Light Sequencing and Decoding DMX with an Arduino
Posted by Ristola - 2011/11/05 20:28

This thread discusses the Content article: Light Sequencing and Decoding DMX with an Arduino

Please explain more on the DMX-512 Plug-in in Vixen as I do not see that option.
Also I assume the interface is USB to FTD2XX (serial) to the arduino.

Can you provide the plugin, how to install

Thanks.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2011/11/05 21:58

Hi. I just updated the page with the reply to this post. I checked the latest Vixen install and they changed the name of the plugin from DMX-512 to Enttec Open DMX. However it maps to the same library. So you will need to choose this as the plugin. Also, it does require an FTDI USB/Serial chip since that is standard on OpenDMX devices.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Dennis Davis - 2011/11/08 11:47

Akiba wrote:
Hi. I just updated the page with the reply to this post. I checked the latest Vixen install and they changed the name of the plugin from DMX-512 to Enttec Open DMX. However it maps to the same library. So you will need to choose this as the plugin. Also, it does require an FTDI USB/Serial chip since that is standard on OpenDMX devices.

Hello,

Thanks for the write up as it appears to allow me to accomplish exactly what I purchased an Arduino for..... going all Griswold with my Christmas lights!

I am completely new to all things DMX and Arduino. I have the hardware side of switching outlets on and off figured out, just trying to wrap my head around how to integrate vixen and the arduino. I've stepped through and believe that I have vixen figured out.

I'm stumbling on this part:
Also, it does require an FTDI USB/Serial chip since that is standard on OpenDMX devices.

What does this mean? I googled for an FTDI USB/Serial chip and came up with what appears to be different USB to Serial adapters.

Do I somehow connect the serial output to the Arduino? If so, how? Or do I need a serial port on my PC that connects to this adapter and then connects to the Arduino via USB?

Thanks for the help.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2011/11/08 16:47

Hi. Sorry I wasn't more specific. Vixen will automatically look for a specific USB device to send the DMX data to. This device is a chip made by a company called FTDI and is a USB to Serial bridge that is used on OpenDMX products. The Arduino Duemilanova uses this IC as well as the Freakduino and various other Arduinos that can be found, ie: Boarduino at Adafruit, Arduino Pro at Sparkfun.

Unfortunately, the more recent Arduino devices, ie: Arduino Uno, uses a different USB to serial chip so its basically unable to be used with OpenDMX.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/08 16:56

Here is a picture of the FTDI chip. You can see that it's on the actual circuit board. These are pics of one of my test boards so they're a bit ...err... battered.

http://freaklabs.org/images/fbfiles/images/DMX__005.jpg

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/09 10:41

Akiba wrote:
Hi. Sorry I wasn't more specific. Vixen will automatically look for a specific USB device to send the DMX data to. This device is a chip made by a company called FTDI and is a USB to Serial bridge that is used on OpenDMX products. The Arduino Duemilanova uses this IC as well as the Freakduino and various other Arduinos that can be found, ie: Boarduino at Adafruit, Arduino Pro at Sparkfun.

Unfortunately, the more recent Arduino devices, ie: Arduino Uno, uses a different USB to serial chip so its basically unable to be used with OpenDMX.

Not sure what the issue is, but my account seems to have disappeared.

I have the MEGA 2560 and that doesn't have the FTDI chip. Is there anything I can do to fool Vixen into allowing the MEGA 2560 to work?

I see that the MEGA 1280 has an FTDI chip, will that work? I just hate to sell the 2560 considering I haven't even powered it up yet.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/09 17:25

Oops, sorry about the account. I was getting a lot of spam registrations and removed all of them, but I think I took your account with it. I really need to upgrade this site and add a CAPTCHA to stop all of that.

Regarding your questions, its possible to fool it into thinking there's an FTDI device, but it wouldn't be pretty. You'd have to go into the USB driver of the Arduino Mega's USB device and modify the USB VID/PID (Vendor ID/Product ID) in the USB descriptors to match what FTDI uses. That will make it look like an FTDI device to Vixen. Unfortunately it takes a bit of expertise with the USB protocol.

The Mega1280 should work since it has an FTDI chip. In general, the FTDI chip is the main device needed to talk to Vixen so any board with that on it should be fine. Alternatively, you can use an FTDI based cable and then plug the output of it into the Tx/Rx of the serial port of the 2560 for testing.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by j reckart - 2011/11/12 08:15

Dennis Davis wrote:
Akiba wrote:
Hi. Sorry I wasn't more specific. Vixen will automatically look for a specific USB device to send the DMX data to. This

device is a chip made by a company called FTDI and is a USB to Serial bridge that is used on OpenDMX products. The Arduino Duemilanova uses this IC as well as the Freakduino and various other Arduinos that can be found, ie: Boarduino at Adafruit, Arduino Pro at Sparkfun.

Unfortunately, the more recent Arduino devices, ie: Arduino Uno, uses a different USB to serial chip so its basically unable to be used with OpenDMX.

Not sure what the issue is, but my account seems to have disappeared.

I have the MEGA 2560 and that doesn't have the FTDI chip. Is there anything I can do to fool Vixen into allowing the MEGA 2560 to work?

I see that the MEGA 1280 has an FTDI chip, will that work? I just hate to sell the 2560 considering I haven't even powered it up yet.

Or at sparkfun get this breakout of the fdi chip only, for only $15 instead of buying a whole arduino
http://www.sparkfun.com/products/9716

or get the cable with chip built into the usb plug itself
here
http://www.sparkfun.com/products/9718

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/12 18:26

Yes, a good way to get around the FTDI chip issue is to use an FTDI based cable and plug the serial rx/tx in directly. Those cables are all over the place and can easily be found on adafruit and sparkfun :)

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/17 21:41

Thanks for the responses on the work around guys. I was able to return the Mega 2560 and have received my Mega 1280.

