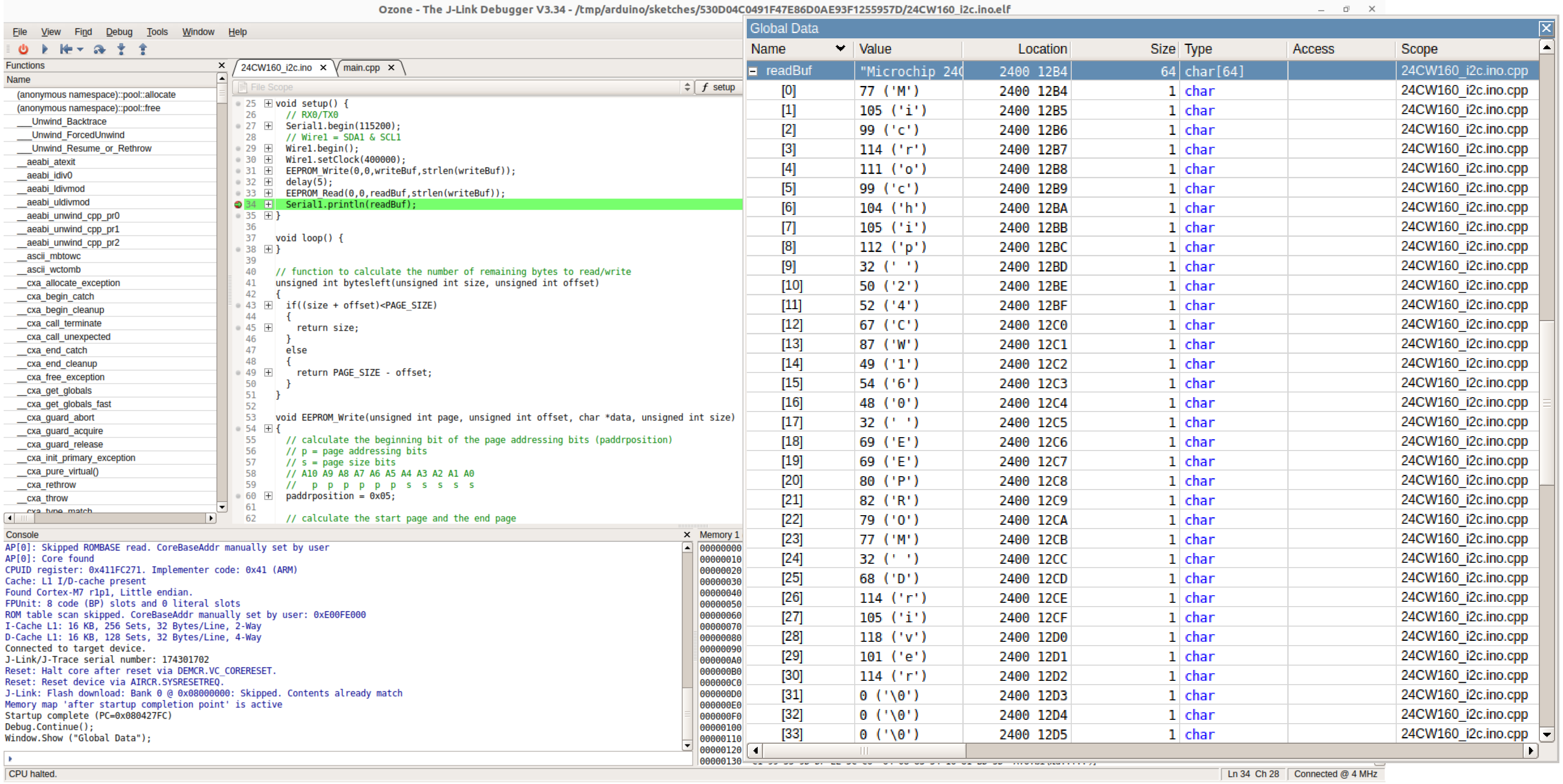


Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

Show Global Data



Ozone - The J-Link Debugger V3.34 - /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf

```
(anonymous namespace)::pool::allocate
(anonymous namespace)::pool::free
___Unwind_Backtrace
___Unwind_ForcedUnwind
___Unwind_Resume_or_Rethrow
__aeabi_atexit
__aeabi_idiv0
__aeabi_ldivmod
__aeabi_uldivmod
__aeabi_unwind_cpp_pr0
__aeabi_unwind_cpp_pr1
__aeabi_unwind_cpp_pr2
__ascii_mbtowc
__ascii_wctomb
__cxa_allocate_exception
__cxa_begin_catch
__cxa_begin_cleanup
__cxa_call_terminate
__cxa_call_unexpected
__cxa_end_catch
__cxa_end_cleanup
__cxa_free_exception
__cxa_get_globals
__cxa_get_globals_fast
__cxa_guard_abort
__cxa_guard_acquire
__cxa_guard_release
__cxa_init_primary_exception
__cxa_pure_virtual()
__cxa_rethrow
__cxa_throw
```

```
void setup() {
  // RX0/TX0
  Serial1.begin(115200);
  // Wire1 = SDA1 & SCL1
  Wire1.begin();
  Wire1.setClock(400000);
  EEPROM_Write(0,0,writeBuf,strlen(writeBuf));
  delay(5);
  EEPROM_Read(0,0,readBuf,strlen(writeBuf));
  Serial1.println(readBuf);
}

void loop() {
}

// function to calculate the number of remaining bytes to read/write
unsigned int bytesleft(unsigned int size, unsigned int offset)
{
  if((size + offset)<PAGE_SIZE)
  {
    return size;
  }
  else
  {
    return PAGE_SIZE - offset;
  }
}

void EEPROM_Write(unsigned int page, unsigned int offset, char *data, unsigned int size)
{
  // calculate the beginning bit of the page addressing bits (paddrposition)
  // p = page addressing bits
  // s = page size bits
  // A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
  //   p  p  p  p  p  p  s  s  s  s  s
  paddrposition = 0x05;

  // calculate the start page and the end page
```

Console

```
AP[0]: Skipped ROMBASE read. CoreBaseAddr manually set by user
AP[0]: Core found
CPUID register: 0x411FC271. Implementer code: 0x41 (ARM)
Cache: L1 I/D-cache present
Found Cortex-M7 r1p1, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
ROM table scan skipped. CoreBaseAddr manually set by user: 0xE00FE000
I-Cache L1: 16 KB, 256 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 16 KB, 128 Sets, 32 Bytes/Line, 4-Way
Connected to target device.
J-Link/J-Trace serial number: 174301702
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link: Flash download: Bank 0 @ 0x08000000: Skipped. Contents already match
Memory map 'after startup completion point' is active
Startup complete (PC=0x080427FC)
Debug.Continue();
Window.Show ("Global Data");
```

```
00000000
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
000000A0
000000B0
000000C0
000000D0
000000E0
000000F0
00000100
00000110
00000120
00000130
```

