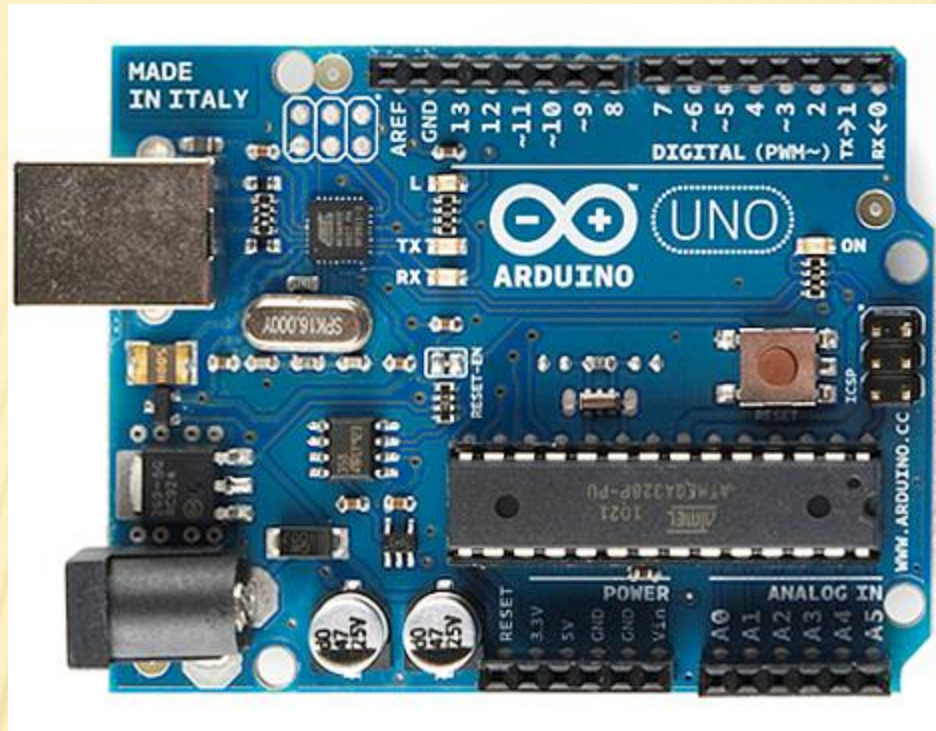




Implementing Multiple Serial Ports On An Arduino

DESERT CODE CAMP 2011.2

Presented by Don Doerres
Embedded Pro Guy
don@azlaborlaw.com

THE ARDUINO

- ✗ Arduino is Italian for “Strong Friend”
- ✗ The basic Arduino may be far more versatile than you might think

TODAY'S TAKEAWAY...

- ✖ Although the Arduino has only one hardware serial port...
- ✖ ...You can readily create additional serial ports in software
- ✖ This can be quite a cost savings over the price of a more expensive single board computer

EXAMPLE–BASIC DEBUG ON THE ARDUINO

- ✖ Basic debug for the Arduino is “old school printf’s”
- ✖ You simply print to the console from the serial port
- ✖ ...and note that we are talking about a USB-to-5V serial interface, not an old style 9-pin or 25-pin DB Connector*

* See Appendix

ONLY ONE HARDWARE SERIAL PORT

- ✗ The basic Arduino has one Hardware serial port
- ✗ If what ever you hook up to the Arduino needs the serial port, you can't print outputs to a console...
- ✗ ...or can you?