I fired up the 1280 today and using the blink sketch was able to successfully toggle my lights on and off.

I then went and made the modifications to the Arduino 0022 HardwareSerial.cpp.

I then downloaded the source code for step 1 (dmx_arduino_part1.pde), and modified the sketch to meet the requirements for the pins I am using. The only changes I have made are listed below:

```
// pins that will toggle based on dmx data
unsigned char dmx_pin = {22, 23, 24, 25};
```

I have my vixen sequence setup to use Enttec Open DMX and channels 1 through 4.

Anyone see anything that I've done wrong to this point? I connect the Arduino, launch Vixen, load the sequence and then play it. The Rx light instantly turns on once the sequence is playing, but I am not getting any outputs firing on the Arduino.

Help please.....

Thanks everyone!

Not sure how this works, but just in case, here is the entire sketch that is loaded to the Arduino.

```
#define DMX_NUM_CHANNELS 4

enum
{
  DMX_IDLE,
  DMX_BREAK,
  DMX_START,
  DMX_RUN
};

volatile unsigned char dmx_state;

// this is the start address for the dmx frame
unsigned int dmx_start_addr = 1;

// this is the current address of the dmx frame
unsigned int dmx_addr;

// pins that will toggle based on dmx data
unsigned char dmx_pin = {22, 23, 24, 25};

// this is used to keep track of the channels
unsigned int chan_cnt;

// this holds the dmx data
unsigned char dmx_data;

// tell us when to update the pins
volatile unsigned char update;

/*****************************************************************************/
/*!
  This is the setup code
*/
/*****************************************************************************/
void setup()
{
  // set update flag idle
  update = 0;

  // set default DMX state
  dmx_state = DMX_IDLE;

  // set DMX pins to ouput and idle value
  for (int i=0; i
```

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/17 23:13

Hi. Can you post the HardwareSerial.cpp code just in case. Also, in the Vixen software, did you declare the DMX channels in the setup wizard? One way to check if your channels are working or not is to use the "test" window in Vixen. I've enclosed a picture of how to access it.