CPU halted.   Ln 34 Ch 28   Connected @ 4 MHz

Global Data

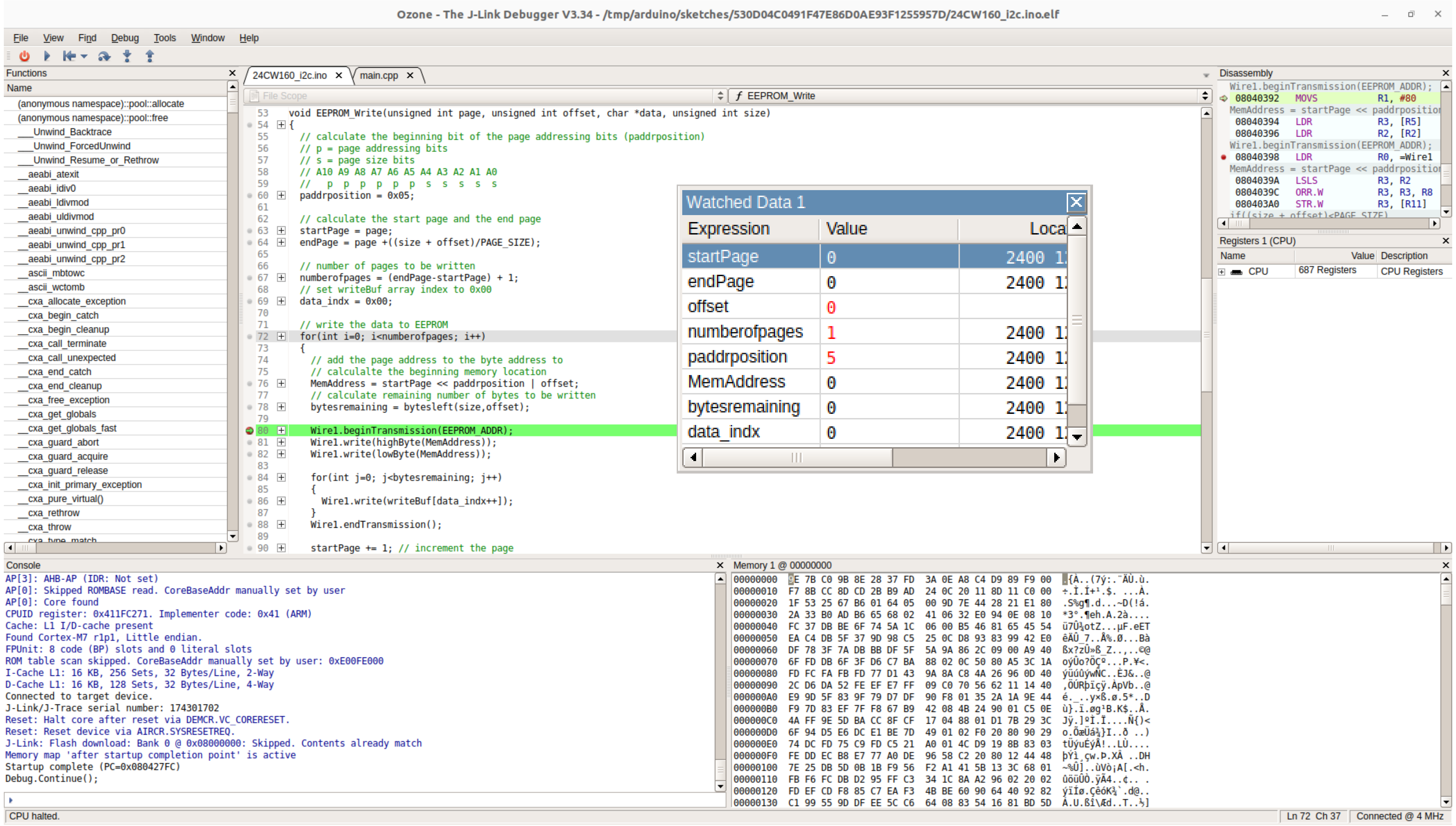
Name	Value	Location	Size	Type	Access	Scope
readBuf	"Microchip 24C	2400 12B4	64	char[64]		24CW160_i2c.ino.cpp
[0]	77 ('M')	2400 12B4	1	char		24CW160_i2c.ino.cpp
[1]	105 ('i')	2400 12B5	1	char		24CW160_i2c.ino.cpp
[2]	99 ('c')	2400 12B6	1	char		24CW160_i2c.ino.cpp
[3]	114 ('r')	2400 12B7	1	char		24CW160_i2c.ino.cpp
[4]	111 ('o')	2400 12B8	1	char		24CW160_i2c.ino.cpp
[5]	99 ('c')	2400 12B9	1	char		24CW160_i2c.ino.cpp
[6]	104 ('h')	2400 12BA	1	char		24CW160_i2c.ino.cpp
[7]	105 ('i')	2400 12BB	1	char		24CW160_i2c.ino.cpp
[8]	112 ('p')	2400 12BC	1	char		24CW160_i2c.ino.cpp
[9]	32 (' ')	2400 12BD	1	char		24CW160_i2c.ino.cpp
[10]	50 ('2')	2400 12BE	1	char		24CW160_i2c.ino.cpp
[11]	52 ('4')	2400 12BF	1	char		24CW160_i2c.ino.cpp
[12]	67 ('C')	2400 12C0	1	char		24CW160_i2c.ino.cpp
[13]	87 ('W')	2400 12C1	1	char		24CW160_i2c.ino.cpp
[14]	49 ('1')	2400 12C2	1	char		24CW160_i2c.ino.cpp
[15]	54 ('6')	2400 12C3	1	char		24CW160_i2c.ino.cpp
[16]	48 ('0')	2400 12C4	1	char		24CW160_i2c.ino.cpp
[17]	32 (' ')	2400 12C5	1	char		24CW160_i2c.ino.cpp
[18]	69 ('E')	2400 12C6	1	char		24CW160_i2c.ino.cpp
[19]	69 ('E')	2400 12C7	1	char		24CW160_i2c.ino.cpp
[20]	80 ('P')	2400 12C8	1	char		24CW160_i2c.ino.cpp
[21]	82 ('R')	2400 12C9	1	char		24CW160_i2c.ino.cpp
[22]	79 ('O')	2400 12CA	1	char		24CW160_i2c.ino.cpp
[23]	77 ('M')	2400 12CB	1	char		24CW160_i2c.ino.cpp
[24]	32 (' ')	2400 12CC	1	char		24CW160_i2c.ino.cpp
[25]	68 ('D')	2400 12CD	1	char		24CW160_i2c.ino.cpp
[26]	114 ('r')	2400 12CE	1	char		24CW160_i2c.ino.cpp
[27]	105 ('i')	2400 12CF	1	char		24CW160_i2c.ino.cpp
[28]	118 ('v')	2400 12D0	1	char		24CW160_i2c.ino.cpp
[29]	101 ('e')	2400 12D1	1	char		24CW160_i2c.ino.cpp
[30]	114 ('r')	2400 12D2	1	char		24CW160_i2c.ino.cpp
[31]	0 ('\0')	2400 12D3	1	char		24CW160_i2c.ino.cpp
[32]	0 ('\0')	2400 12D4	1	char		24CW160_i2c.ino.cpp
[33]	0 ('\0')	2400 12D5	1	char		24CW160_i2c.ino.cpp

Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

Show Watch Data



Ozone - The J-Link Debugger V3.34 - /tmp/arduino/sketches/530D04C0491F47E86D0AE93F1255957D/24CW160_i2c.ino.elf

```
void EEPROM_Write(unsigned int page, unsigned int offset, char *data, unsigned int size)
{
  // calculate the beginning bit of the page addressing bits (paddrposition)
  // p = page addressing bits
  // s = page size bits
  // A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
  //   p  p  p  p  p  p  s  s  s  s  s
  paddrposition = 0x05;

  // calculate the start page and the end page
  startPage = page;
  endPage = page +((size + offset)/PAGE_SIZE);

  // number of pages to be written
  numberofpages = (endPage-startPage) + 1;
  // set writeBuf array index to 0x00
  data_indx = 0x00;

  // write the data to EEPROM
  for(int i=0; i<numberofpages; i++)
  {
    // add the page address to the byte address to
    // calculalte the beginning memory location
    MemAddress = startPage << paddrposition | offset;
    // calculate remaining number of bytes to be written
    bytesremaining = bytesleft(size,offset);

    Wire1.beginTransmission(EEPROM_ADDR);
    Wire1.write(highByte(MemAddress));
    Wire1.write(lowByte(MemAddress));

    for(int j=0; j<bytesremaining; j++)
    {
      Wire1.write(writeBuf[data_indx++]);
    }
    Wire1.endTransmission();

    startPage += 1; // increment the page
```

Watched Data 1

Expression	Value	Loca
startPage	0	2400 1
endPage	0	2400 1
offset	0	
numberofpages	1	2400 1
paddrposition	5	2400 1
MemAddress	0	2400 1
bytesremaining	0	2400 1
data_indx	0	2400 1

Disassembly

```
Wire1.beginTransmission(EEPROM_ADDR);
 08040392   MOVS          R1, #80
MemAddress = startPage << paddrpositio
 08040394   LDR           R3, [R5]
 08040396   LDR           R2, [R2]
Wire1.beginTransmission(EEPROM_ADDR);
 08040398   LDR           R0, =Wire1
MemAddress = startPage << paddrpositio
 0804039A   LSLS          R3, R2
 0804039C   ORR.W         R3, R3, R8
 080403A0   STR.W         R3, [R11]
if((size + offset)<PAGE_SIZE)
```

Registers 1 (CPU)

Name	Value	Description
CPU	687 Registers	CPU Registers

Console

```
AP[3]: AHB-AP (IDR: Not set)
AP[0]: Skipped ROMBASE read. CoreBaseAddr manually set by user
AP[0]: Core found
CPUID register: 0x411FC271. Implementer code: 0x41 (ARM)
Cache: L1 I/D-cache present
Found Cortex-M7 r1p1, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
ROM table scan skipped. CoreBaseAddr manually set by user: 0xE00FE000
I-Cache L1: 16 KB, 256 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 16 KB, 128 Sets, 32 Bytes/Line, 4-Way
Connected to target device.
J-Link/J-Trace serial number: 174301702
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link: Flash download: Bank 0 @ 0x08000000: Skipped. Contents already match
Memory map 'after startup completion point' is active
Startup complete (PC=0x080427FC)
Debug.Continue();
```

Memory 1 @ 00000000

```
00000000  9E 7B C0 9B 8E 28 37 FD  3A 0E A8 C4 D9 89 F9 00  .{À..(7ý:.¨ÄÙ.ù.
00000010  F7 8B CC 8D CD 2B B9 AD  24 0C 20 11 8D 11 C0 00  ÷.Ì.Í+¹.$. ...À.
00000020  1F 53 25 67 B6 01 64 05  00 9D 7E 44 28 21 E1 80  .S%g¶.d...~D(!á.
00000030  2A 33 B0 AD B6 65 68 02  41 06 32 E0 94 0E 08 10  *3°.¶eh.A.2à....
00000040  FC 37 DB BE 6F 74 5A 1C  06 00 B5 46 81 65 45 54  ü7Û¾otZ...µF.eET
00000050  EA C4 DB 5F 37 9D 98 C5  25 0C D8 93 83 99 42 E0  êÄÛ_7..Å%.Ø...Bà
00000060  DF 78 3F 7A DB BB DF 5F  5A 9A 86 2C 09 00 A9 40  ßx?zÛ»ß_Z..,..©@
00000070  6F FD DB 6F 3F D6 C7 BA  88 02 0C 50 80 A5 3C 1A  oýÛo?ÖÇº...P.¥<.
00000080  FD FC FA FB FD 77 D1 43  9A 8A C8 4A 26 96 0D 40  ýüúûýwÑC..ÈJ&..@
00000090  2C D6 DA 52 FE EF E7 FF  09 C0 70 56 62 11 14 40  ,ÖÚRþïçÿ.ÀpVb..@
000000A0  E9 9D 5F 83 9F 79 D7 DF  90 F8 01 35 2A 1A 9E 44  é._..y×ß.ø.5*..D
000000B0  F9 7D 83 EF 7F F8 67 B9  42 08 4B 24 90 01 C5 0E  ù}.ï.øg¹B.K$..Å.
000000C0  4A FF 9E 5D BA CC 8F CF  17 04 88 01 D1 7B 29 3C  Jÿ.]ºÌ.Ï....Ñ{)<
000000D0  6F 94 D5 E6 DC E1 BE 7D  49 01 02 F0 20 80 90 29  o.ÕæÜá¾}I..ð ..)
000000E0  74 DC FD 75 C9 FD C5 21  A0 01 4C D9 19 8B 83 03  tÜýuÉýÅ!..LÙ....
000000F0  FE DD EC B8 E7 77 A0 DE  96 58 C2 20 80 12 44 48  þÝì¸çw.Þ.XÂ ..DH
00000100  7E 25 DB 5D 0B 1B F9 56  F2 A1 41 5B 13 3C 68 01  ~%Û]..ùVò¡A[.<h.
00000110  FB F6 FC DB D2 95 FF C3  34 1C 8A A2 96 02 20 02  ûöüÛÒ.ÿÃ4..¢.. .
00000120  FD EF CD F8 85 C7 EA F3  4B BE 60 90 64 40 92 82  ýïÍø.ÇêóK¾`.d@..
00000130  C1 99 55 9D DF EE 5C C6  64 08 83 54 16 81 BD 5D  Á.U.ßî\Æd..T..½]
```

