

たったの30分で

GR-KURUMIでデジカメを作ろう



2014/4/24

がじえっとるねさす 鈴木

Rev. 1.00

使用部品

■ カメラ LS-Y201

<https://www.sparkfun.com/products/retired/10061>

解像度: 640x480(VGA) / 320x480(QVGA) / 160x120

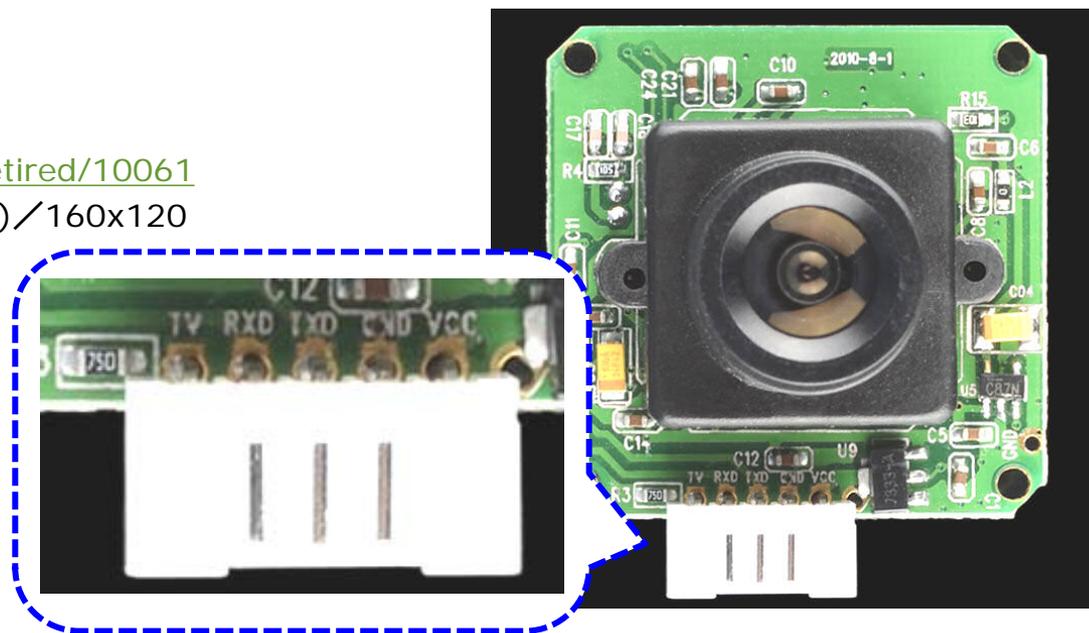
ボーレート: 38400 bps

電源電圧: 3.3V または 5V

サイズ: 32x32 mm、消費電流: 80~100mA

RXD、TXD、GND、VCCのみ使用。

出力はJpegフォーマット



■ GR-KURUMI



■ FTDI (GR-KURUMIとPCをつなぐUSBシリアル変換)