http://freaklabs.org/images/fbfiles/images/vixentest.jpg

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/18 05:15

_____

```
/*
  HardwareSerial.cpp - Hardware serial library for Wiring
  Copyright (c) 2006 Nicholas Zambetti.  All right reserved.

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.

  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
  Lesser General Public License for more details.

  You should have received a copy of the GNU Lesser General Public
  License along with this library; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

  Modified 23 November 2006 by David A. Mellis
  Modified 28 September 2010 by Mark Sproul
*/

#include
#include
#include
#include
#include "wiring.h"
#include "wiring_private.h"

// this next line disables the entire HardwareSerial.cpp,
// this is so I can support Attiny series and any other chip without a uart
#if defined(UBRRH) || defined(UBRR0H) || defined(UBRR1H) || defined(UBRR2H) || defined(UBRR3H)

#include "HardwareSerial.h"

// Define constants and variables for buffering incoming serial data.  We're
// using a ring buffer (I think), in which rx_buffer_head is the index of the
// location to which to write the next incoming character and rx_buffer_tail
// is the index of the location from which to read.
#if (RAMEND < 1000)
  #define RX_BUFFER_SIZE 32
#else
  #define RX_BUFFER_SIZE 128
#endif

struct ring_buffer
{
  unsigned char buffer;
  int head;
  int tail;
};

#if defined(UBRRH) || defined(UBRR0H)
  ring_buffer rx_buffer  =  { { 0 }, 0, 0 };
#endif
#if defined(UBRR1H)
  ring_buffer rx_buffer1  =  { { 0 }, 0, 0 };
#endif
```

```
#if defined(UBRR2H)
  ring_buffer rx_buffer2  =  { { 0 }, 0, 0 };
#endif
#if defined(UBRR3H)
  ring_buffer rx_buffer3  =  { { 0 }, 0, 0 };
#endif

inline void store_char(unsigned char c, ring_buffer *rx_buffer)
{
  int i = (unsigned int)(rx_buffer->head + 1) % RX_BUFFER_SIZE;

  // if we should be storing the received character into the location
  // just before the tail (meaning that the head would advance to the
  // current location of the tail), we're about to overflow the buffer
  // and so we don't write the character or advance the head.
  if (i != rx_buffer->tail) {
    rx_buffer->buffer = c;
    rx_buffer->head = i;
  }
}

/* Commenting out the ISR from HardwareSerial.cpp

#if defined(USART_RX_vect)
  SIGNAL(USART_RX_vect)
  {
  #if defined(UDR0)
    unsigned char c  =  UDR0;
  #elif defined(UDR)
    unsigned char c  =  UDR;  //  atmega8535
  #else
    #error UDR not defined
  #endif
    store_char(c, &rx_buffer);
  }
#elif defined(SIG_USART0_RECV) && defined(UDR0)
  SIGNAL(SIG_USART0_RECV)
  {
    unsigned char c  =  UDR0;
    store_char(c, &rx_buffer);
  }
#elif defined(SIG_UART0_RECV) && defined(UDR0)
  SIGNAL(SIG_UART0_RECV)
  {
    unsigned char c  =  UDR0;
    store_char(c, &rx_buffer);
  }
//#elif defined(SIG_USART_RECV)
#elif defined(USART0_RX_vect)
  // fixed by Mark Sproul this is on the 644/644p
  //SIGNAL(SIG_USART_RECV)
  SIGNAL(USART0_RX_vect)
  {
  #if defined(UDR0)
    unsigned char c  =  UDR0;
  #elif defined(UDR)
    unsigned char c  =  UDR;  //  atmega8, atmega32
  #else
    #error UDR not defined
  #endif
    store_char(c, &rx_buffer);
  }
#elif defined(SIG_UART_RECV)
  // this is for atmega8
  SIGNAL(SIG_UART_RECV)
```

```
  {
  #if defined(UDR0)
    unsigned char c  =  UDR0;  //  atmega645
  #elif defined(UDR)
    unsigned char c  =  UDR;  //  atmega8
  #endif
    store_char(c, &rx_buffer);
  }
#elif defined(USBCON)
  #warning No interrupt handler for usart 0
  #warning Serial(0) is on USB interface
#else
  #error No interrupt handler for usart 0
#endif

//#if defined(SIG_USART1_RECV)
#if defined(USART1_RX_vect)
  //SIGNAL(SIG_USART1_RECV)
  SIGNAL(USART1_RX_vect)
  {
    unsigned char c = UDR1;
    store_char(c, &rx_buffer1);
  }
#elif defined(SIG_USART1_RECV)
  #error SIG_USART1_RECV
#endif

#if defined(USART2_RX_vect) && defined(UDR2)
  SIGNAL(USART2_RX_vect)
  {
    unsigned char c = UDR2;
    store_char(c, &rx_buffer2);
  }
#elif defined(SIG_USART2_RECV)
  #error SIG_USART2_RECV
#endif

#if defined(USART3_RX_vect) && defined(UDR3)
  SIGNAL(USART3_RX_vect)
  {
    unsigned char c = UDR3;
    store_char(c, &rx_buffer3);
  }
#elif defined(SIG_USART3_RECV)
  #error SIG_USART3_RECV
#endif


*/


// Constructors ////////////////////////////////////////////////////////////////

HardwareSerial::HardwareSerial(ring_buffer *rx_buffer,
  volatile uint8_t *ubrrh, volatile uint8_t *ubrrl,
  volatile uint8_t *ucsra, volatile uint8_t *ucsrb,
  volatile uint8_t *udr,
  uint8_t rxen, uint8_t txen, uint8_t rxcie, uint8_t udre, uint8_t u2x)
{
  _rx_buffer = rx_buffer;
  _ubrrh = ubrrh;
  _ubrrl = ubrrl;
  _ucsra = ucsra;
  _ucsrb = ucsrb;
  _udr = udr;
```

```
  _rxen = rxen;
  _txen = txen;
  _rxcie = rxcie;
  _udre = udre;
  _u2x = u2x;
}

// Public Methods //////////////////////////////////////////////////////////////

void HardwareSerial::begin(long baud)
{
  uint16_t baud_setting;
  bool use_u2x = true;

#if F_CPU == 16000000UL
  // hardcoded exception for compatibility with the bootloader shipped
  // with the Duemilanove and previous boards and the firmware on the 8U2
  // on the Uno and Mega 2560.
  if (baud == 57600) {
    use_u2x = false;
  }
#endif

  if (use_u2x) {
    *_ucsra = 1 > 8;
  *_ubrrl = baud_setting;

  sbi(*_ucsrb, _rxen);
  sbi(*_ucsrb, _txen);
  sbi(*_ucsrb, _rxcie);
}

void HardwareSerial::end()
{
  cbi(*_ucsrb, _rxen);
  cbi(*_ucsrb, _txen);
  cbi(*_ucsrb, _rxcie);
}

int HardwareSerial::available(void)
{
  return (unsigned int)(RX_BUFFER_SIZE + _rx_buffer->head - _rx_buffer->tail) % RX_BUFFER_SIZE;
}

int HardwareSerial::peek(void)
{
  if (_rx_buffer->head == _rx_buffer->tail) {
    return -1;
  } else {
    return _rx_buffer->buffer;
  }
}

int HardwareSerial::read(void)
{
  // if the head isn't ahead of the tail, we don't have any characters
  if (_rx_buffer->head == _rx_buffer->tail) {
    return -1;
  } else {
    unsigned char c = _rx_buffer->buffer;
    _rx_buffer->tail = (unsigned int)(_rx_buffer->tail + 1) % RX_BUFFER_SIZE;
    return c;
  }
}
```

```
void HardwareSerial::flush()
{
  // don't reverse this or there may be problems if the RX interrupt
  // occurs after reading the value of rx_buffer_head but before writing
  // the value to rx_buffer_tail; the previous value of rx_buffer_head
  // may be written to rx_buffer_tail, making it appear as if the buffer
  // don't reverse this or there may be problems if the RX interrupt
  // occurs after reading the value of rx_buffer_head but before writing
  // the value to rx_buffer_tail; the previous value of rx_buffer_head
  // may be written to rx_buffer_tail, making it appear as if the buffer
  // were full, not empty.
  _rx_buffer->head = _rx_buffer->tail;
}

void HardwareSerial::write(uint8_t c)
{
  while (!((*_ucsra) & (1
```

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by j reckart - 2011/11/18 07:42

_____

OK! Figured you out i think! you are jumping over a bunch of steps in the middle!
your signal chain is missing a big part in the middle!

From the top....
Vixen is a computer program, that sequences dmx lights to music! First Lets define DMX... Very technical wiki article here http://en.wikipedia.org/wiki/DMX512

but the down and dirty Tennessee country version is this!

