

マイコンをはじめよう

パソコンと通信しよう

徳島大学大学院ソシオテクノサイエンス研究部
技術専門職員 辻 明典

連絡先：

770-8506 徳島市南常三島町2-1

TEL/FAX： 088-656-7485

E-mail: : a-tsuji@is.tokushima-u.ac.jp

パソコンと通信しよう

第4回

2013/8/10(Sat) 10:00—11:30

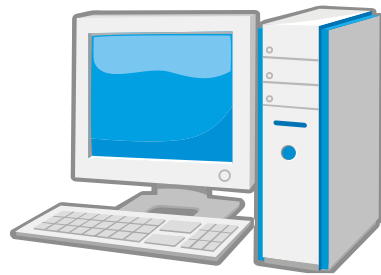
本日の予定

- 1 シリアル通信
- 2 シリアル通信の基本スケッチ
 - 送信
 - 受信
 - 送受信
- 3 LEDを調光する（送信）
- 4 スイッチの状態を調べる（受信）
- 5 パソコンからLEDを光らせる（応用）

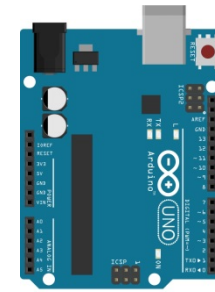
1 シリアル通信

1 シリアル通信

パソコンとマイコンの間で、データをやりとり



パソコン



マイコン

あらかじめ決めておくこと
通信の速度，向き，データ

1.1 シリアル通信の速度

- シリアル通信

8ビット（1バイト）のデータ，スタート・ストップビットで構成



- 通信速度

bps : bit per second 1秒間に送れるデータ量

- 送受信ポート

RX : Receiver 受信

TX : Transmitter 送信

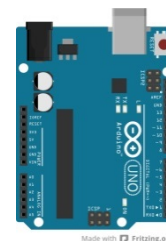
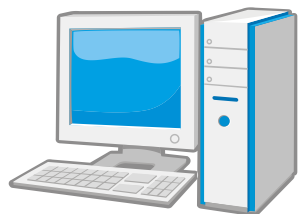
1.2 シリアル通信の向き

パソコン側

方向

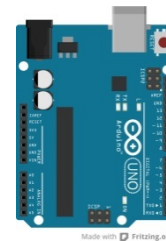
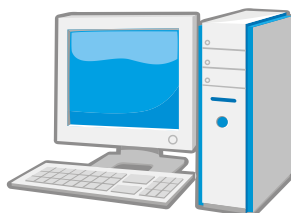
マイコン側

① 受信



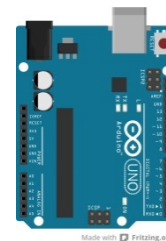
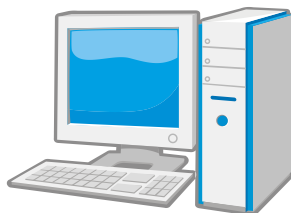
送信