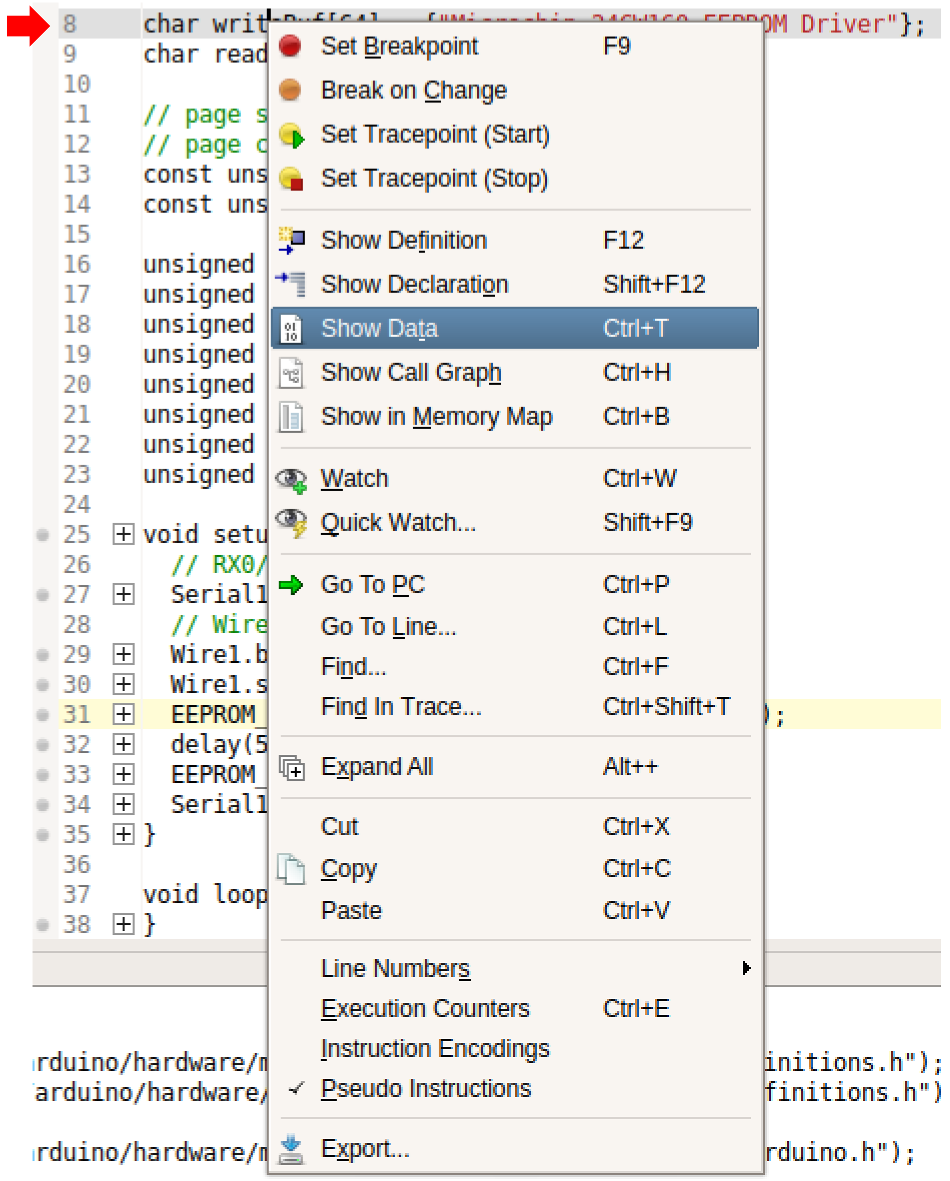
CPU halted.   Ln 72 Ch 37   Connected @ 4 MHz

Arduino Pro Primer

Debugging Arduino Sketches

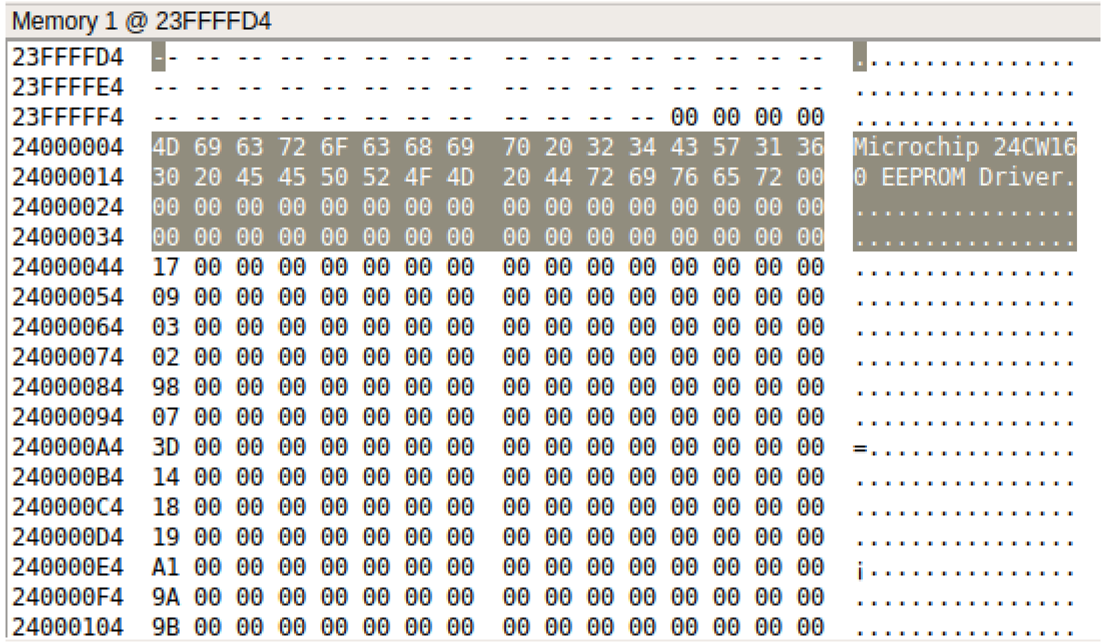
Debug the Microchip 24CW160 EEPROM Driver