<http://www.switch-science.com/catalog/1032/>

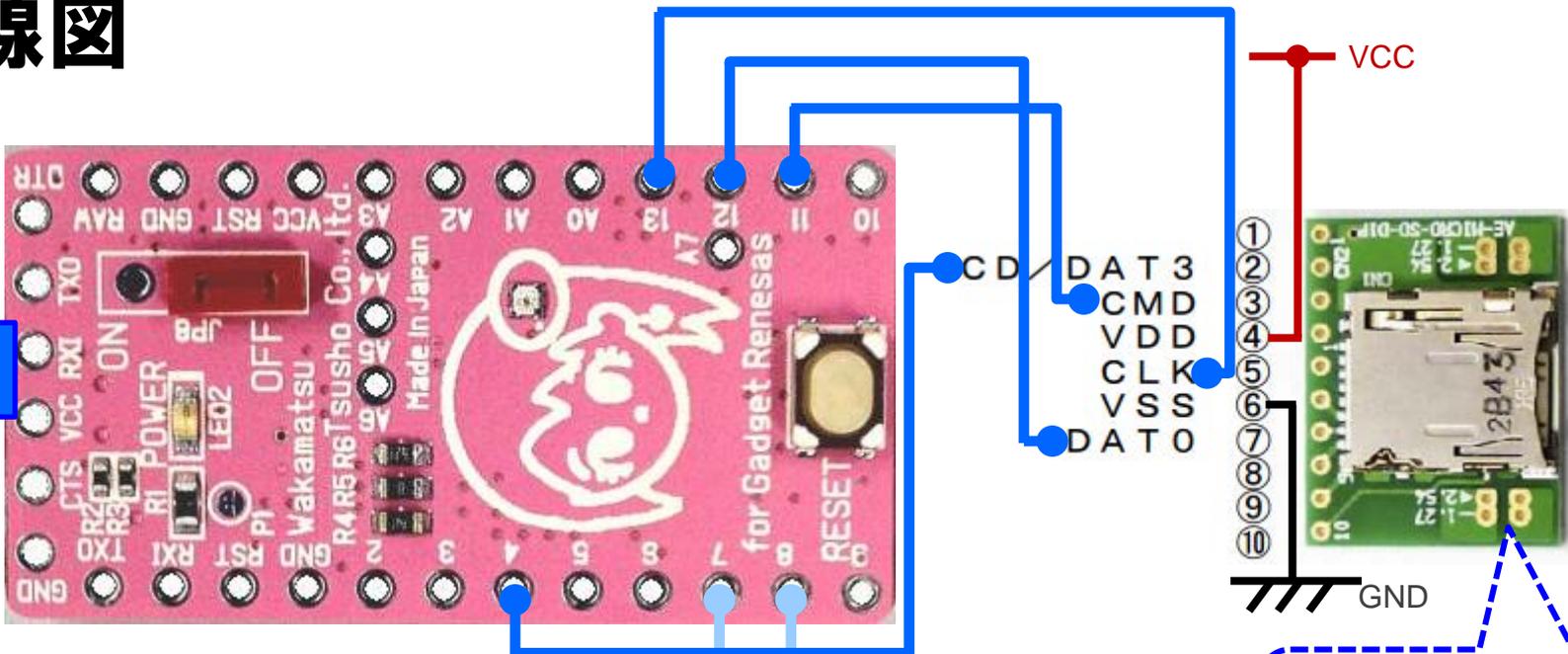


■ マイクロSDカードスロット

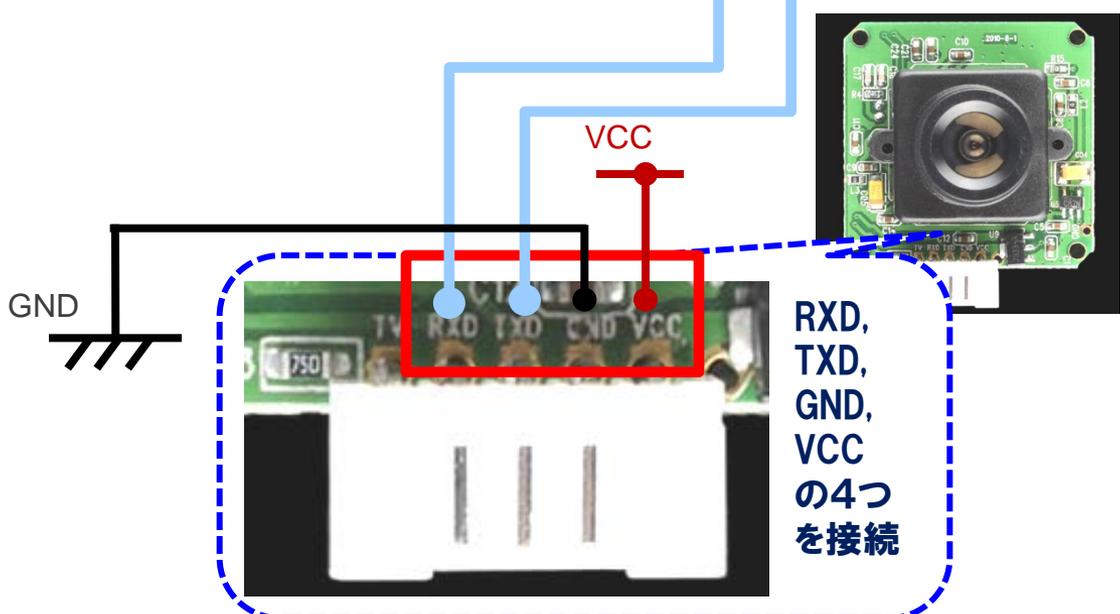
<http://akizukidenshi.com/catalog/g/gK-05488>