② 送信



受信

③ 送受信



送受信

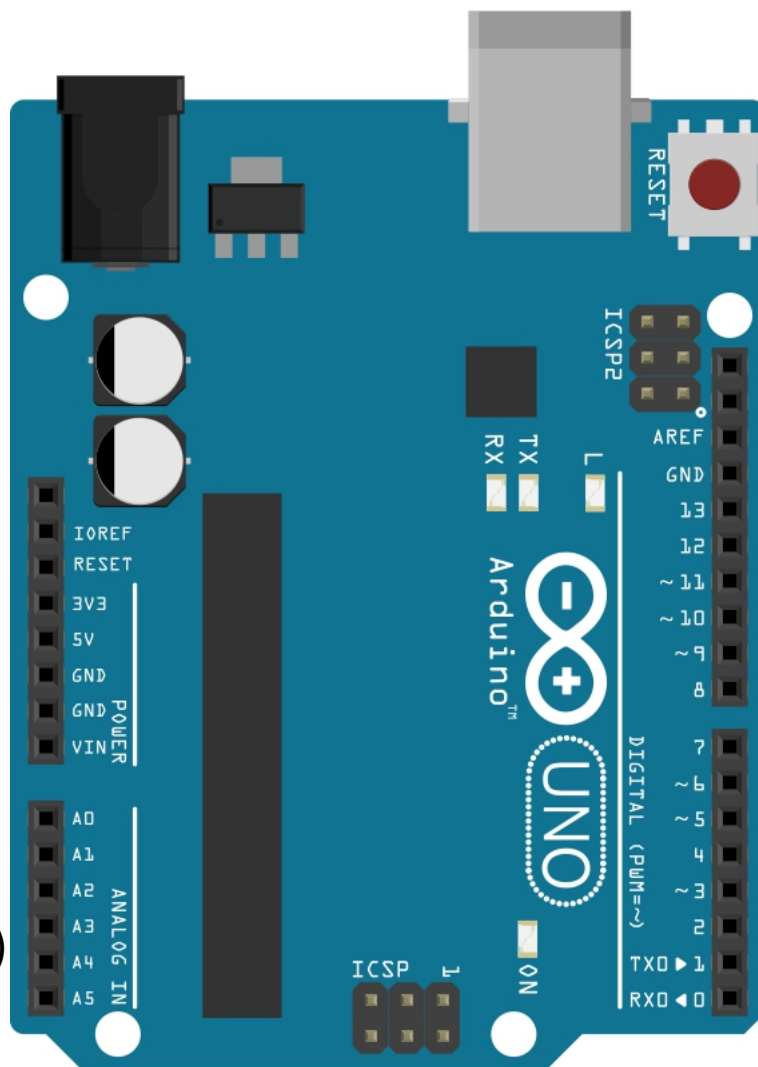
1.3 シリアル通信の種類

Arduino UNOで使用する
できる通信プロトコル：

- シリアル通信（共用）
→ USBに接続
- SPI通信
- I2C通信

I2C通信

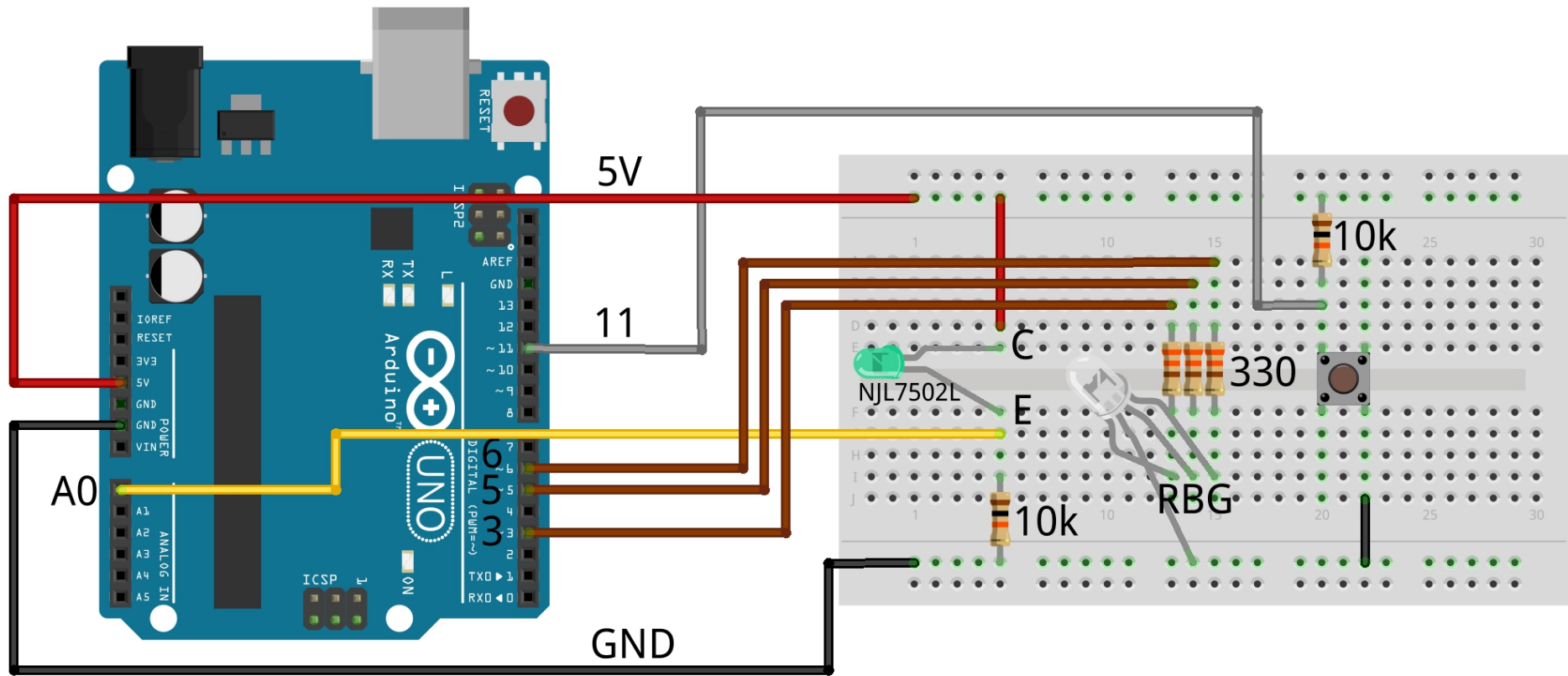
SDA(A4)
SCL(A5)