Show Variable Data - Array



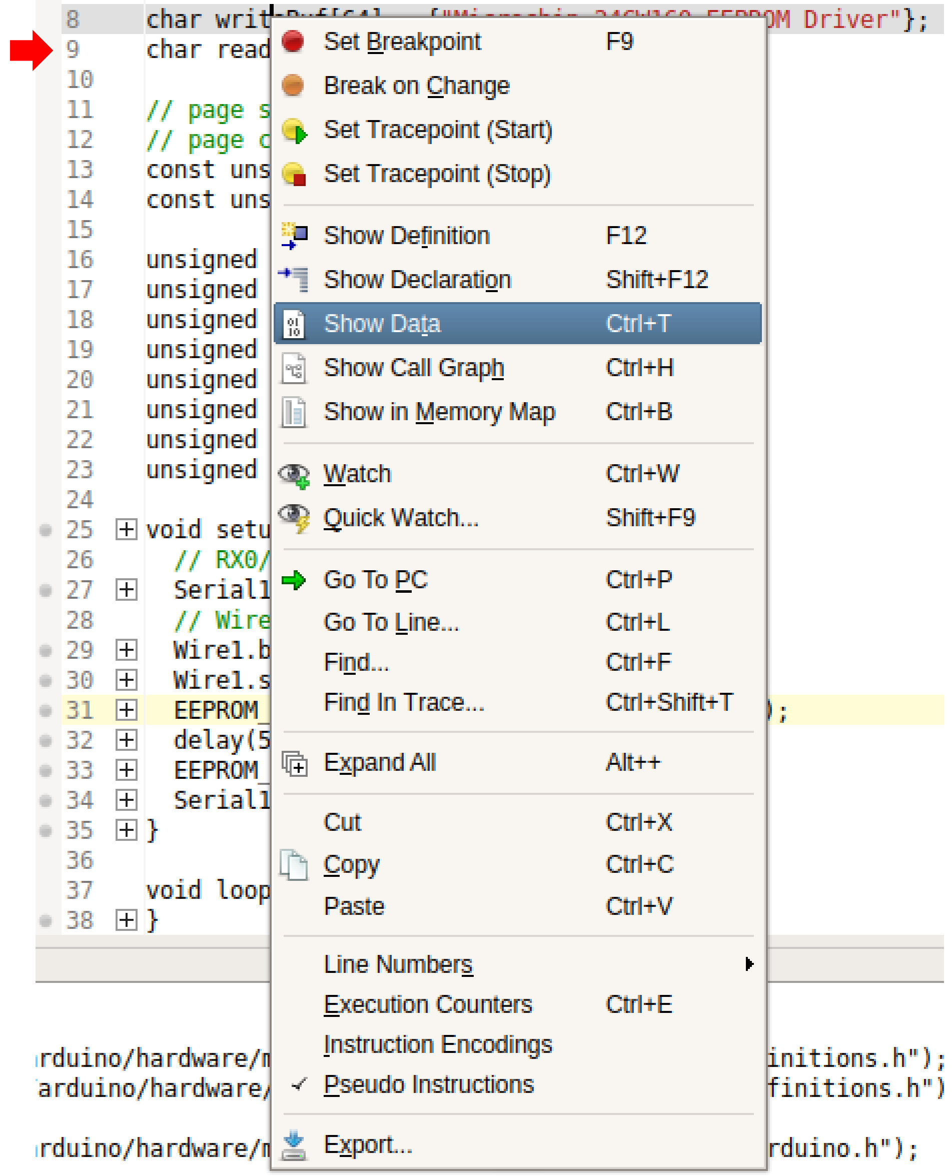
```
char writ               {"Microchip 24CW160 EEPROM Driver"};
char read

// page s
// page c
const uns
const uns

unsigned
unsigned
unsigned
unsigned
unsigned
unsigned
unsigned
unsigned

void setu
  // RX0/
  Serial1
  // Wire
  Wire1.b
  Wire1.s
  EEPROM_                                       );
  delay(5
  EEPROM_
  Serial1
}

void loop
}
```

```
rduino/hardware/m                               initions.h");
arduino/hardware/                               finitions.h")

rduino/hardware/m                               rduino.h");
```

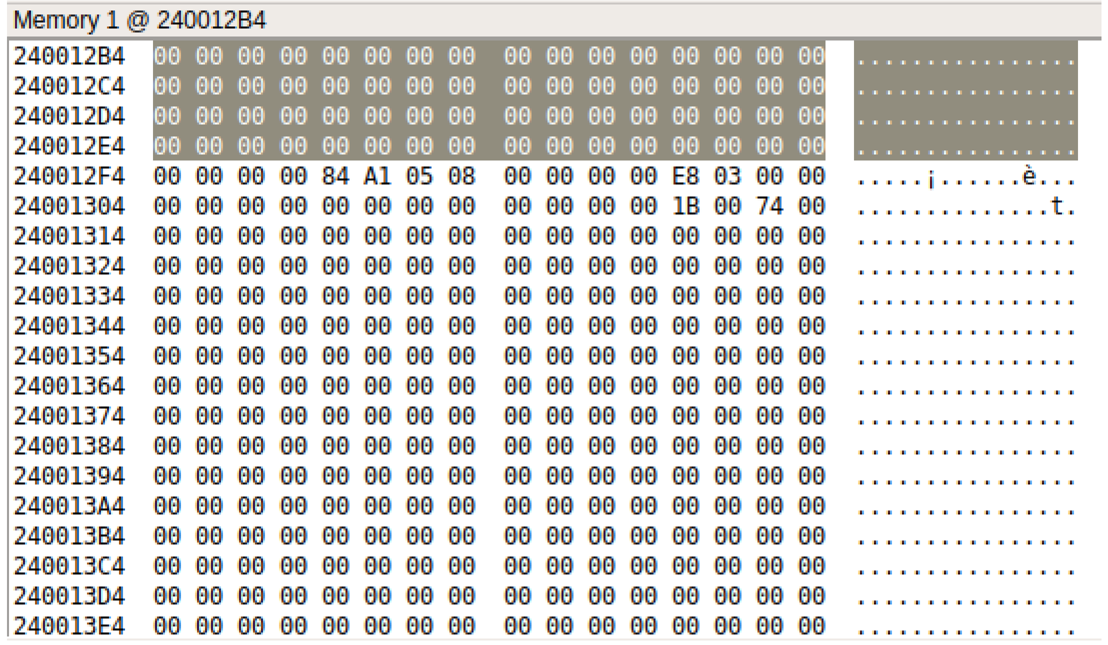

Memory 1 @ 23FFFFD4

```
23FFFFD4  -- -- -- -- -- -- -- --  -- -- -- -- -- -- -- --  ................
23FFFFE4  -- -- -- -- -- -- -- --  -- -- -- -- -- -- -- --  ................
23FFFFF4  -- -- -- -- -- -- -- --  -- -- -- -- 00 00 00 00  ................
24000004  4D 69 63 72 6F 63 68 69  70 20 32 34 43 57 31 36  Microchip 24CW16
24000014  30 20 45 45 50 52 4F 4D  20 44 72 69 76 65 72 00  0 EEPROM Driver.
24000024  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000034  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000044  17 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000054  09 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000064  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000074  02 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000084  98 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000094  07 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240000A4  3D 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  =...............
240000B4  14 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240000C4  18 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240000D4  19 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240000E4  A1 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ¡...............
240000F4  9A 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24000104  9B 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Show Variable Data - Array