ENTER SOFTWARE SERIAL

- ✗ The Arduino is plenty fast enough and has plenty of I/O pins to perform what is called “Software Serial”
- ✗ The basic Arduino package even comes with a SW serial library, `SoftwareSerial`
- ✗ ...the stock library is not entirely optimal

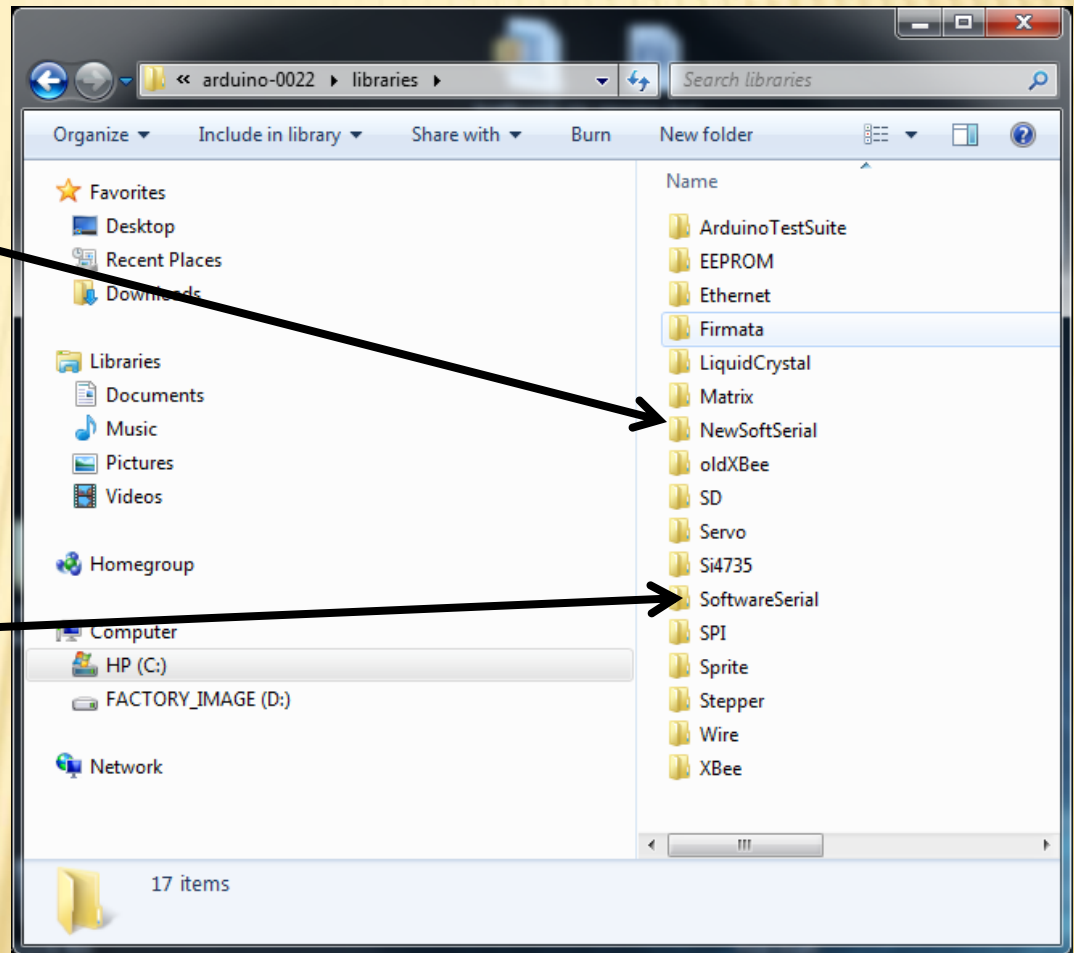
BETTER SOFTWARE SERIAL

- ✗ <http://arduiniana.org/libraries/newsoftserial/>
- ✗ A key advantage is that this library is interrupt driven rather than polled
- ✗ Other virtues are extolled on the web page
- ✗ Installation is simple—just copy the NewSoftSerial library to your Arduino directory