SCK(13)
MISO(12) SPI通信
MOSI(11)
SS(10)

TXD(1) シリアル通信
RXD(0)

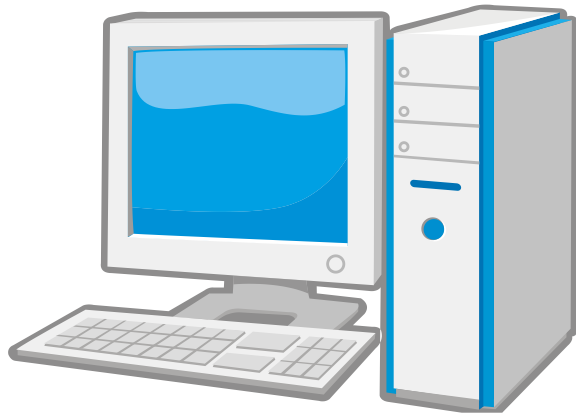
1.4 ブレッドボードの回路作成



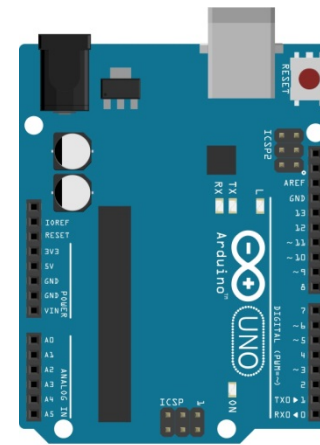
Made with  Fritzing.org

2 シリアル通信のスケッチ

- 送信
- 受信
- 送受信

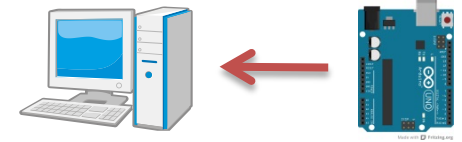


送信



Made with  Fritzing.org

2.1 シリアル通信（送信）



- マイコンからパソコンへデータ送信

setup()
通信速度設定

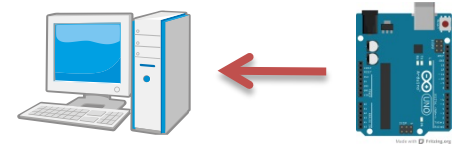
```
Serial.begin(9600);
```

loop()
文字を送信

```
Serial.println("Hello");
```

A blue circular arrow on the right side of the orange box, indicating that the code inside the box is repeated in a loop.

E4.1 シリアル通信（送信）

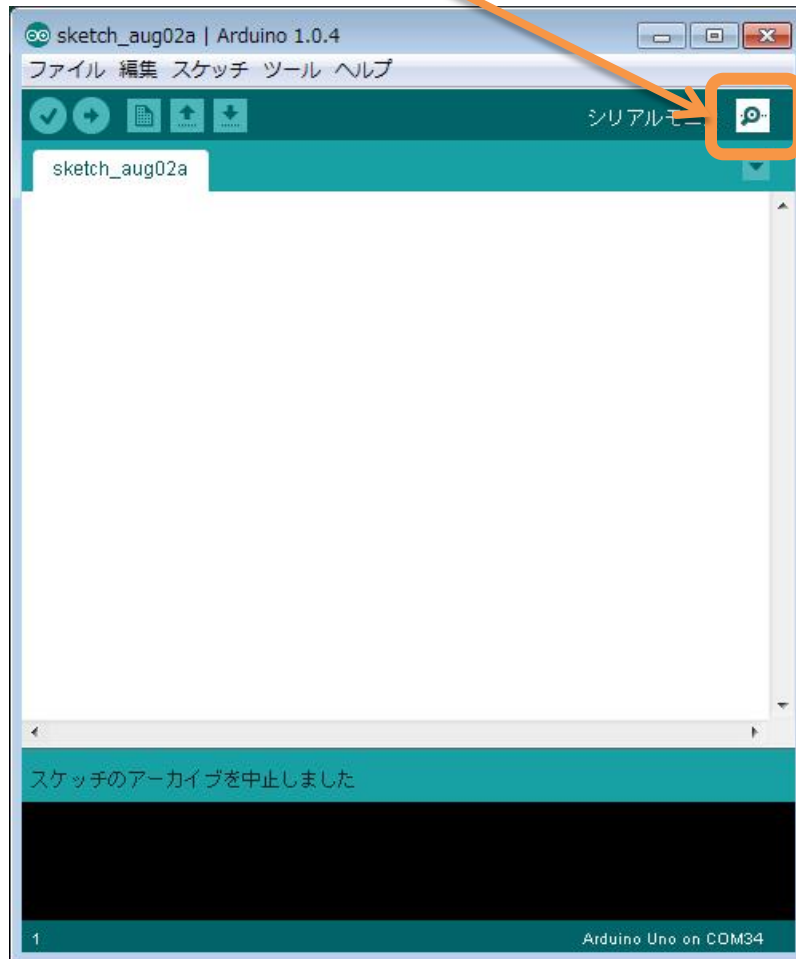


```
void setup() {  
    Serial.begin(9600);  
}
```

```
void loop() {  
    Serial.println("Hello");  
}
```