```
char writ                         OM Driver"};
char read

// page s
// page c
const uns
const uns

unsigned
unsigned
unsigned
unsigned
unsigned
unsigned
unsigned
unsigned

void setu
  // RX0/
  Serial1
  // Wire
  Wire1.b
  Wire1.s
  EEPROM_                         );
  delay(5
  EEPROM_
  Serial1
}

void loop
}

rduino/hardware/m                 initions.h");
arduino/hardware/                 finitions.h")

rduino/hardware/m                 rduino.h");
```



```
Memory 1 @ 240012B4
240012B4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240012C4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240012D4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240012E4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240012F4  00 00 00 00 84 A1 05 08  00 00 00 00 E8 03 00 00  .....¡......è...
24001304  00 00 00 00 00 00 00 00  00 00 00 00 1B 00 74 00  ..............t.
24001314  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001324  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001334  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001344  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001354  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001364  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001374  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001384  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
24001394  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240013A4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240013B4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240013C4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240013D4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
240013E4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

CEC Continuing Education Center

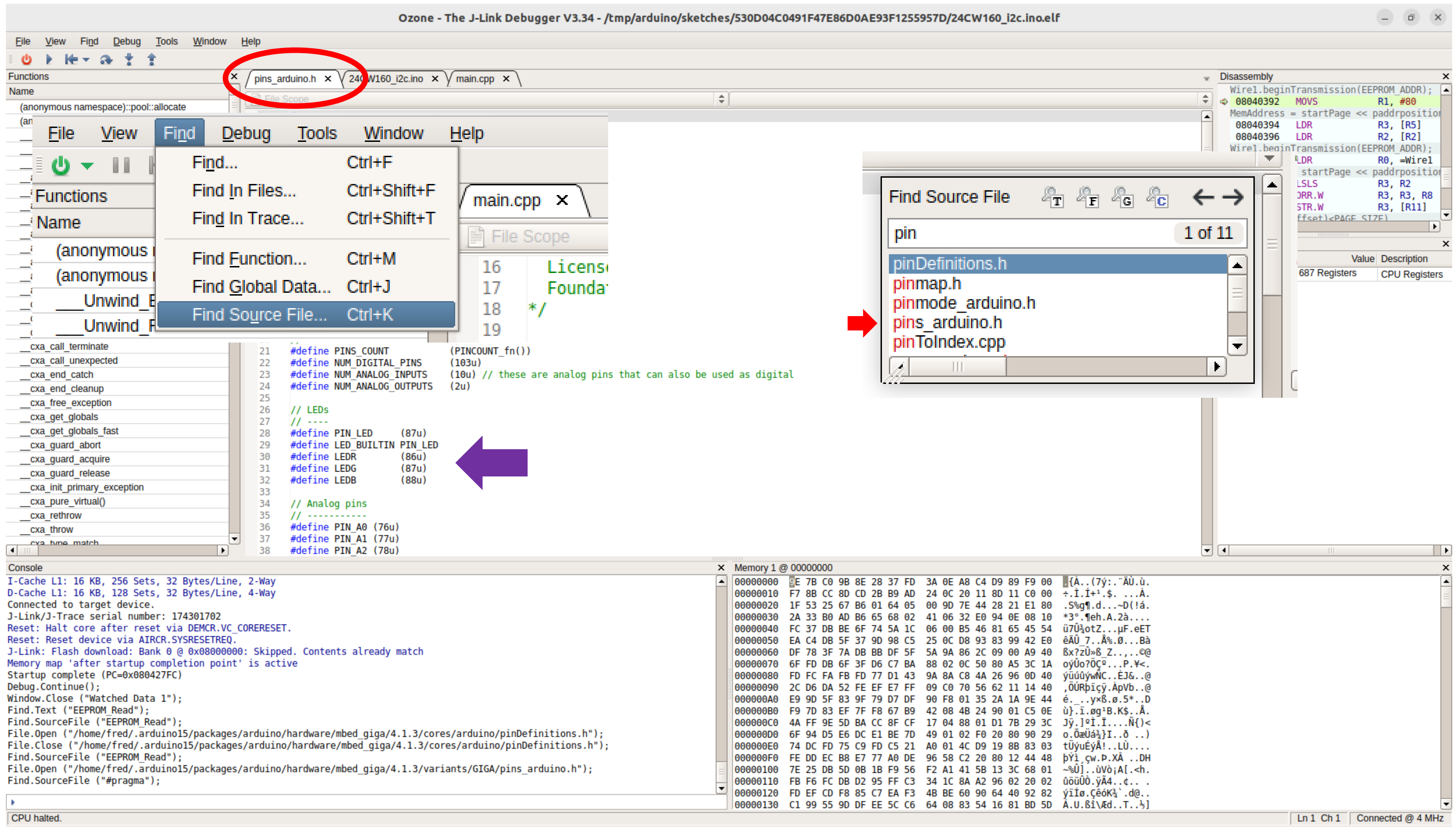
Arduino Pro Primer

Debugging Arduino Sketches

Debug the Microchip 24CW160 EEPROM Driver

DigiKey

Find Arduino Source Files



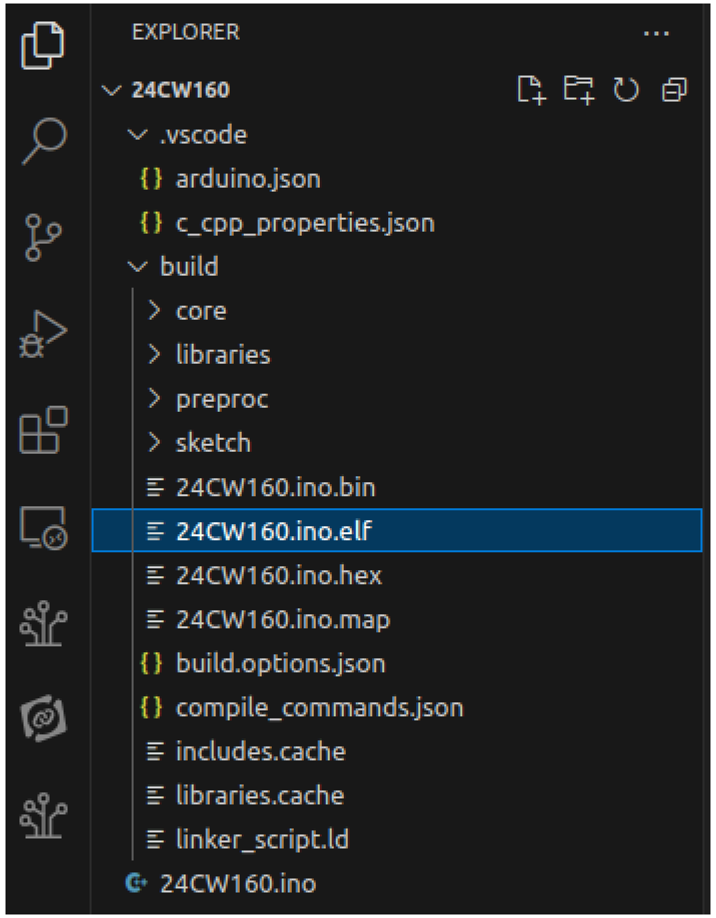
```
   Licens
   Founda
*/

#define PINS_COUNT           (PINCOUNT_fn())
#define NUM_DIGITAL_PINS     (103u)
#define NUM_ANALOG_INPUTS    (10u) // these are analog pins that can also be used as digital
#define NUM_ANALOG_OUTPUTS   (2u)

// LEDs
// ----
#define PIN_LED     (87u)
#define LED_BUILTIN PIN_LED
#define LEDR        (86u)
#define LEDG        (87u)
#define LEDB        (88u)

// Analog pins
// -----------
#define PIN_A0 (76u)
#define PIN_A1 (77u)
#define PIN_A2 (78u)
```