 DMX is a serial protocol, meaning "a 'master' or controller sends pulses down a line in a very specific format, at a very specific timing, that means something to a 'slave' device reading these pulses,decodeing them to mean something, and then doing that something, "

having said that, you have several components speaking several languages, with several conversions happening in between!
The ftdi chip converts USB to serial! A real serial port outputs signals conforming to the RS-232 standard. this standard usually uses pulses around +12/-12 volts, which is way to "big" for your arduino. Your arduino uses +/-5v to comunicate on it's serial port. the Lower voltages that the arduion uses is as "ttl" levels. To have comunicate with real serial port, you would need a ttl to serial "level" shifter! The serial pulses are the same, just at different voltage levels.
Most arduino use use USB to comunicate with a computer. USB is a form of RS232, but at the lower 5v levels.
So the arduino has a usb to rs232 convert chip installed, the FTDI chip! (the newer arduino Uno's use a different USB chip, therefore not useful in this specific case.)

 the DMX512 Protocol is based on something called EIA485, or RS485. not RS232!
So what you need is a converter chip to convert RS232 to RS485(which DMX is based on!)

Step one...
Vixen is programed the way you want it to display what light when ever.... you set it up to use the Enttec Open dmx dongle (which coincidently uses the say ftdi chip arduino does!) Vixen the does not output "real" DMX. it puts out "DMX lanuage" but in RS232 format. it does this to a "virual" com port, ie the FTDI chip, wheter the cable or arduino, because they both use the same chip as the Enttec OpenDMX.

 The FTDI cable, or onboard chip on an older Arduino, will convert it from usb to TTL level RS232.
This then needs to be converted to RS485, the basis for DMX. A chip such as the Maxim
max485 converts the RS232 TTL to RS485, which "pulses" at +/- 2.5volt! you now have a real DMX signal!

all of that before you get to the arduino flashing lights to a DMX signal!

now that you have a DMX signal, send it down a line 10, 20, 100, or 1000 feet to a dmx slave device, or a daisy chain of dmx devices.

your dmx decoder project would fit the description of a DMX slave. But by itself it only understands RS232 @TTL levels, so again another converter is need! The max485 chip is what is need, but this is a seperate one. this one will convert the DMX/RS485 back to RS232 @ TTL. Now that the DMX signal has been converted back and forth, until it gets to a usuable format to the end device, you arduino project can understand it, decode it, and use it to know when to flash lights on/off, or whatever!
I'm sure someone will chime in and say this or that isn't technicly correct, but I told you this was a down and dirty explanation just to get you to understand the concept.
a recap:
vixen/RS232 -->Virual com port (Enttec OpenDMX/Arduion "fake OpenDMX)still RS232 but a TTL levels --> max485 chip to convert RS232 TTL to RS485/DMX --> DMX cable extending however many feet needed! --> 2nd max485 to convert RS485/DMX back to RS232 TTL --> into the RX pin on the end Arduino to do what ever!

Why all this? RS232 specs call for a cable length of max 50'. although longer distances are possible in certain circumstances, 50 is the offical max. RS485 is good out like 4000', and up 32 devices can be daisy chained. so when DMX was designed to be used in theater or arenas, rs485 was chosen as the base of rs232!

very simplfied, in a nutshell description!

so you are trying to make you arduino ftdi chip speak DMX straight to you arduino, with out converting to DMX first! at the least you are going to need an ftdi cable or beakout board as listed above, and 2 max485 chips, plus all of the asscoiated resistors, connectors, and component to go along!

JReckart

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2011/11/18 08:15

Yes, I'm not using the DMX physical media which is RS-485. I'm taking the signal directly from the USB and then feeding it into an Arduino. In the case of OpenDMX, they take the signal from USB via the FTDI chip and then feed it directly into a serial to RS-485 transceiver, ie: MAX485. If you already have a DMX universe going, then its best to have the Arduino as an end node, like a light. Otherwise, simultaneously decoding and forwarding the DMX signals might overwhelm the Arduino.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2011/11/18 08:22

Dennis, thanks for posting the code. It looks like you commented out the right part. I thought I might have a Mega lying around here but I couldn't find it. Can you try out the test controls on Vixen to see if you can toggle an LED on and off? Also, can you check to make sure your LEDs are wired up properly and you're selecting the proper board in the Arduino IDE?

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Dennis Davis - 2011/11/18 10:40

Akiba, first of all I wanted to thank you for all of your help on this. It is definitely appreciated.

As far as the test controls on Vixen, I'm not really sure exactly how it is supposed to work. I clicked the button that you had in the screen shot and toggled check marks on and off as well as adjusting the slider from 100 to various positions and nothing was happening with the Arduino.

I did not change up the sketch to start with pin 13 so that I could test with the onboard LED.

I'm not doing any testing with an actual LED. I have a 4 outlet box with individual control over each outlet through a

electro-mechanical relay that is powered with a separate 5V power supply and a transistor array to isolate the Arduino. I know those are working properly as I'm able to control them with no problems using the basic blink sketch.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/18 14:13

I got home and was able to alter my sketch so that it was starting with pin 13 for the outputs.

I loaded the sketch, shut down the Arduino IDE and then ran Vixen and have the same results as before. Nothing happening.

Then I went into Vixen and tried using the test channels feature and nothing appeared to be working.

Attached is a screenshot of the Arduino IDE showing the board I have selected. COM port is set as COM8 which mathces what's in device manager. http://freaklabs.org/images/fbfiles/images/Board_Selection.jpg

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/18 20:38

Hi. I just found and dug out my Arduino Mega 1280. It was apparently hiding from me in an old drawer. Anyways, I tested out the code and I believe I found the problem. Or at least I got it working on my system. The name of the interrupt service routine (ISR) is different between the standard Arduino (Duemilanova) which the Freakduino is based on and the Arduino Mega. In the standard Arduino, the ISR is declared as "ISR(USART_RX_vect)". In the Mega, its declared as "ISR(SIG_USART0_RECV)". This is because the Mega has multiple UARTs.