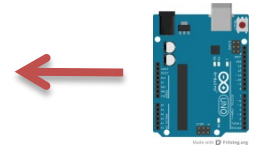
シリアルモニタ

パソコンとArduino間の通信状況を確認



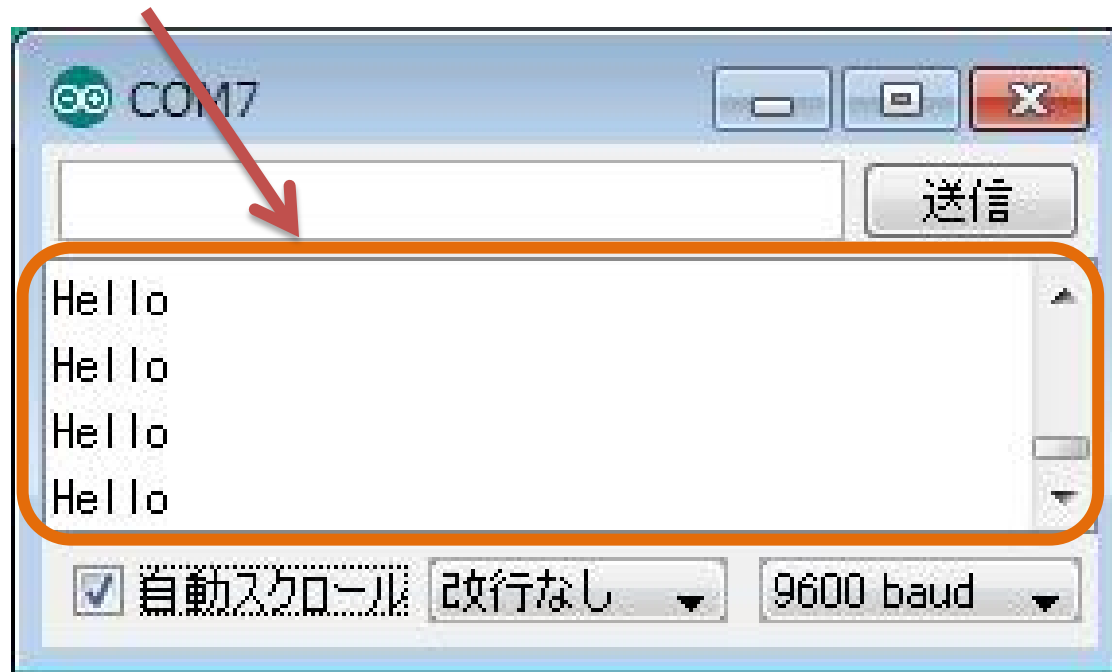
シリアルモニタ画面

E4.1 シリアル通信（送信）

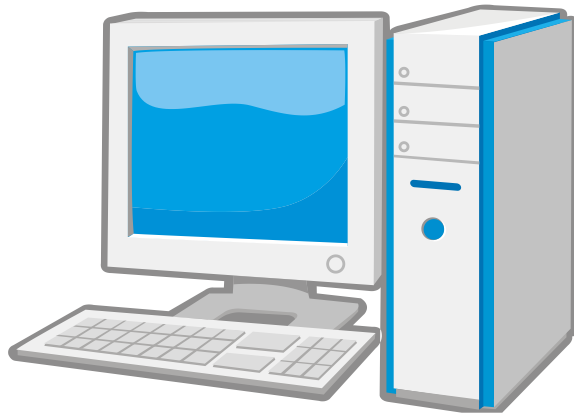


- パソコン上で、マイコンから受信した文字を確認

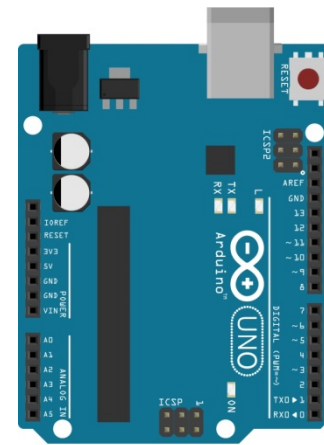
受信した文字が表示される



シリアルモニタ

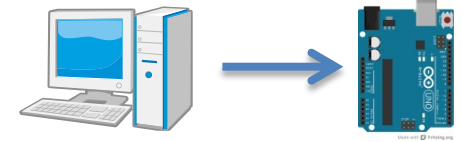


受信



Made with  Fritzing.org

2.2 シリアル通信（受信）



- パソコンからマイコンへ

setup()

通信速度設定

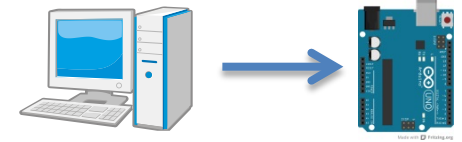
```
Serial.begin(9600);
```

loop()

データの受信確認
データの読み込み

```