配線図



GR-KURUMIは電源が3.3Vなので、そのままSDカードが動作する。



GR-KURUMIの環境を作成

■ IDE for GR E0.4.0を使う

実は便利なArduino互換の環境があります。下記からダウンロードしてください

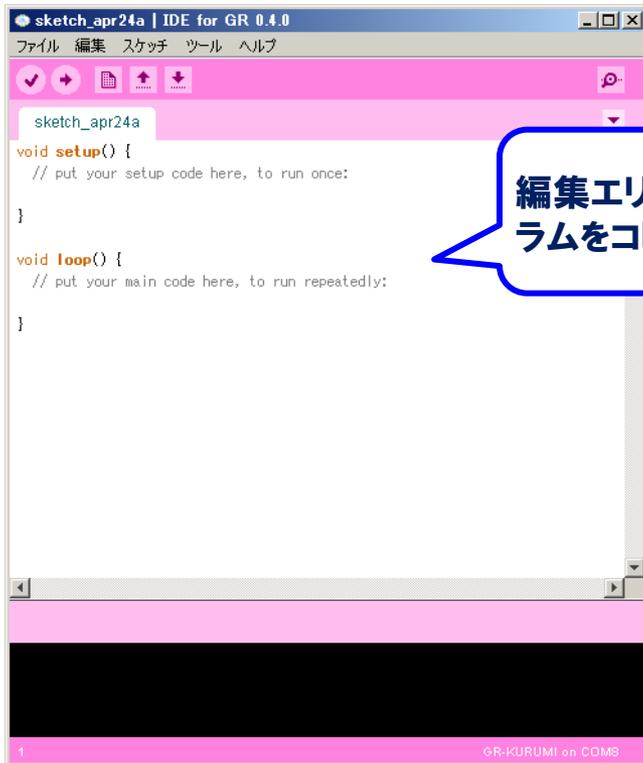
http://japan.renesasrulz.com/gr_user_forum_japanese/m/mediagallery/146.aspx

http://japan.renesasrulz.com/gr_user_forum_japanese/m/mediagallery/156.aspx
カメラライブラリを上記URLよりダウンロードして追加してください。
(JPEGCamera.zip)ファイルを選択して追加します。

ライブラリを使用 ▶ ライブラリをインストール...
Gadget Renesas Boards
AndroidAccessory
EEPROM
Ethernet
JPEGCamera

JPEGCameraが追加されました。

プログラム編集(初期化部分)



```
#include <Arduino.h> // webコンパイラを使う場合はコメントにする
#include <SD.h>
#include <JPEGCamera.h>

#define SD_CSPIN 4 // SDカードのCSを接続したGR-KURUMIの端子
#define CAM_BUFFER 32 // カメラの受信バッファ

// JPEGCameraインスタンスを作成
// コンストラクタにStreamクラスの派生クラスを指定する
JPEGCamera Camera( Serial1 );

void dateTime(uint16_t* date, uint16_t* time);

unsigned char gRecBuf[ CAM_BUFFER ]; // 受信バッファ
unsigned int gRecSize; // 受信データサイズ
uint16_t gPicSize; // 撮影したJPGE画像サイズ
unsigned int gPicAdrs; // 次に読み込むJPEG画像のアドレス
char gFileName[12] = "pic0.jpg"; // 保存する最初のファイル名

void setup()
{
  Serial1.begin( 38400 ); // JPEG cameraとの通信速度
  Serial.begin( 115200 ); // シリアルモニタの通信速度

  if ( Camera.reset() == false )
  {
    Serial.println("reset failed");
  }
  Serial.print( F("Initializing SD card...") );

  if (!SD.begin( SD_CSPIN ))
  {
    Serial.println(F("Card failed, or not present"));
    // 失敗、何もしない
    while(1);
  }
  Serial.println(F("SD ok. "));

  // 日付と時刻を返す関数を登録
  SdFile::dateTimeCallback( &dateTime );

  // 4秒程まってから撮影を開始する
  delay( 4000 );
}
```