THE SOFTWARESERIAL LIBRARIES

NewSoftSerial

Stock
SoftwareSerial

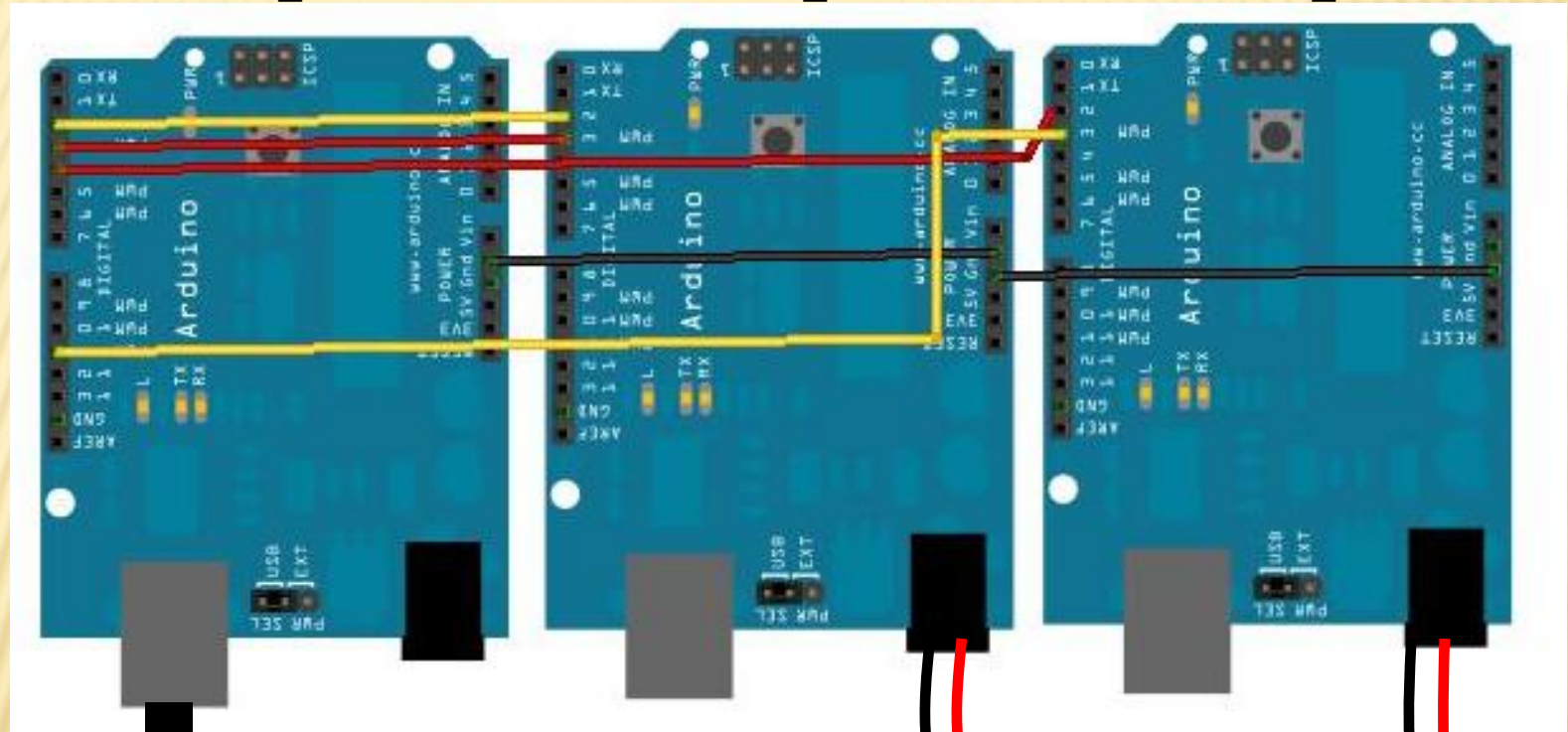


TEST CONFIGURATION THREE ARDUINOS

UNO_1

UNO_2

UNO_3



USB (includes power)
from Control Computer

External Adafruit Power Boxes

PIN LIST

If the Fritz figure isn't clear,
here is the pin list

Signal	Uno_1	Uno_2	Uno_3	Color
Input to Uno_1, Output from Uno_2	D2	D2		Yellow
Output from Uno_1, Input to Uno_2	D3	D3		Red
Input to Uno_1, Output from Uno_3	D4		D2	Red
Output from Uno_1, Input to Uno_3	D11		D3	Yellow
	GND	GND	GND	Black

BAUD RATE TABLE

✖ Table from NewSoftSerial.cpp

```
static const DELAY_TABLE PROGMEM table[] =  
{  
    //  baud      rxcenter  rxintra  rxstop  tx  
    { 115200,    3,        21,      21,     18,     },  
    { 57600,     20,       43,      43,     41,     },  
    { 38400,     37,       73,      73,     70,     },  
    { 31250,     45,       89,      89,     88,     },  
    { 28800,     46,       98,      98,     95,     },  
    { 19200,     71,      148,     148,    145,    },  
    { 14400,     96,      197,     197,    194,    },  
    { 9600,      146,     297,     297,    294,    },  
    { 4800,      296,     595,     595,    592,    },  
    { 2400,      592,    1189,    1189,   1186,   },  
    { 1200,     1187,    2379,    2379,   2376,   },  
    { 300,      4759,    9523,    9523,   9520,   },  
};
```

BAUD RATE LIMITATIONS

- ✘ With HW serial, I've had good luck to 57.6Kbaud
- ✘ With SW serial, I've had good luck to 38.4 Kbaud

MORE ON TEST CONFIGURATION

- ✗ I just happen to have three Uno's
- ✗ One will be set up as the controller
 - + USB power
 - + USB serial interface at 9600 baud
- ✗ Two Uno's will have DC power
 - + NewSoftSerial interface only!
- ✗ This whole set up will be controlled from Tera Term

THE SKETCH FOR UNO_1 PARSED

- ✗ Standard Arduino Sketch
- ✗ Three parts:
 - + #includes,
 - + setup(),
 - + loop()

```
#include's <NewSoftSerial.h> // the include file
#define's
Serial port instantiation
void setup()
Void loop()
```

SETTING UP UNO_1

- ✗ #include the header for the library
- ✗ Pin order on the board is not important
- ✗ Note the order in the class setup → in, out

```
#include <NewSoftSerial.h> // the include file required

#define INSOFT1    2
#define OUTSOFT1   3

#define INSOFT2    11
#define OUTSOFT2   4
// special handling function prototypes...
void queryArduino_2(unsigned char x);
void queryArduino_3(unsigned char y);
// NewSoftSerial port instantiation—note order of pins!
NewSoftSerial SWSerial1(INSOFT1, OUTSOFT1);
NewSoftSerial SWSerial2(INSOFT2, OUTSOFT2);
```

SKETCH SETUP() SECTION UNO_1

- ✗ Set up the baud rates for each port
- ✗ They may be different

```
void setup()
{
  // All we need to set here is the baud rate
  // For fun, let's make them all different
    Serial.begin(9600); // good to at least 57600
  SWSerial1.begin(19200); // good to at least 38400
  SWSerial2.begin(38400); // good to at least 38400
  // For testing, I like a message to the console...
  Serial.println("Starting up Uno_1...");
}
```

SKETCH LOOP() SECTION 1 UNO_1

```
void loop()                                // run over and over
    again
{
/*
We will endlessly loop for input
on the hardware serial port
*/
if( Serial.available() )
{
    char ch = Serial.read();
    Serial.flush(); // may be CR or LF, don't care
                    // just flush() them away
    Serial.print("Uno_1 got letter '");
    Serial.print(ch, BYTE);
    Serial.println("' on the hardware serial port...");
    Serial.println("Looking up what to do with it...");
    //...continue
}
```

SKETCH LOOP() SECTION 2 UNO_1

```
switch(ch)
{
  case 'a':
  case 'A':
    queryArduino_2('a');
  break;
  case 'b':
  case 'B':
    queryArduino_3('b');
  break;
  case 'c':
  case 'C':
    queryArduino_2('c');
  break;
  default:
    Serial.print("Uno_1 says character '");
    Serial.print(ch, BYTE);
    Serial.println("' is not a usable control character");
  break;
} // end switch

Serial.flush(); // may be CR or LF, don't care
               // just flush() them away

} // end if
} // end loop
```

UNO_2 SETUP()

```
#define INSOF1 3
#define OUTSOFT1 2

#include <NewSoftSerial.h> // the include file required

NewSoftSerial SWSerial1(INSOF1, OUTSOFT1);

void setup()
{
// All we need to set here is the baud rate
Serial.begin(9600); // good to at least 57600
SWSerial1.begin(19200); // good to at least 38400
Serial.println("Starting Uno_2...\n");
}
```

UNO_2 LOOP()

```
void loop()    // run over and over again
{
  static unsigned int dataRequest = 0;  // static so we can retain value as we run
  if( SWSerial1.available() ) {
    char ch = SWSerial1.read();
    if( ch == 'a' ) {
      SWSerial1.print("Uno_2 'a' response number ");
      SWSerial1.print(dataRequest++);
      SWSerial1.println(" via software serial port");
    }
    else if ( ch == 'c' ) {
      SWSerial1.print('f', BYTE);  SWSerial1.print('o', BYTE); SWSerial1.print('u', BYTE);
      SWSerial1.print('r', BYTE);
    }
    else {
      SWSerial1.print("The character '");
      SWSerial1.print(ch, BYTE);
      SWSerial1.println("' is not an Uno_2 usable control character");
    }
    SWSerial1.flush(); // may be CR or LF, don't care just flush() them away
  } // end outer if
} // end loop
```

UNO_3 SETUP

```
#define INSOF1 3
#define OUTSOFT1 2

#include <NewSoftSerial.h> // the include file required

NewSoftSerial SWSerial1(INSOF1, OUTSOFT1);

void setup()
{
// All we need to set here is the baud rate
Serial.begin(9600); // good to at least 57600
SWSerial1.begin(38400); // good to at least 38400
Serial.println("Starting Uno_3...\n");
}
```

UNO_3 LOOP

```
void loop()                                // run over and over again
{
  static unsigned int dataRequest = 0; // static to retain value between runs

  if( SWSerial1.available() )
  {
    char ch = SWSerial1.read();

    if( (ch == 'b') )
    {
      SWSerial1.print("Uno_3 'b' response number ");
      SWSerial1.print(dataRequest++);
      SWSerial1.println(" via software serial port");
    }
    else
    {
      SWSerial1.print("The character '");
      SWSerial1.print(ch, BYTE);
      SWSerial1.println("' is not an Uno_3 usable control character");
    }

    SWSerial1.flush(); // may be CR or LF, don't care
                      // just flush() them away
  }
}
```

IN CONCLUSION

- ✖ You can use multiple serial ports on an Arduino by implementing them in software
- ✖ I hope this simple example gives you ideas you can use in your projects.

SOURCE CODE FOLLOWS

- ✖ The next set of slides is the source code in detail for the three Arduinos
- ✖ The very last slide is the Appendix

BEER-WARE.H

- ✗ This code is released under the BEER-WARE license
- ✗ Applies to Uno_1.pde, Uno_2.pde, and Uno_3.pde

```
/*
 * Released under THE BEER-WARE LICENSE:
 * -----
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <don@azlaborlaw.com> wrote this file. As long as you retain
 * this notice you can do whatever you want with this stuff.
 * If we meet some day, and you think this stuff is worth it,
 * you can buy me a beer in return
 * Don Doerres
 * -----
 */
```