if (Serial.available() > 0) {  
  byte data = Serial.read();  
  // 処理  
}
```

E4.2 シリアル通信（受信）



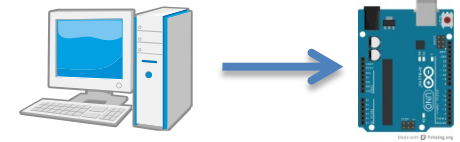
```
int ledPin = 3; // 5, 6
void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}
void loop() {
    if (Serial.available() > 0) {
        byte data = Serial.read();
        if (data == 'H') digitalWrite(ledPin, HIGH);
        else digitalWrite(ledPin, LOW);
    }
}
```

データを受信したか？

データの取り込み

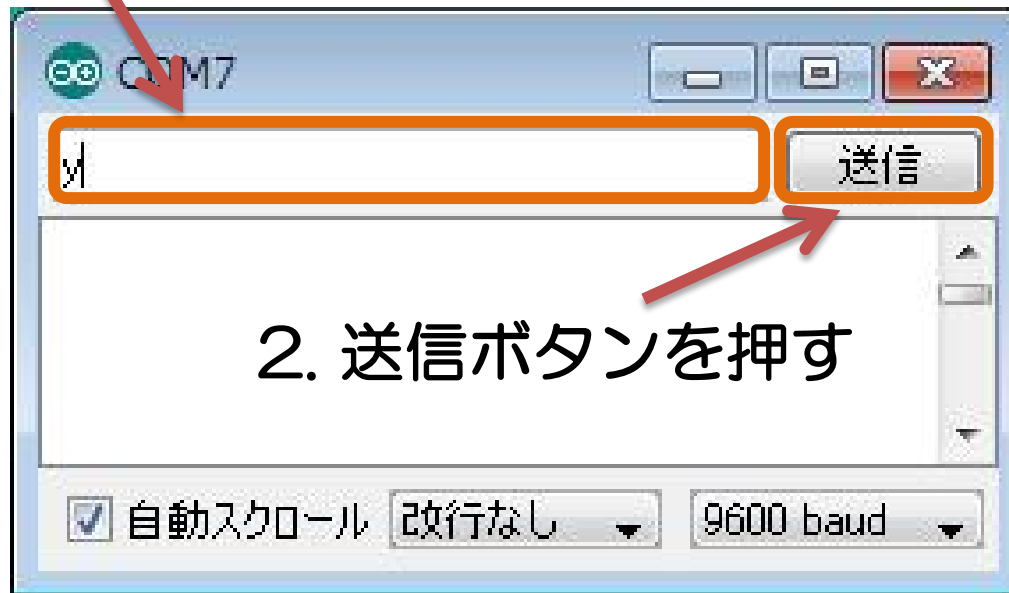
- ファイル→スケッチの例→04.Communication
→PhysicalPixel