プログラム編集(ループ部分)

```
void loop()
{
  int i;

  // PCからの撮影トリガを待つ
  Serial.println( "Hit Any Key" );
  while (!(Serial.available())) { }
  Serial.read();

  Serial.println( "taking picture..." );
  if( Camera.takePicture() == false )
  {
    Serial.println("fail to take picture");
  }
  Serial.println("ok. ");

  Serial.println("taking size...");
  if( Camera.getSize( &gPicSize ) == false )
  {
    Serial.println("getSize failed");
  }
  Serial.print( gPicSize );
  Serial.println();

  i = 0;
  while ( SD.exists( gFleName ) )
  {
    i++;
    sprintf( gFleName, "pic%d.jpg", i );
  }
  File dataFile = SD.open( gFleName, FILE_WRITE );
  Serial.print( "filename is " );
  Serial.println( gFleName );
}
```

```
// 開始アドレスを0に、gPicSize分読み込むまで繰り返す
gPicAdrs = 0;
while( gPicAdrs < gPicSize )
{
  // 32byte読み込む
  gRecSize = Camera.readData( gRecBuf, CAM_BUFFER, gPicAdrs );
  if ( gRecSize )
  {
    //SDに書き込む
    for (int i=0 ; i<CAM_BUFFER; i+=2 )
    {
      dataFile.write( gRecBuf[i] );
      dataFile.write( gRecBuf[i+1] );
      if (( gRecBuf[i] == 0xFF) && (gRecBuf[i+1] == 0xD9))
      {
        // Jpegの終わりを検出
        break;
      }
    }
    //受信したデータサイズ分、アドレスを更新する
    gPicAdrs += gRecSize;
  }
  else
  {
    Serial.println("readData failed");
  }
  delay(1); //これがないとread failする
}

// 撮影停止 (一旦止めないと新しい画像は撮影されない)
Camera.stopPictures();
dataFile.close();
Serial.println("Success to save picture");
}
```

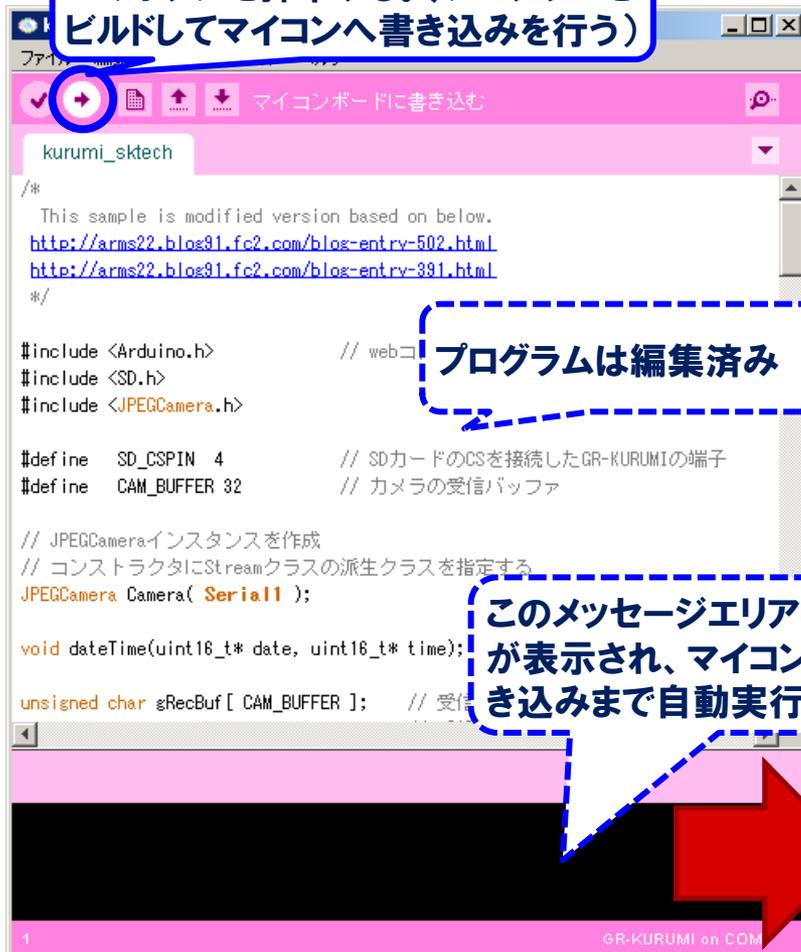
```
void dateTime (uint16_t* date, uint16_t* time)
{
  uint16_t year = 2014; // Set time for saving files
  uint8_t month = 4, day = 24, hour = 18, minute = 0, second = 0;
  *date = FAT_DATE(year, month, day);
  *time = FAT_TIME(hour, minute, second);
}
```