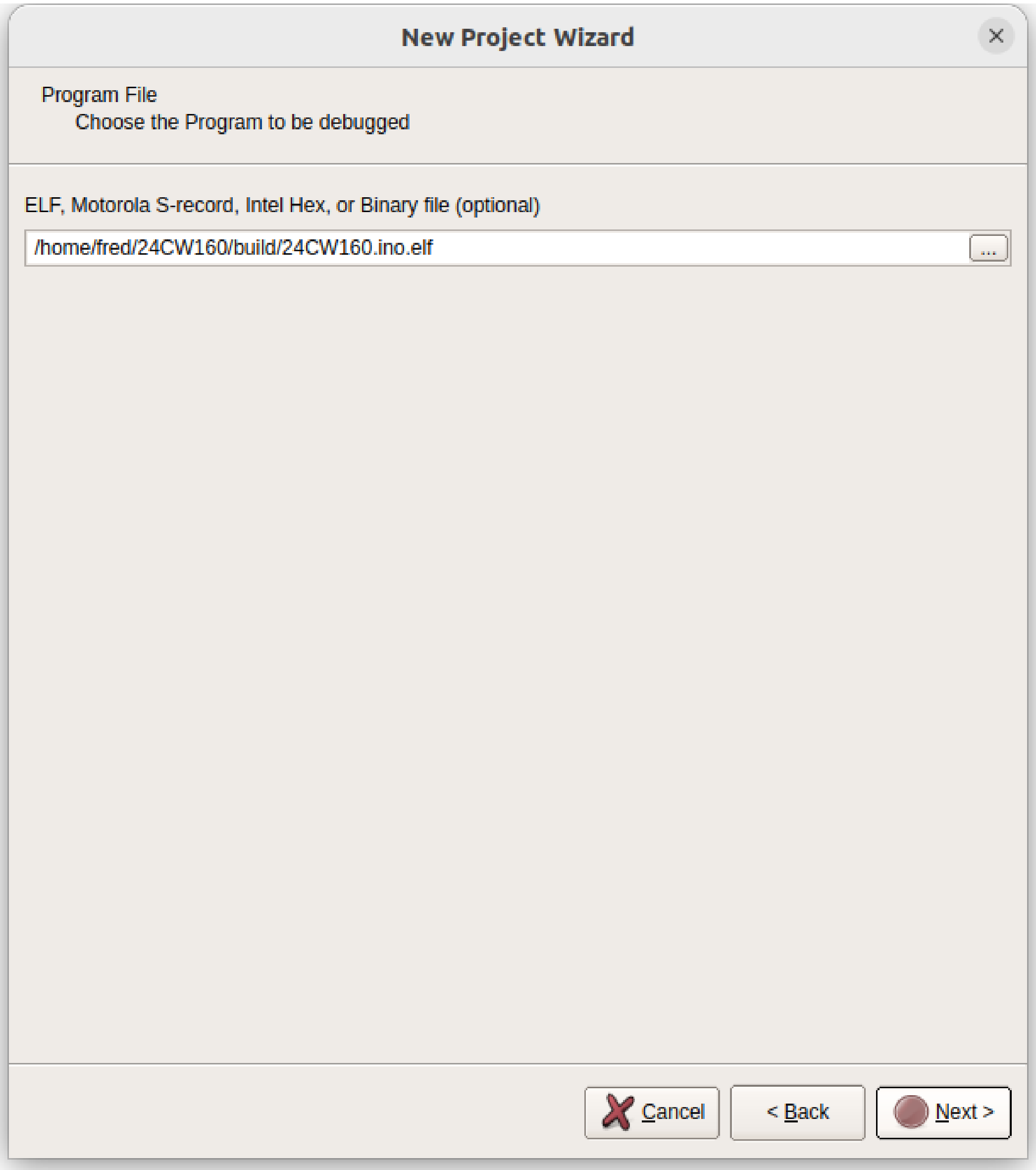
```
Wire1.beginTransmission(EEPROM_ADDR);
 08040392   MOVS          R1, #80
MemAddress = startPage << paddrpositior
 08040394   LDR           R3, [R5]
 08040396   LDR           R2, [R2]
Wire1.beginTransmission(EEPROM_ADDR);
            LDR           R0, =Wire1
 startPage << paddrpositior
            LSLS          R3, R2
            ORR.W         R3, R3, R8
            STR.W         R3, [R11]
ffset)<PAGE_SIZE)
```

```
I-Cache L1: 16 KB, 256 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 16 KB, 128 Sets, 32 Bytes/Line, 4-Way
Connected to target device.
J-Link/J-Trace serial number: 174301702
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link: Flash download: Bank 0 @ 0x08000000: Skipped. Contents already match
Memory map 'after startup completion point' is active
Startup complete (PC=0x080427FC)
Debug.Continue();
Window.Close ("Watched Data 1");
Find.Text ("EEPROM_Read");
Find.SourceFile ("EEPROM_Read");
File.Open ("/home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/cores/arduino/pinDefinitions.h");
File.Close ("/home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/cores/arduino/pinDefinitions.h");
Find.SourceFile ("EEPROM_Read");
File.Open ("/home/fred/.arduino15/packages/arduino/hardware/mbed_giga/4.1.3/variants/GIGA/pins_arduino.h");
Find.SourceFile ("#pragma");
```

```
00000000  9E 7B C0 9B 8E 28 37 FD  3A 0E A8 C4 D9 89 F9 00  .{À..(7ý:.¨ÄÙ.ù.
00000010  F7 8B CC 8D CD 2B B9 AD  24 0C 20 11 8D 11 C0 00  ÷.Ì.Í+¹.$. ...À.
00000020  1F 53 25 67 B6 01 64 05  00 9D 7E 44 28 21 E1 80  .S%g¶.d...~D(!á.
00000030  2A 33 B0 AD B6 65 68 02  41 06 32 E0 94 0E 08 10  *3°.¶eh.A.2à....
00000040  FC 37 DB BE 6F 74 5A 1C  06 00 B5 46 81 65 45 54  ü7Û¾otZ...µF.eET
00000050  EA C4 DB 5F 37 9D 98 C5  25 0C D8 93 83 99 42 E0  êÄÛ_7..Å%.Ø...Bà
00000060  DF 78 3F 7A DB BB DF 5F  5A 9A 86 2C 09 00 A9 40  ßx?zÛ»ß_Z..,..©@
00000070  6F FD DB 6F 3F D6 C7 BA  88 02 0C 50 80 A5 3C 1A  oýÛo?ÖÇº...P.¥<.
00000080  FD FC FA FB FD 77 D1 43  9A 8A C8 4A 26 96 0D 40  ýüúûýwÑC..ÈJ&..@
00000090  2C D6 DA 52 FE EF E7 FF  09 C0 70 56 62 11 14 40  ,ÖÚRþïçÿ.ÀpVb..@
000000A0  E9 9D 5F 83 9F 79 D7 DF  90 F8 01 35 2A 1A 9E 44  é._..y×ß.ø.5*..D
000000B0  F9 7D 83 EF 7F F8 67 B9  42 08 4B 24 90 01 C5 0E  ù}.ï.øg¹B.K$..Å.
000000C0  4A FF 9E 5D BA CC 8F CF  17 04 88 01 D1 7B 29 3C  Jÿ.]ºÌ.Ï....Ñ{)<
000000D0  6F 94 D5 E6 DC E1 BE 7D  49 01 02 F0 20 80 90 29  o.ÕæÜá¾}I..ð ..)
000000E0  74 DC FD 75 C9 FD C5 21  A0 01 4C D9 19 8B 83 03  tÜýuÉýÅ!..LÙ....
000000F0  FE DD EC B8 E7 77 A0 DE  96 58 C2 20 80 12 44 48  þÝì¸çw.Þ.XÂ ..DH
00000100  7E 25 DB 5D 0B 1B F9 56  F2 A1 41 5B 13 3C 68 01  ~%Û]..ùVò¡A[.<h.
00000110  FB F6 FC DB D2 95 FF C3  34 1C 8A A2 96 02 20 02  ûöüÛÒ.ÿÃ4..¢.. .
00000120  FD EF CD F8 85 C7 EA F3  4B BE 60 90 64 40 92 82  ýïÍø.ÇêóK¾`.d@..
00000130  C1 99 55 9D DF EE 5C C6  64 08 83 54 16 81 BD 5D  Á.U.ßî\Æd..T..½]
```