E4.2 シリアル通信（受信）



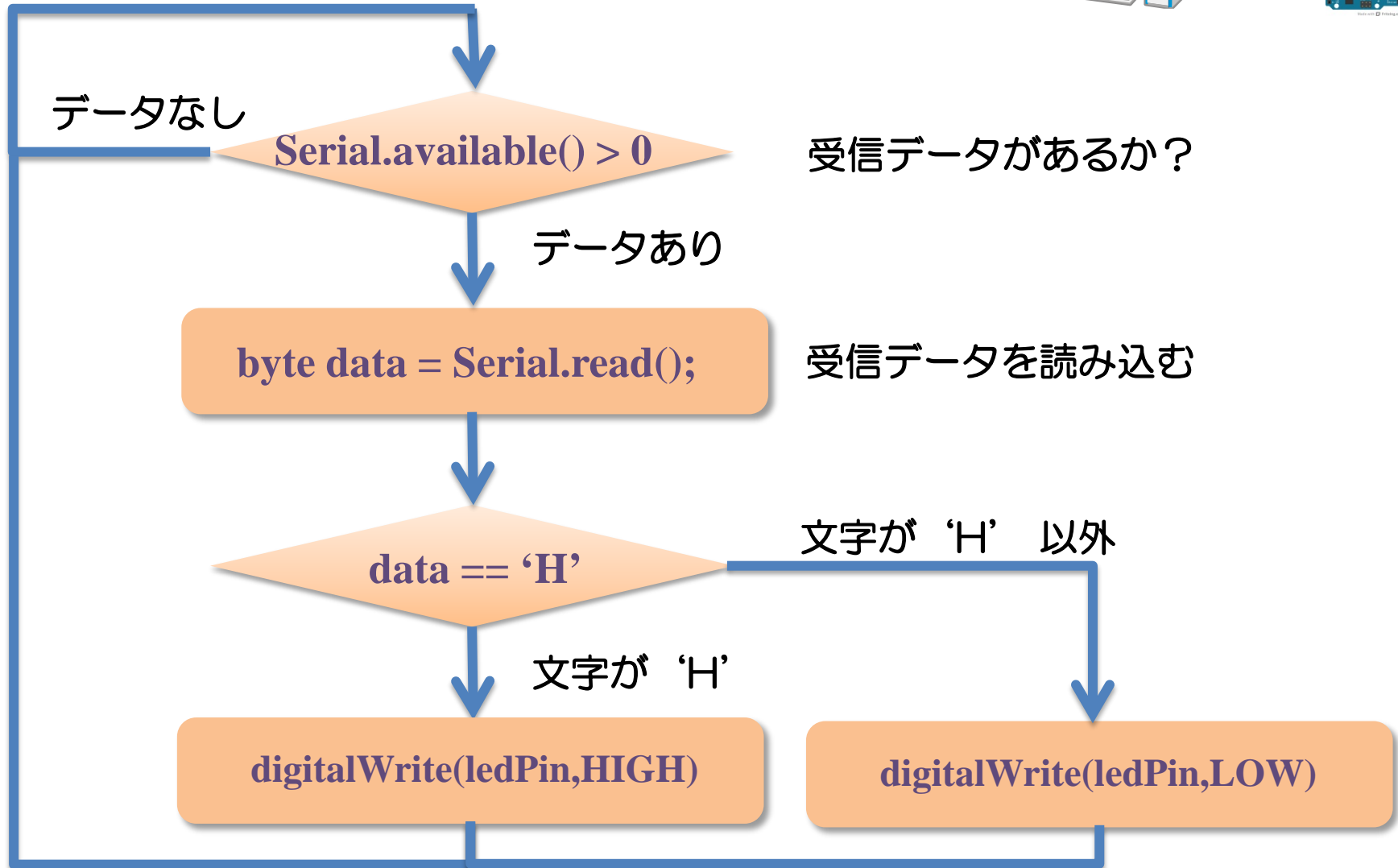
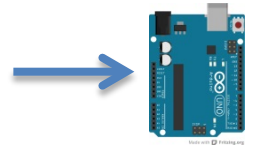
- パソコンからマイコンへ文字を送信

1. 送信したい文字を入力



シリアルモニタ

E4.2 シリアル通信（受信）



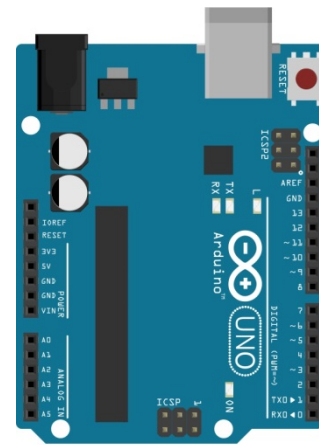
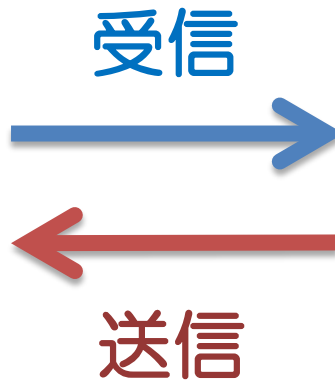
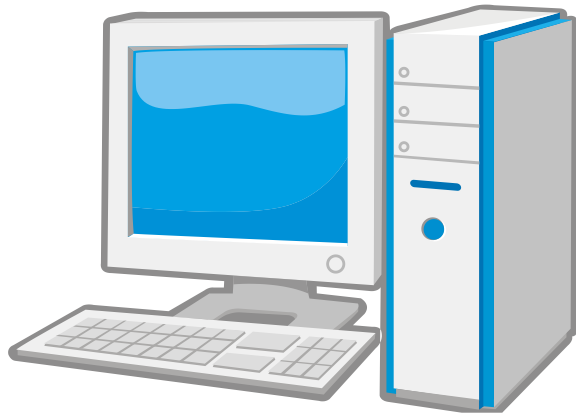
E4.2.1 シリアル通信応用（受信）