前ページも合わせて編集エリアにコピーする。

GR-KURUMIへ書き込み

2

このボタンを押下する。(プログラムをビルドしてマイコンへ書き込みを行う)



```
kurumi_sktech
/*
  This sample is modified version based on below.
  http://arms22.blog91.fc2.com/blog-entry-502.html
  http://arms22.blog91.fc2.com/blog-entry-391.html
*/

#include <Arduino.h>           // webカメラ
#include <SD.h>
#include <JPEGCamera.h>

#define SD_CSPIN 4             // SDカードのCSを接続したGR-KURUMIの端子
#define CAM_BUFFER 32         // カメラの受信バッファ

// JPEGCameraインスタンスを作成
// コンストラクタにStreamクラスの派生クラスを指定する
JPEGCamera Camera( Serial1 );

void dateTime(uint16_t* date, uint16_t* time);

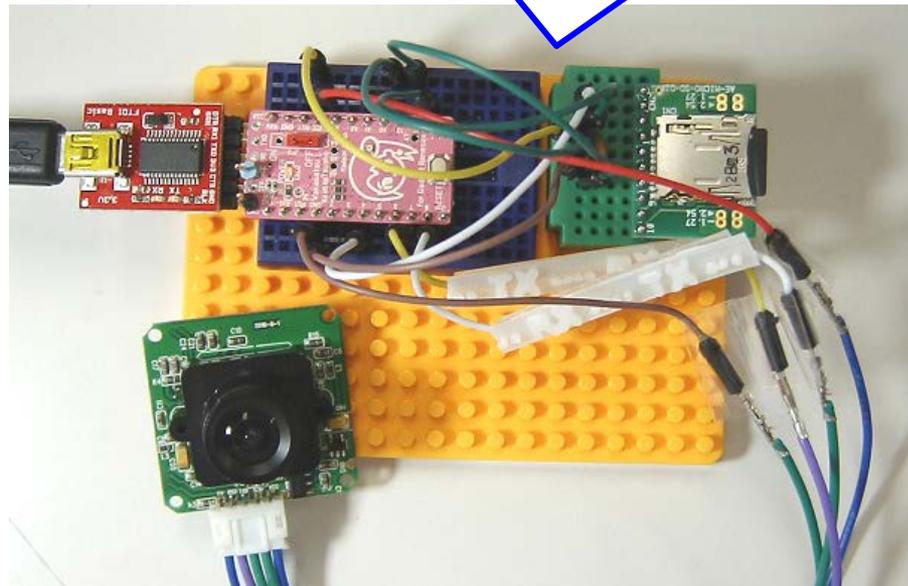
unsigned char gRecBuf[ CAM_BUFFER ]; // 受信バッファ
```