So after I made the modification, then everything was working and I was toggling on pins 22 through 25.

The zip file is attached with the modified files for the Mega and also the Arduino v022 HardwareSerial.cpp file. The serial file was not modified, but I included it for completeness.

http://freaklabs.org/images/fbfiles/files/dmx_arduino_mega.zip

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/19 08:46

Akiba wrote:
Hi. I just found and dug out my Arduino Mega 1280. It was apparently hiding from me in an old drawer. Anyways, I tested out the code and I believe I found the problem. Or at least I got it working on my system. The name of the interrupt service routine (ISR) is different between the standard Arduino (Duemilanova) which the Freakduino is based on and the Arduino Mega. In the standard Arduino, the ISR is declared as "ISR(USART_RX_vect)". In the Mega, its declared as "ISR(SIG_USART0_RECV)". This is because the Mega has multiple UARTs.

So after I made the modification, then everything was working and I was toggling on pins 22 through 25.

The zip file is attached with the modified files for the Mega and also the Arduino v022 HardwareSerial.cpp file. The serial file was not modified, but I included it for completeness.

http://freaklabs.org/images/fbfiles/files/dmx_arduino_mega.zip

Akiba, you are the man!!

It works flawlessly. This will be so much better than a massive and complicated blink program. I'll try and get some pics and video up. Took a short test video last night.

Thanks a ton.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Dennis Davis - 2011/11/19 08:52

Dennis Davis wrote:

Akiba, you are the man!!

It works flawlessly. This will be so much better than a massive and complicated blink program. I'll try and get some pics and video up. Took a short test video last night.

Thanks a ton.

Take off the .txt and replace with .mov for the video. http://freaklabs.org/images/fbfiles/files/IMG_1053.txt

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/19 09:36

Ha ha ha. That's cool. I'm glad you were able to get it up and running. Good luck with your Xmas display. Hope you make your neighbors' jaws drop :)

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Other Baud Rates? - 2011/11/23 00:09

Thanks so much for all of the work done, video and code provided.
I have had lots of fun playing with Vixen and the Arduino.

I was wondering, will the changes to the hardware.cpp still work if selecting a slower baud rate such as (57600) versus the (250000) ??

I have a few XBEE Pro SB2's laying around, and was curious if using the DMX USB PRO in Vixen would work.
I have made several attempts to try it, but not sure if the problem is in the XBEE setup, or the hardware.cpp code changes and arduino example?

It would be so nice to go Wireless and with only a few DMX channels used, I wouldn't think 57.6k would be any issue.

Anyone have any suggestions?

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/23 18:16

I don't think the XBees would work because the DMX speed is standardized at 250kbps and the XBee's have no way to decode the bitstream. The FTDI would have to feed directly into the XBee's serial port but without access to the ability to program XBees (aka toolchain and source code), then you won't be able to implement the DMX decoding.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Ristola - 2011/11/23 19:23

Will the Arduino code only Decode the DMX at 250k baud rate?
I am under the impression with all of the available drivers, they were using DMX at different baud rates.

I was trying to change the baud rate in the Arduino DMX example to 57600 from the FTDI with no luck using the Generic Serial driver or the UsbDmxPro driver.

I have a FTDI driving (1) of the XBee's and another on the Arduino. Using as wireless serial link. (Max baud rate is 57600) The data is being passed, but the Arduino code isn't decoding at anything less then 250K trying all of the serial drivers in Vixen.

Thanks for the input.
Trying to figure out what is being sent out from Vixen with the other drivers, if it is not DMX.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/23 20:08

The Vixen DMX driver sends out data at 250 kbps. If the code were open source, you'd probably be able to change it. Unfortunately, it's not an open source application and I haven't found anything else open source that can both sequence lights and sync them to music.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Mark - 2011/11/27 18:43

This is great... just what I was looking for.

I bought an UNO board 6 months or so back, and just happen to have a FIDI cable... so is it possible to use the UNO and the cable, just by bypassing the usb input and going direct to the tx/rx on the board? And if so, does the code change much?

Thanks!

Markm

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/11/27 19:31

I believe the UNO has the AVR USB IC feeding directly into the Tx/Rx of the UART. I'm not sure if this can be bypassed. However you can certainly use the FTDI cable to get the raw serial outputs. The best is the Duemilanova or an Arduino clone that uses FTDI since everything is on the same board. Other than the Freakduino, I personally like the Duemilanova and the Adafruit Boarduino.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Mark - 2011/11/27 19:34

Great, thanks Akiba.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by DeepBlue - 2011/12/08 09:24

Hey Akiba

First off, great tutorial. Informative but straight to the point, I wish there were more tutorials like this.

Now to business! I currently have this set up to a solid state relay that I made (optocouplers & triacs). The SSR is controlling 4 120v christmas lights and the SSR is fully working, and your code is kinda working. It can turn the lights on, but when sending a continuous signal to turn the lights on (either from the test panel or a run of "on" on a particular channel), the lights flash. The flash is random, and sometimes a light down the chain will flash on.

I believe this is a coding problem. That somewhere along the line, its reading the wrong address / channel, turning that channel off and turning another one on (You don't always see the other one turn on because of the zero cross). Ive tried version 0022-1.0 and different pin configurations but to no avail.

So I just wanted to know your input. Oh, and im using a Duimilanove.

Copious amounts of gratitude

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/12/08 18:11

Hmmm...I haven't seen this issue before. Have you tested it on a setup with just LEDs? I would probably start small and see if there are any issues there first. It would be much easier to isolate. As far as my own testing goes, I've used the sequencer and the test panel to check using high power LEDs and haven't seen any problems with wrong sequencing. Can you let me know if it works okay on a small testbed using just LEDs as loads?