- 文字Rを受信したとき、赤のLED（3番ピン）を点灯、
 - 文字Gを受信したとき、緑のLED（5番ピン）を点灯、
 - 文字Bを受信したとき、青のLED（6番ピン）を点灯、
 - 文字Cを受信したとき、すべてを消灯
- するには、どうすればよいでしょう？

ヒント：• LED赤，青，緑のピン番号

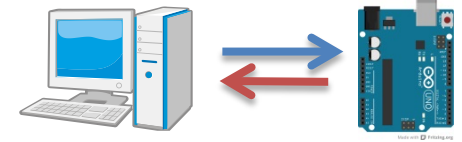
```
int ledRpin = 3;
int ledGpin = 5;
int ledBpin = 6; を定義し， if ~ else if ~ を使う
if (data == '?' ) {
} else if (data == '?' ) {
} else if (data == '?' ) {
} else if (data == '?' ) {
}
```

- pinModeも忘れず設定



Made with  Fritzing.org

2.3 シリアル通信（送受信）



- パソコンからマイコンへ，マイコンからパソコンへ

setup()

通信速度設定

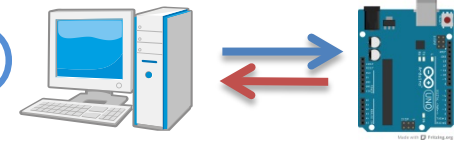
```
Serial.begin(9600);
```

loop()

データの受信確認
データの読み込み
データの送信

```
if (Serial.available() > 0) {  
  byte data = Serial.read();  
  Serial.write(data);  
}
```

E4.3 シリアル通信 (送受信)



```
int ledPin = 3;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  if (Serial.available() > 0) {
    byte data = Serial.read();
    if (data == 'H') {
      digitalWrite(ledPin, HIGH);
      Serial.println("LED: On");
    } else {
      digitalWrite(ledPin, LOW);
      Serial.println("LED: Off");
    }
  }
}
```

データを受信したか？

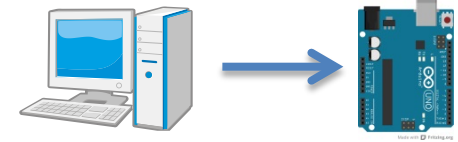
データの取り込み

データの送信

データの送信

3 LEDを調光する（受信）

E4.4 シリアル通信（受信）



- ファイル→スケッチの例→04.Communication→Dimmer

```
const int ledPin = 3;
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int brightness;
  if (Serial.available() > 0) {
    brightness = Serial.parseInt();
    brightness = constrain(brightness, 0, 255);
    analogWrite(ledPin, brightness);
  }
}
```

データを受信したか？

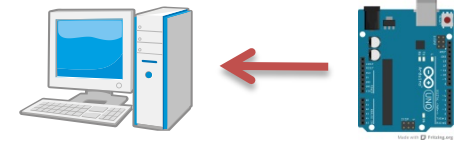
文字を整数に変換

0~255の範囲内に

PWM出力

4 スイッチの状態を調べる（送信）

E4.5 シリアル通信（送信）



```
const int swPin = 11;
void setup() {
  Serial.begin(9600);
}
void loop() {
  byte sw;
  sw = digitalRead(swPin);
  if (sw == LOW) {
    Serial.println("SW: On");
  } else {
    Serial.println("SW: Off");
  }
}
```

スイッチピンの状態読み込み

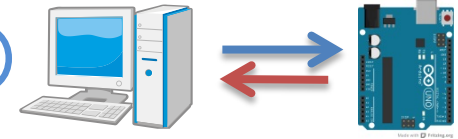
スイッチが押された

データを送信

スイッチが押されていない

5 パソコンからLEDを光らせる（応用）

E4.6 シリアル通信 (送受信)