プログラムは編集済み

このメッセージエリアにビルドの過程が表示され、マイコンへプログラム書き込みまで自動実行される

1

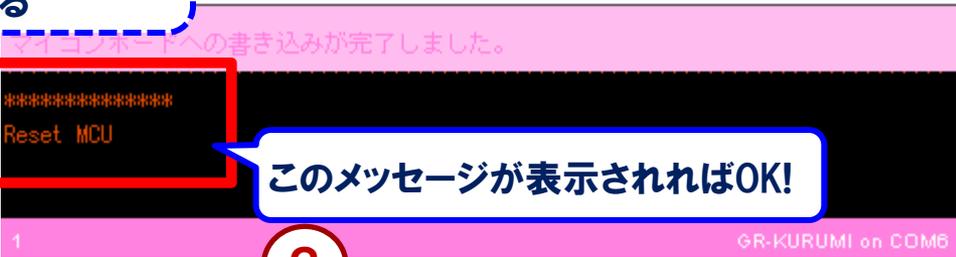
配線も完了させて、PCと接続を行った状態で実行すること



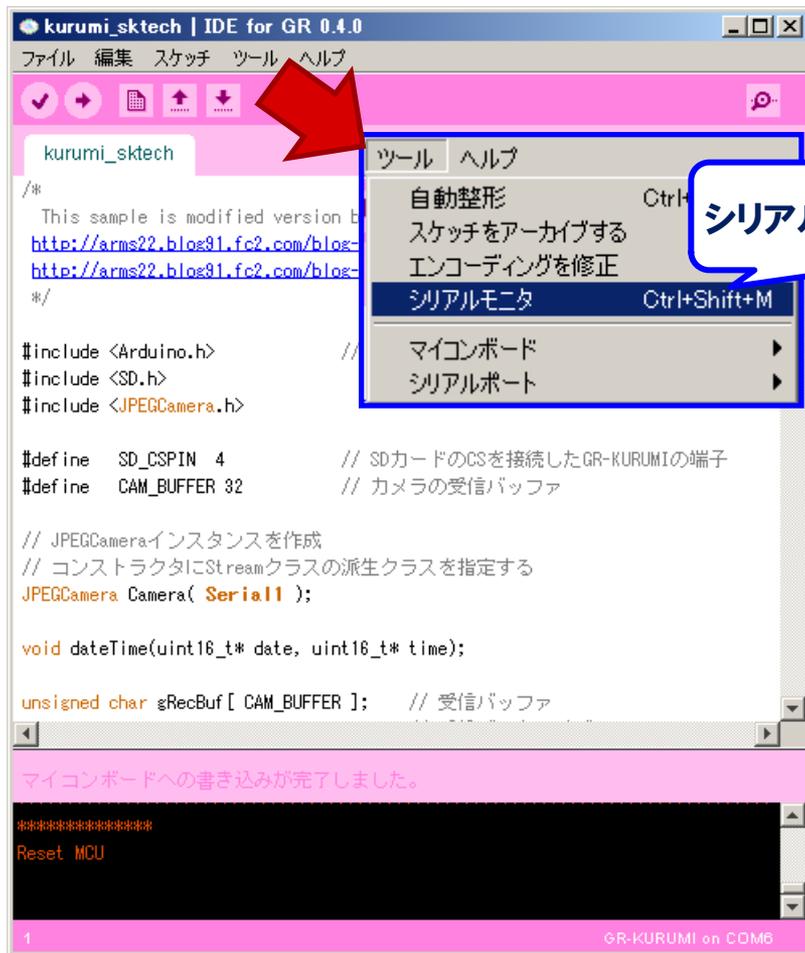
Reset MCU

このメッセージが表示されればOK!