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by chris - 2011/12/16 09:13

I'm having the same problem i think.

using a mega 1280 with 0022 and the files for the mega provided. when i set any channel to on using the test the others will fliker

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by chris - 2011/12/16 09:17

DeepBlue wrote:
Hey Akiba

First off, great tutorial. Informative but straight to the point, I wish there were more tutorials like this.

Now to business! I currently have this set up to a solid state relay that I made (optocouplers & triacs). The SSR is controlling 4 120v christmas lights and the SSR is fully working, and your code is kinda working. It can turn the lights on, but when sending a continuous signal to turn the lights on (either from the test panel or a run of "on" on a particular channel), the lights flash. The flash is random, and sometimes a light down the chain will flash on.

I believe this is a coding problem. That somewhere along the line, its reading the wrong address / channel, turning that channel off and turning another one on (You don't always see the other one turn on because of the zero cross). Ive tried version 0022-1.0 and different pin configurations but to no avail.

So I just wanted to know your input. Oh, and im using a Duimilanove.

Copious amounts of gratitude

sorry meant to quote.

just to add that im getting the same think with only 4 leds

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2011/12/17 04:04

Yeah, that's pretty weird. Are you using the Arduino Mega files in this forum thread with Arduino IDE v022? I've tested that combination on the Arduino Mega board and it works. Otherwise, I'm not too sure what could be causing the issue.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Nate Kankiewicz - 2012/02/10 12:30

Hey everyone,

Akiba, since this is my first post, let me say thanks for all this regarding dmx.

Now, on to business. Has anyone here tried dimming with the dmx values? I started using bit angle modulation recently, but it the code here doesn't hold the current channel value until the next one arrives. This is causing a very noticeable flicker. The dimming is working, btw.

Heres a snippet of my BAM code. This goes in place of digitalWrite(dmx_pin, HIGH);
//////////

```
int period = 10; //delays are 1,2,4,8,16,32,64,128 to achieve a duty cycle like PWM.
//each bit corresponds to a specific period value.  11111111 ->100% duty cycle
if(bitRead(dmx_data,0)>0){digitalWrite(dmx_pin,HIGH);delayMicroseconds(period);}
else{digitalWrite(dmx_pin, LOW);delayMicroseconds(period);}
digitalWrite(dmx_pin, LOW);
////
```

I need to repeat the BAM cycle continuously until the next channel value can be read. Any thoughts?

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2012/02/10 18:09

For dimming, you might want to try it out with some lights on the PWM pins. That would be the easiest way to do it. Then the DMX program would pass in numerical values for the dimming and that would just map into the PWM value that you pass into analogWrite().

I haven't tried BAM yet, but I suspect that the flicker you're seeing is because the Arduino MCU is very busy processing the interrupts from the DMX stream. Those interrupts come in ~25000 times per second so it heavily loads the MCU. It shouldn't affect PWM though since it is handled in hardware and not software.

==========

What code should I comment out in the Arduino 1.0 library?

Posted by Michael Morlan - 2012/02/16 06:04

Hi Akiba,

Thanks for sharing such useful code.

I started playing with the DMX code at mathertel.de without success. (I'm using a Boarduino with USB/TTL cable.) He asserts you don't have to disable the USART in HardwareSerial.cpp as long as you don't actually make any calls to the

serial port. But, I should try. I'm going to try your code as well.

What lines should you comment out in v1.0 of the Arduino library?

Thanks,

M

==========

Re:What code should I comment out in the Arduino 1.0 library?

Posted by Akiba - 2012/02/16 06:10

I haven't tested it out, but it looks like the following needs to be commented out in v1.0 of the HardwareSerial.cpp:

```
// start comment here

#if !defined(USART0_RX_vect) && defined(USART1_RX_vect)
// do nothing - on the 32u4 the first USART is USART1
#else
#if !defined(USART_RX_vect) && !defined(SIG_USART0_RECV) && \
    !defined(SIG_UART0_RECV) && !defined(USART0_RX_vect) && \
 !defined(SIG_UART_RECV)
  #error "Don't know what the Data Received vector is called for the first UART"
#else
  void serialEvent() __attribute__((weak));
  void serialEvent() {}
  #define serialEvent_implemented
#if defined(USART_RX_vect)
  SIGNAL(USART_RX_vect)
#elif defined(SIG_USART0_RECV)
  SIGNAL(SIG_USART0_RECV)
#elif defined(SIG_UART0_RECV)
  SIGNAL(SIG_UART0_RECV)
#elif defined(USART0_RX_vect)
  SIGNAL(USART0_RX_vect)
#elif defined(SIG_UART_RECV)
  SIGNAL(SIG_UART_RECV)
#endif
  {
  #if defined(UDR0)
    unsigned char c  =  UDR0;
  #elif defined(UDR)
    unsigned char c  =  UDR;
  #else
    #error UDR not defined
  #endif
    store_char(c, &rx_buffer);
  }
#endif
#endif

// end comment here
```

==========

Re:What code should I comment out in the Arduino 1.0 library?

Posted by Michael Morlan - 2012/02/16 06:58

Thanks. I'll give it a try and report back.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Michael Morlan - 2012/02/17 17:09

Hi Akiba,

So, I must report utter failure. Meh. I'm so enthused by your results but can't seem to recreate them.

I ran your code with both Arduino .022 and 1.0 with the HardwareSerial.cpp sections commented out and experienced the same, poor results either way.

My setup is;
Vixen - Enttec Open DMX - COM1 - FTDI cable - DC Boarduino + hi/low state outputs to three 5mm LED's.