- ファイル→スケッチの例→04.Communication→ReadASCIIString

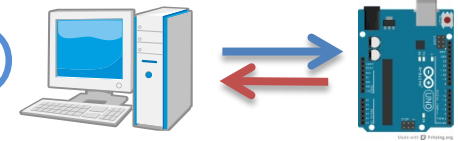
// E4.6

```
const int redPin = 3;  
const int greenPin = 6;  
const int bluePin = 5;
```

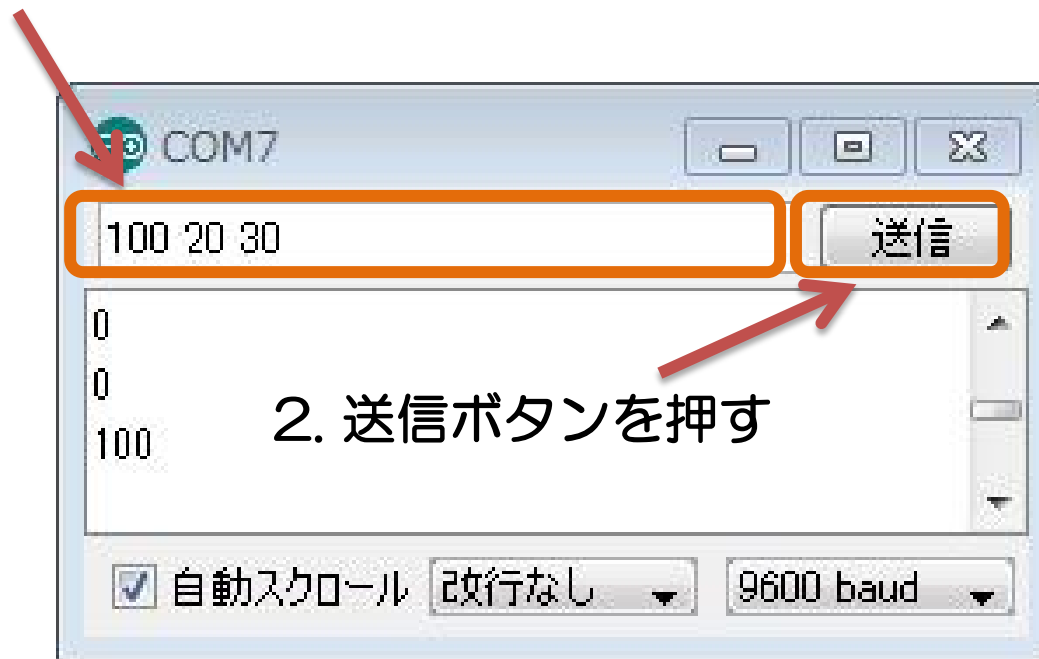
```
void setup() {  
  Serial.begin(9600);  
  pinMode(redPin, OUTPUT);  
  pinMode(greenPin, OUTPUT);  
  pinMode(bluePin, OUTPUT);  
}
```

```
void loop() {  
  while (Serial.available() > 0) {  
    int red = Serial.parseInt();  
    int green = Serial.parseInt();  
    int blue = Serial.parseInt();  
  
    red = constrain(red, 0, 255);  
    green = constrain(green, 0, 255);  
    blue = constrain(blue, 0, 255);  
    analogWrite(redPin, red);  
    analogWrite(greenPin, green);  
    analogWrite(bluePin, blue);  
    Serial.println(red);  
    Serial.println(green);  
    Serial.println(blue);  
  }  
}
```

E4.6 シリアル通信（送受信）



1. 送信したい数値を入力, ” “, “,” でデータを区切る
R, G, B（各 0~255の値）



2. 送信ボタンを押す

シリアルモニタ

付録1： シリアル通信に使用する関数

- Serial.begin(speed)** シリアル通信のデータ転送速度を設定
300/1200/2400/9600/14400/19200
38400/57600/115200から選択
- Serial.print(data, format)** データ（文字列）を送信
data: 送信したい値
format: DEC 10進数, HEX 16進数, BIN 2進数
- Serial.println(data, format)** 改行付きでデータを出力
- Serial.available()** データの受信の有無を確認
- Serial.write(data)** バイナリデータを送信
- Serial.read()** バイナリデータを受信
- Serial.parseInt()** 文字を整数に変換
- Serial.parseFloat()** 文字を浮動小数点に変換

付録2：シリアル通信ソフトウェア



TeraTerm・・・シリアル通信端末（ターミナル）

```
COM28:9600baud - Tera Term VT
File Edit Setup Control Window Help
sensor = 527      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 529      output = 131
sensor = 527      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
sensor = 528      output = 131
```