Visual Studio Code Works Too




New Project Wizard

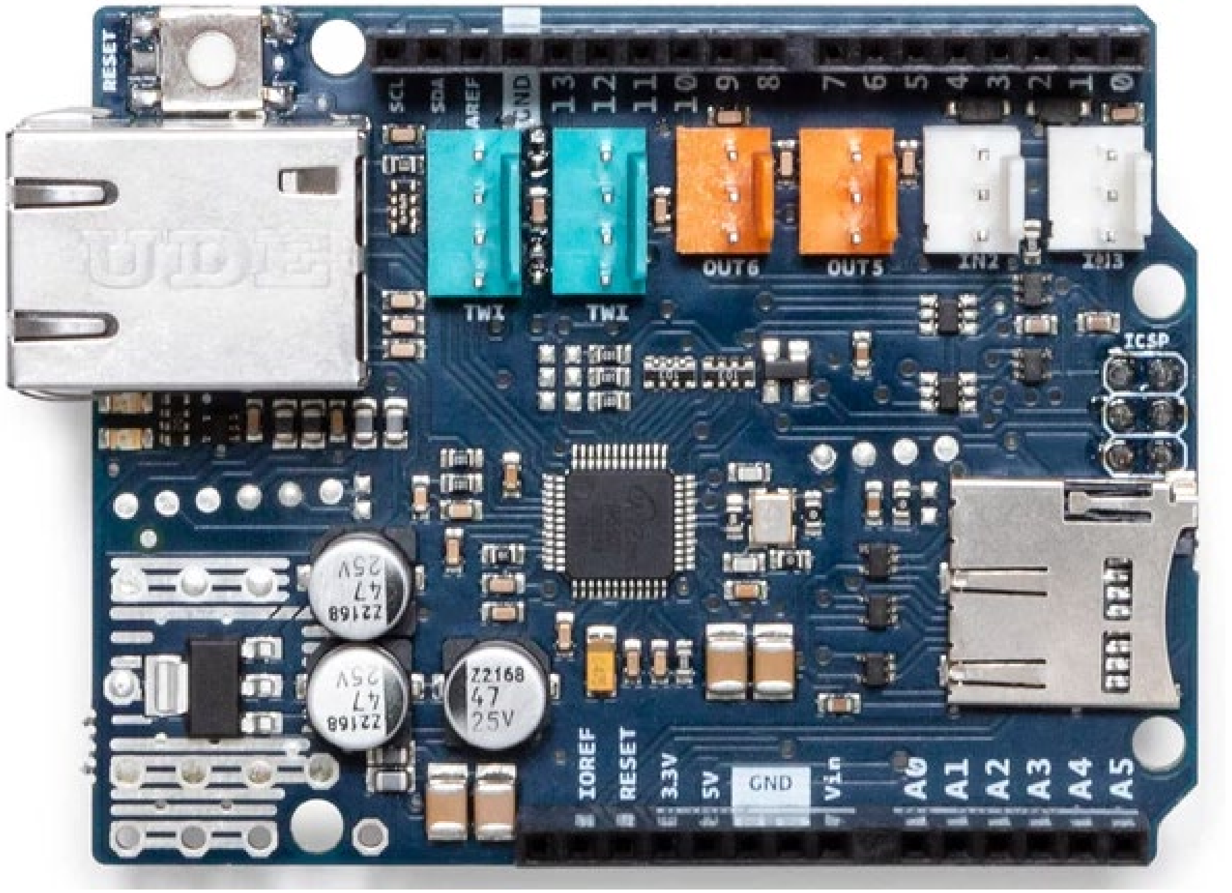
Program File
Choose the Program to be debugged

ELF, Motorola S-record, Intel Hex, or Binary file (optional)

/home/fred/24CW160/build/24CW160.ino.elf

Next Time…

MORE TO COME..



Thank you for attending!!!

Please consider the resources below:

- Today’s Download Package
- arduino.cc

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets