

[Atmega\\_Board\\_Programmer.zip](#)と[Atmega\\_Fuse\\_Calculator.zip](#) をダウンロードし、解凍して適切な場所に置きます。

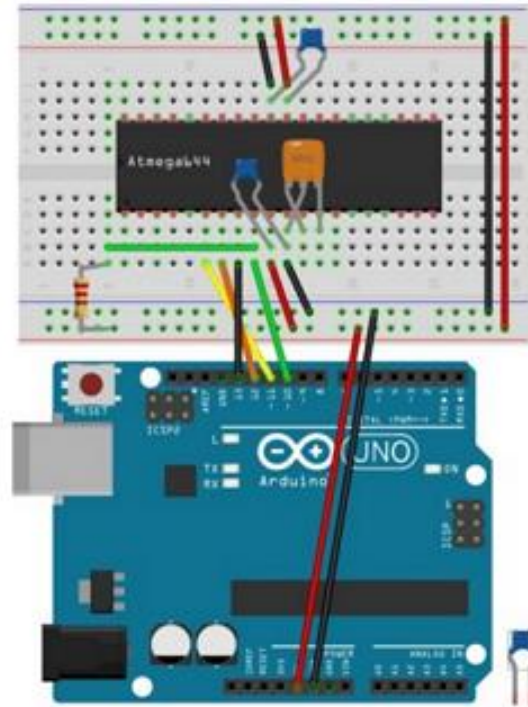
次に、以下のように Atmega\_Board\_Programmer.ino を変更しておきます。avrdude.conf も、要変更とありますが、これだけで書き込みが可能でした。

#### 以下WEBページ抜粋

Atmega\_Board\_Programmer.ino ファイルを

```
// ATmega1284P family { { 0x1E, 0x97, 0x05 }, "ATmega1284P",  
    128 * kb, 1 * kb, optiboot_atmega1284p_hex, 0x1FC00,  
// start address sizeof optiboot_atmega1284p_hex, 256,  
// page size (for committing) [変更前] 0xFF,  
// fuse low byte: external clock, max start-up time [変更後] 0xF7,  
// fuse low byte: external clock, max start-up time
```

以下の通り、配線します..



16MHz水晶発振子, 22pFセラミックコンデンサでも可能です。

参照ページ.

<http://d.hatena.ne.jp/SriVidyut/20141106/p1>

記述の通りに実行して、書き込むことができました。ここでは、重複となりますが筆者のメモも兼ねて、記述しておきます。(大変感謝致します。)

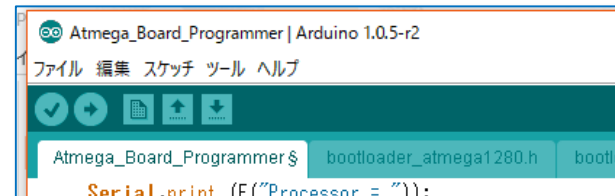
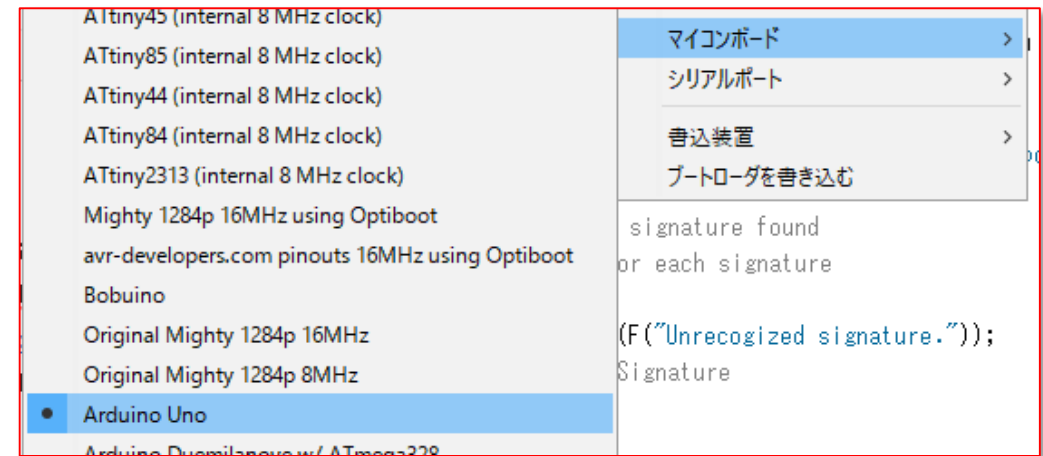
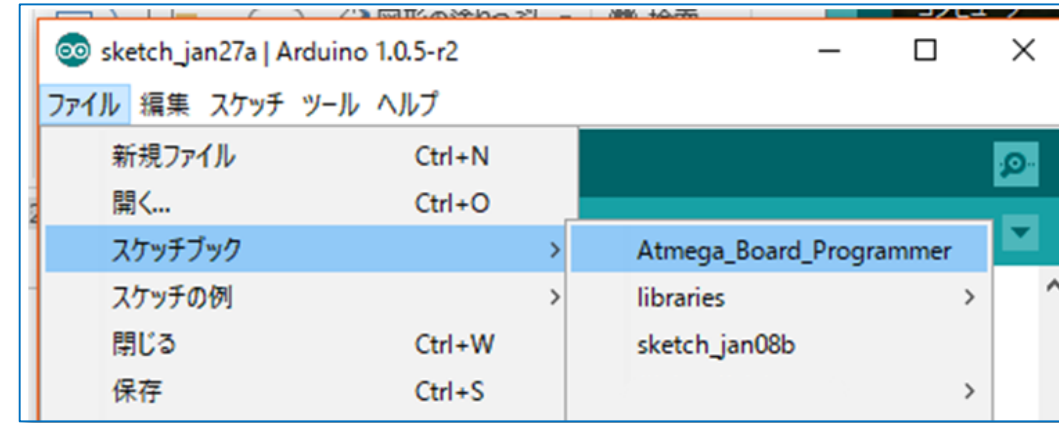
次は、ブートローダ書き込みです。

## —— Arduino Unoを用いたブートローダ書き込み ——

### 書き込み手順:

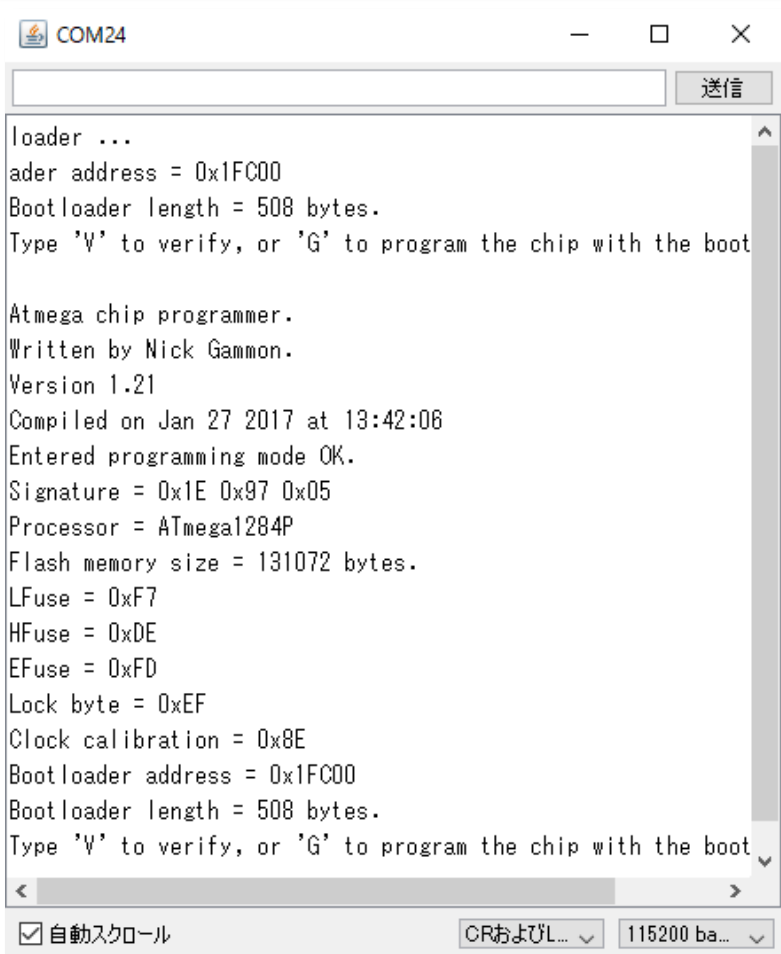
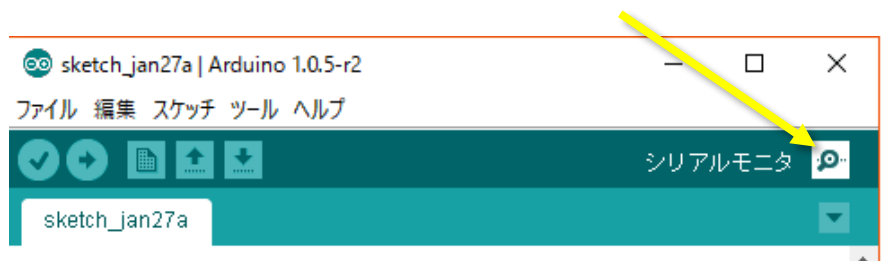
1. IDE (ここではv1.0.5-r2)を立ち上げ, “スケッチブック”から, Atmega\_Board\_Programmer.ino を開きます.
2. “ツール”, “マイコンボード”で, “Arduino Uno”を選択します.
3. “ツール”, “シリアルポート”で, 適切なCOMポートを選択します.
4. “ツール”, “書き込装置”で, “Arduino as ISP”を選択します.  
(通常のスケッチ書き込み手順と同様です)
5. Atmega\_Board\_Programmer.inoを書き込みます.

以上で, Arduino Uno に1284Pのブートプログラム  
書き込み用のスケッチが書き込まれます.  
直後に, このスケッチがスタートします.



## — Arduino Unoを用いたブートローダ書き込み —

6. シリアルモニタを開き、ボーレートを115200bpsとします。

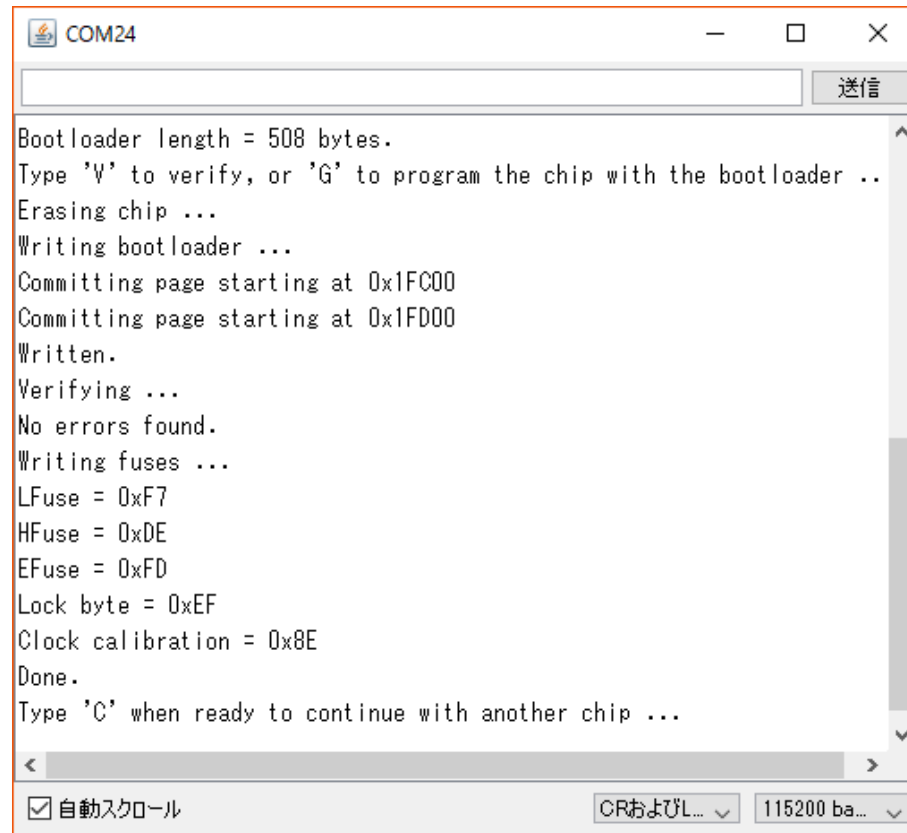


```
loader ...
ader address = 0x1FC00
Bootloader length = 508 bytes.
Type 'V' to verify, or 'G' to program the chip with the boot

Atmega chip programmer.
Written by Nick Gammon.
Version 1.21
Compiled on Jan 27 2017 at 13:42:06
Entered programming mode OK.
Signature = 0x1E 0x97 0x05
Processor = ATmega1284P
Flash memory size = 131072 bytes.
LFuse = 0xF7
HFuse = 0xDE
EFuse = 0xFD
Lock byte = 0xEF
Clock calibration = 0x8E
Bootloader address = 0x1FC00
Bootloader length = 508 bytes.
Type 'V' to verify, or 'G' to program the chip with the boot
```

7. “G”を入力します。

ブートローダが一瞬で書き込まれます。508 bytesですから...



```
Bootloader length = 508 bytes.
Type 'V' to verify, or 'G' to program the chip with the bootloader ..
Erasing chip ...
Writing bootloader ...
Committing page starting at 0x1FC00
Committing page starting at 0x1FD00
Written.
Verifying ...
No errors found.
Writing fuses ...
LFuse = 0xF7
HFuse = 0xDE
EFuse = 0xFD
Lock byte = 0xEF
Clock calibration = 0x8E
Done.
Type 'C' when ready to continue with another chip ...
```

“V”などの実行は任意です (Verify)。以降, Arduino Unoは不要なので取り外します。

次は, シリアルケーブル(ここではFTDI USB-シリアル 5V)を使ってスケッチを書き込みます。

# ATmega1284P-PU スケッチ

— USBシリアル変換ケーブルのドライバ共存のための Windows 設定 —

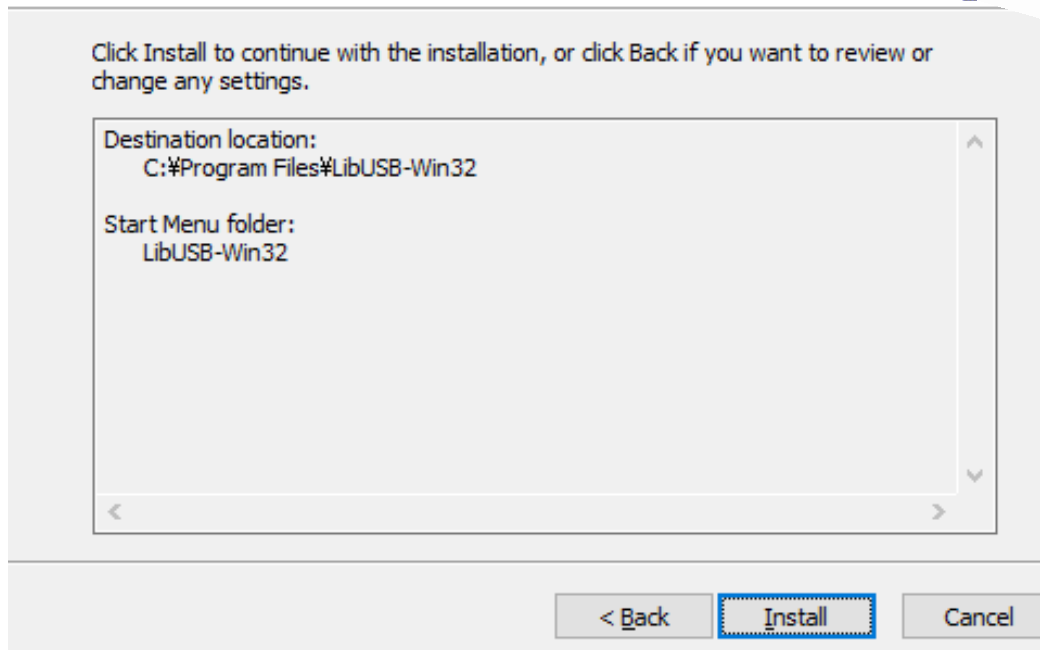
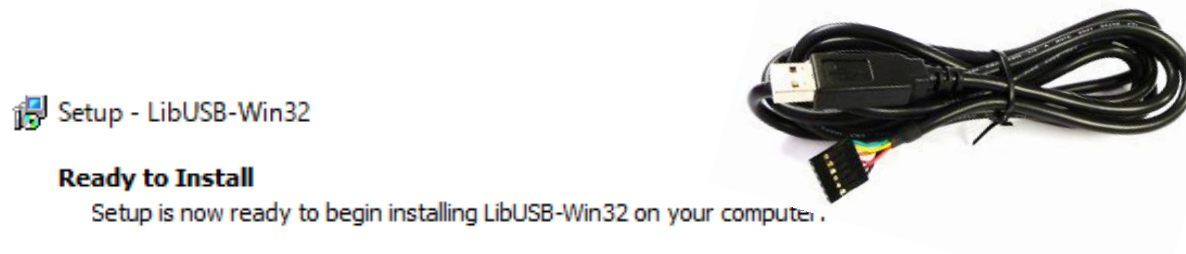
“USBドライバ” をインストールします。

libusb-win32-devel-filter-1.2.6.0

Windows のドライバと共存させるために必要だそうです。

但し、**IDEバージョンは、1.0.x.x に限ります**

<http://d.hatena.ne.jp/SriVidyut/touch/20141106/p1>



IDEバージョン：1.0.5-r2 を起動します。

“FTDI USB・シリアル変換ケーブル(5V)” のポート番号を確認します。