Result:
Intermittent LED blinking in no relation to the simple on/off sequence in Vixen.

In trying to debug, I enhanced the setup and code. The enhanced setup:

Vixen - Enttec Open DMX - COM1 - FTDI cable - DC Boarduino - PWM outputs - LCD display

(I also tried Freestyler with similarly incorrect results.)

Result:
Some more accurate blinking and channel values displayed but, again, nothing like the sequence in Vixen. Perhaps you can suggest some fixes. I don't have the means to snoop the serial channel further. Perhaps you could suggest a means for me to do so.

I posted a quick video of the Vixen sequence running with the Boarduino and LCD.
http://talltalepictures.com/projects/arduino_akiba_dmx.mov

Code: (please forgive the variable name changes):

See attached file: arduino_akiba_dmx.txt

Thanks for any insight you may offer.

Michael http://freaklabs.org/images/fbfiles/files/arduino_akiba_dmx.txt

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2012/02/17 20:33

Okay, I'll try to check it out this weekend.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Michael Morlan - 2012/02/18 09:17

Thanks, in advance, for taking the time.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Michael Morlan - 2012/03/22 08:17

Hi Akiba,

Thanks, again, for sharing your project. I'm not sure why I couldn't replicate your results.

I finally gave up trying to get your code to work on my setup and found success with the DMXSerial library created by Matthias Hertel over at http://www.mathertel.de/Arduino/DMXSerial.aspx.

Best,

Michael

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2012/03/22 18:42

Sorry about not getting back to you. Things got super hectic over here. But thanks for sharing the link. I'd like to check it out and see if there's anything I can learn from his project. Good luck on yours!

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Michael Morlan - 2012/03/23 08:16

You've been very generous to share your own projects. Thank you.

I now have a Boarduino driving a 2x16 LCD and TLF5940 chip. The TLF chip drives two hobby servo motors on a pan/tilt rig and a ULN2803 transistor array to a 3w RBG LED.

I have a multi-channel DMX scheme, much like intelligent moving heads, with acceleration/deceleration functions to handle smooth movement of the servos.

Having a lot of fun.

Michael

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by j reckart - 2012/03/26 06:11

Michael Morlan wrote:
You've been very generous to share your own projects. Thank you.

I now have a Boarduino driving a 2x16 LCD and TLF5940 chip. The TLF chip drives two hobby servo motors on a pan/tilt rig and a ULN2803 transistor array to a 3w RBG LED.

I have a multi-channel DMX scheme, much like intelligent moving heads, with acceleration/deceleration functions to handle smooth movement of the servos.

Having a lot of fun.

Michael

I'd be very interested in your moving light project! I have always thought of doing that. Can you post more specifics, schematics , pictures, or videos?

Jr

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Michael Morlan - 2012/03/26 09:47

Yes, of course. I'm still debugging the acceleration algorithm (which is essentially a simple, physics-based accel/de-accel routine rather than a math-based s-curve calculation.)

I'm not sure where best to post the project. Akiba, is it okay to post in this thread or do you recommend another forum?

Michael

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by j reckart - 2012/04/05 17:27

I would like to build a wall mounted DMX controller that would fit into a single gang box. I would like to have maybe four buttons on the faceplate, that i would pre-program to various "scenes" for a church, ie full on, full off, worship, prayer.

the Arduino would do this, but i have a twist. I would sometimes control the lights with full DMX console. so i would want to have the Arduino be a "middle man".

DMX Console --> Arduino wall mount controller --> wire to dimmer packs --> lights

I would want the DMX to check for an incoming signal, and if none is found act as a DMX master and "take over" sending the signal down the line, according to which scene's button was pressed.

But if it detects a signal, I would want it to "repeat" the incoming signal from my Console down the line.

Is the Arduino Robust enough, or fast enough to do this?

JR

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2012/04/05 18:42

No problem to post it here :)

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2012/04/05 18:45

I think you'd have to check to see if there is any speed issue on the Arduino. At 8 MHz, which is what the Freakduino runs at, I can decode the DMX and send/receive wireless data. For standard Arduinos, they run at 16 MHz so you can theoretically get twice the performance (or utilization) out of them. So in theory, you should be able to both decode, toggle lighting, and forward the data.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by j reckart - 2012/04/06 07:01

Is there an appropriate spot here that i can start a discussion on this? I would be using the full arduino, not the freakduino, but the folks here seem to be morte friendly and helpful that other places. I would be willing to post all info or schematics to help others also.
thanks JR

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2012/04/06 17:47

Ha ha ha. Start away. Might be good to use a new thread though. I'm actually starting up a project in Tokyo Hackerspace for DMX light sequencing for some stage work using an Arduino so I should be actively developing on the software too.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Mitch - 2012/04/22 07:51

Hi everyone. I'm new to the forum and I'm glad I've found this precious resource.

In particular, I'm planning to realize an arduino-based project involving wireless DMX lighting control.

I've consulted the tutorial above and I think it is brilliant work. Actually, I'm planning to realize this with a RFM12B transmitters (they are very cheap and support 115K baud rate, on paper at least). What I want to do is connect a PC running e.g. FreeStyler (multi-platform lighting control software) to an arduino which will then send DMX data via the RFM12 transmitter to a RFM12 receiver on another arduino board which in turn will be connected to the 3-pin DMX input of a fixture (e.g. a scanner, moving head or LED PAR).

The above I think is covered by the tutorial (with some tweaking due to the different wireless protocol). However, I'll also need to get going with the DMX "daisy-chain" to more fixtures (e.g. PC -> LED PAR -> scanner -> Laser -> moving head -> 120ohm DMX terminator). All this I'm planning to realize wirelessly.