UNO_1.PDE PART 1 OF 5

```
/* Uno_1, COM7 on Don's computer */  
/*
```

```
NewSoftSerial example code for Desert Code Camp 2011.2
```

```
This code uses three Arduino's. Mine just happen to be Uno's,  
just about any variant should work. I've run it on Duemilanove's.  
This code is the Uno_1 for the demo. It uses the Uno_1's  
hardware serial port to talk to the control PC. It use one NewSoftSerial  
to talk to a NewSoftSerial port on Uno_2, and one NewSoftSerial port to talk to  
Uno_3.
```

```
Read lots about NewSoftSerial here:
```

```
http://arduiniiana.org/libraries/newsoftserial/
```

```
Down load NewSoftSerial from here:
```

```
http://arduiniiana.org/NewSoftSerial/NewSoftSerial10c.zip
```

```
*/
```

```
#include <NewSoftSerial.h> // the include file required
```

```
// Any of the digital lines will do--
```

```
// order is not important
```

```
#define INSOF1 2
```

```
#define OUTSOFT1 3
```

```
#define INSOF2 11
```

```
#define OUTSOFT2 4
```

```
void queryArduino_2(unsigned char x);
```

```
void queryArduino_3(unsigned char y);
```

```
NewSoftSerial SWSerial1(INSOF1, OUTSOFT1); // order is input, output
```

```
NewSoftSerial SWSerial2(INSOF2, OUTSOFT2);
```

UNO_1.PDE PART 2 OF 5

```
void setup()
{
  // All we need to set here is the baud rate
  // For fun, let's make them all different
  Serial.begin(9600); // good to at least 57600
  SWSerial1.begin(19200); // good to at least 38400, hooks up to Uno_2
  SWSerial2.begin(38400); // good to at least 38400, hooks up to Uno_3

  Serial.println("Starting up Uno_1...");
}
```

UNO_1.PDE PART 3 OF 5

```
void loop() // run over and over again
{
  /* We will endlessly loop for input on the hardware serial port */
  if( Serial.available() )
  {
    char ch = Serial.read();
    Serial.flush(); // may be CR or LF, don't care just flush() them away
    Serial.print("Uno_1 got letter '");
    Serial.print(ch, BYTE);
    Serial.println("' on the hardware serial port...");
    Serial.println("Looking up what to do with it...");
    switch(ch)
    {
      case 'a':
      case 'A':
        queryArduino_2('a');
        break;
      case 'b':
      case 'B':
        queryArduino_3('b');
        break;
      case 'c':
      case 'C':
        queryArduino_2('c');
        break;
      default:
        Serial.print("Uno_1 says character '");
        Serial.print(ch, BYTE);
        Serial.println("' is not a usable control character");
        break;
    } // end switch
    Serial.flush(); // may be CR or LF, don't care, just flush() them away
  } // end if
}
```

UNO_1.PDE PART 4 OF 5

```
void queryArduino_2(unsigned char x)
{
  SWSerial1.print(x);

  while( !SWSerial1.available() )
    { ; }

  switch(x)
  {
    case 'a':
      while( SWSerial1.available() )
      {
        char softCh = SWSerial1.read();
        Serial.print(softCh, BYTE);
      }
      break;
    case 'c': // waiting for four characters
      while ( SWSerial1.available() < 4 )
      {
        ;
      }
      while( SWSerial1.available() )
      {
        char softCh = SWSerial1.read();
        Serial.print(softCh, BYTE);
      }
      Serial.println("\r\nReceived four characters from Uno_2");
      break;
    default:
      break;
  }
  Serial.println("");
}
```

UNO_1.PDE PART 5 OF 5

```
void queryArduino_3(unsigned char y)
{
  int delayCounter = 0;
  char softCh = 0;

  SWSerial2.print(y);

  while( !SWSerial2.available() )
    { ; }

  switch(y)
  {
    case 'b':
      while( SWSerial2.available() )
      {
        softCh = SWSerial2.read();
        Serial.print(softCh, BYTE);
      }
      break;
    default:
      break;
  }
  Serial.println("");
}
```

UNO_2.PDE PART 1 OF 2

```
/* Uno_2, COM11 on Don's computer */
/*
NewSoftSerial example code for Desert Code Camp 2011.2

This code uses three Arduino's. Mine just happen to be Uno's,
just about any variant should work. I've run it on Duemilanove's.
This code is the Uno_2 file for the demo. It uses the Uno_1's
hardware serial port to talk to the control PC. It use one NewSoftSerial
to talk to a NewSoftSerial port on Uno_2, and one NewSoftSerial port to talk to
Uno_3.
*/

#define INSOF1 3
#define OUTSOFT1 2

#include <NewSoftSerial.h> // the include file required

NewSoftSerial SWSerial1(INSOF1, OUTSOFT1); // note order of input, output

void setup()
{
// All we need to set here is the baud rate
  Serial.begin(9600); // good to at least 57600
  SWSerial1.begin(19200); // good to at least 38400
  Serial.println("Starting Uno_2...\n");
}
```

UNO_2.PDE PART 2 OF 2

```
void loop()    // run over and over again
{
  static unsigned int dataRequest = 0; // static to retain value between runs

  if( SWSerial1.available() )
  {
    char ch = SWSerial1.read();

    if( ch == 'a' )
    {
      SWSerial1.print("Uno_2 'a' response number ");
      SWSerial1.print(dataRequest++);
      SWSerial1.println(" via software serial port");
    }
    else if ( ch == 'c' )
    {
      SWSerial1.print('f', BYTE);
      SWSerial1.print('o', BYTE);
      SWSerial1.print('u', BYTE);
      SWSerial1.print('r', BYTE);
    }
    else
    {
      SWSerial1.print("The character '");
      SWSerial1.print(ch, BYTE);
      SWSerial1.println("' is not an Uno_2 usable control character");
    }

    SWSerial1.flush(); // may be CR or LF, don't care
                      // just flush() them away
  }
}
```

UNO_3.PDE PART 1 OF 2

```
/* Uno_3, COM13 on Don's computer */  
/*
```

```
NewSoftSerial example code for Desert Code Camp 2011.2
```

This code uses three Arduino's. Mine just happen to be Uno's, just about any variant should work. I've run it on Duemilanove's. This code is the Uno_3 file for the demo. It uses the Uno_1's hardware serial port to talk to the control PC. It use one NewSoftSerial to talk to a NewSoftSerial port on Uno_2, and one NewSoftSerial port to talk to Uno_3.

```
#define INSOF1 3  
#define OUTSOFT1 2
```

```
#include <NewSoftSerial.h> // the include file required
```

```
NewSoftSerial SWSerial1(INSOF1, OUTSOFT1); // note order of input, output
```

```
void setup()  
{  
  // All we need to set here is the baud rate  
  Serial.begin(9600); // good to at least 57600  
  SWSerial1.begin(38400); // good to at least 38400  
  Serial.println("Starting Uno_3...\n");  
}
```

UNO_3.PDE PART 2 OF 2

```
void loop() // run over and over again
{
  static unsigned int dataRequest = 0; // static to retain value between runs

  if( SWSerial1.available() )
  {
    char ch = SWSerial1.read();

    if( (ch == 'b') )
    {
      SWSerial1.print("Uno_3 'b' response number ");
      SWSerial1.print(dataRequest++);
      SWSerial1.println(" via software serial port");
    }
    else
    {
      SWSerial1.print("The character '");
      SWSerial1.print(ch, BYTE);
      SWSerial1.println("' is not an Uno_3 usable control character");
    }

    SWSerial1.flush(); // may be CR or LF, don't care
                      // just flush() them away
  }
}
```

APPENDIX

- ✗ The Arduino uses 0 to 5V logic
- ✗ Standard RS-232 is +/-12 V logic
- ✗ Level Shifters are available
- ✗ For example: <http://www.sparkfun.com/products/449>
- ✗ Contact Don if you would like to see a presentation on this item