The problem I see is in using only 1 arduino per fixture to manage both the DMX input TO the fixture and the forwarding of the DMX output FROM the same fixture to the next light in the daisy-chain.

Would in your opinion 1 arduino uno board be capable of processing both Input and Output streams? would 2 separate RFM12 transmitters per board be needed? Is it possible to go for a point-multipoint wireless solution where, instead of a "chain" of fixtures, the first arduino connected to the PC broadcasts the DMX universe separately to all receiving arduinos and lights connected thereto? This I think could be possible as I understand that the DMX protocols works by addresses so that every light only takes into account the part of the signal addressed to it and just forwards the DMX signal stream without processing.

As you see I'm in pre-alpha stage here. Any suggestion would be highly appreciated.

Also, do please move this post as appropriate.

Sorry if I've been long. Thank you very much for your appreciated attention =)

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Akiba - 2012/04/22 19:55

I don't think you'd want to decode the data and then re-encode the data to forward it. I would probably recommend tying the DMX input pins to the DMX output pins for the forwarding. Then tap off the input pins to the Arduino to decode the data. That way, you won't have to burn CPU cycles re-encoding things. Even processing 250kbps serial data is a huge burden on the Arduino.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Michael Morlan - 2012/04/23 10:01

Yes, exactly. Tap off a pass-thru DMX circuit. (Don't forget to have a 120ohm terminator on the last DMX device in the

line.)

M

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Alex - 2012/05/10 15:05

Hello.I have a question. I made the same(as on video), but on usb cord. But when i connect Xbee- light-emitting diodes stop to blink. What Wireless you used, and what setting is?
 can i do the same using Xbee? Explane plz how to make it!
I use arduino, xbee shield, and xbee 2 series

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Kevin McLoughlin - 2012/11/05 15:58

I have just made a similar setup using and arduino uno r2 and v1.01
Vixen - Enttec Open DMX - USB-RS485 (FTDI Adaptor cable) - max 485 chip output to RX pin on Uno.
I have added a line to Akib`s setup code... and left in (commented out) a couple of other things i tried. Not sure why it works but it does!
Thanks for this excellent site, i would be totally lost without the help of the software guys.

```
void setup()
{
  // set update flag idle
  update = 0;

  // set default DMX state
  dmx_state = DMX_IDLE;

  // set DMX pins to ouput and idle value
  for (int i=0; i
```

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Eddie Noyes - 2013/01/31 21:37

Maybe I'm missing something, and I feel a simple answer exists so excuse me if this is already posted on another thread.

I have the DMX receiver working with the software through the USB, but I'd like to make this a standalone device that can accept DMX input from another lighting console through a max485 chip and the pins on the arduino. What modifications need to be made in the code for this to work? I'm relatively new to digital logic design and the arduino platform itself, so any help would be greatly appreciated.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2013/02/01 00:30

The USB looks just like a serial port from the Arduino's perspective. If you want to make it standalone from a 485 transceiver, just connect the serial pins from the RS-485 chip to the Arduino's serial pins. Don't forget to terminate the RS-485 properly.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Eddie Noyes - 2013/02/01 17:06

Thanks so much. This is one of the most helpful forums I've found on the topic so far.

==========

Re:Light Sequencing and Decoding DMX with an Arduino
Posted by Eddie Noyes - 2013/03/24 21:21

I finally had the opportunity this past weekend to prototype a DMX reception circuit to dim an LED up and down using an external DMX controller. I had some issues, the main being that the LED flickers the entire time, and ch 2 on the controller decides that it also wants to control the LED. I made a video demonstrating this and offering a better explanation to my problem, any help would be great. I wasn't sure if anyone else here was having the same problem as I am. I would like to get this working to control strips of red, green, and blue, thus the reason my code has 3 channels declared.

Video:
http://www.youtube.com/watch?v=ZzbTI0iwSjU&feature=youtu.be

Code:

```
#define DMX_NUM_CHANNELS 3

enum
{
  DMX_IDLE,
  DMX_BREAK,
  DMX_START,
  DMX_RUN
};

volatile unsigned char dmx_state;

// this is the start address for the dmx frame
unsigned int dmx_start_addr = 1;

// this is the current address of the dmx frame
unsigned int dmx_addr;

// pins that will toggle based on dmx data
unsigned char dmx_pin = {9,10,11};

// this is used to keep track of the channels
unsigned int chan_cnt;

// this holds the dmx data
unsigned char dmx_data;

// tell us when to update the pins
volatile unsigned char update;

/*****************************************************************************/
/*!
  This is the setup code
*/
/*****************************************************************************/
void setup()
{
// set update flag idle
update = 0;

// set default DMX state
```

dmx_state = DMX_IDLE;

// set DMX pins to ouput and idle value
for (int i=0; i

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Eddie Noyes - 2013/03/24 21:22

I also don't want anyone to get the wrong impression, I don't claim any ownership of this code, it was all borrowed from earlier posts on this thread. I really only added the analogWrite line instead of using the digitalWrite.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2013/03/24 21:26

How many channels are you decoding?

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Eddie Noyes - 2013/03/24 21:43

Right now for testing purposes I only have 1 LED hooked up, however the code is ready to use 3 channels for RGB using PWM pins 9,10,11 on the Uno.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Danny - 2013/04/01 17:15

For starting quickly writing you arduino dmx master or slave app you might want to take a look at the following library: http://sourceforge.net/projects/dmxlibraryforar

It is a uart based library and still under development but support all basic functionality required to get running.

If you want advice on shields to use just leave me a message.

==========

Re:Light Sequencing and Decoding DMX with an Arduino

Posted by Akiba - 2013/04/01 17:56

Nice. Thanks for the link!

==========