3



動作させてみる



kurumi_sktech | IDE for GR 0.4.0

ファイル 編集 スケッチ ツール ヘルプ

kurumi_sktech

ツール ヘルプ

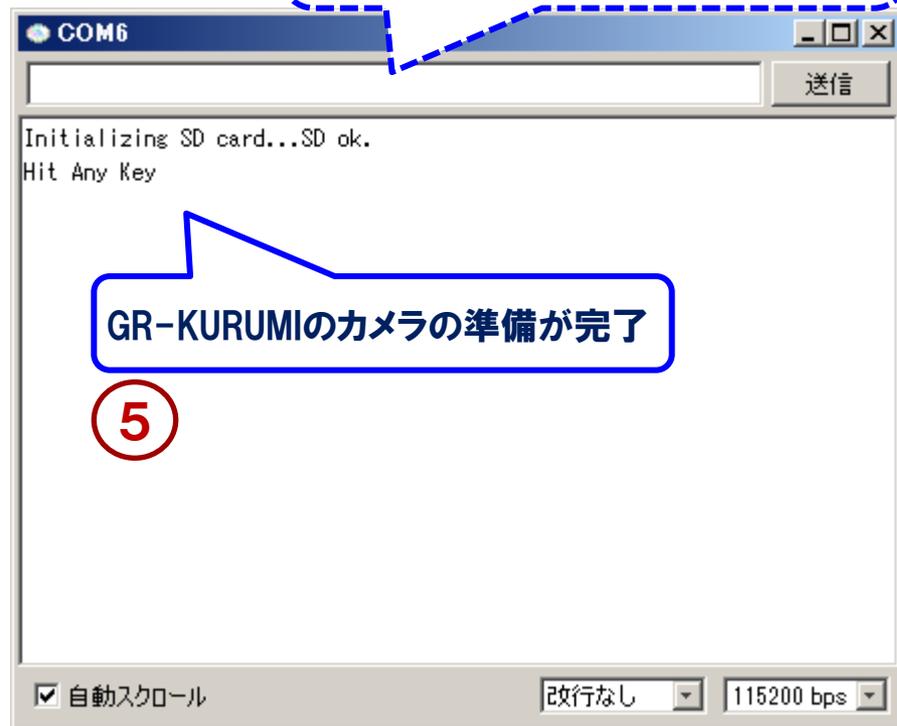
- 自動整形 Ctrl+T
- スケッチをアーカイブする
- エンコーディングを修正
- シリアルモニタ Ctrl+Shift+M
- マイコンボード ▶
- シリアルポート ▶

```
/*  
 * This sample is modified version of the example code from the  
 * http://arms22.blog91.fc2.com/blog-entry-100.html  
 * http://arms22.blog91.fc2.com/blog-entry-101.html  
 */  
  
#include <Arduino.h> //  
#include <SD.h> //  
#include <JPEGCamera.h> //  
  
#define SD_CSPIN 4 // SDカードのCSを接続したGR-KURUMIの端子  
#define CAM_BUFFER 32 // カメラの受信バッファ  
  
// JPEGCameraインスタンスを作成  
// コンストラクタにStreamクラスの派生クラスを指定する  
JPEGCamera Camera( Serial1 );  
  
void dateTime(uint16_t* date, uint16_t* time);  
  
unsigned char gRecBuf[ CAM_BUFFER ]; // 受信バッファ  
  
マイコンボードへの書き込みが完了しました。  
*****  
Reset MCU  
  
1 GR-KURUMI on COM6
```

4

シリアルモニタを選択

GR-KURUMIとシリアル入出力を行うためのウィンドウ。これが表示されればGR-KURUMIが動作している。



COM6

送信

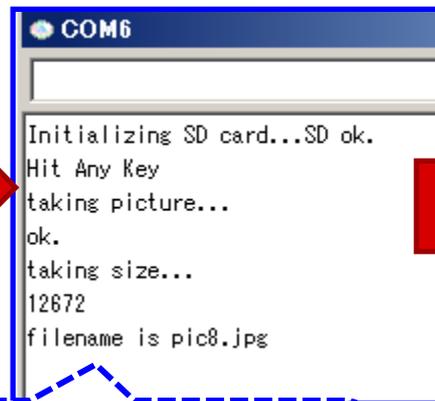
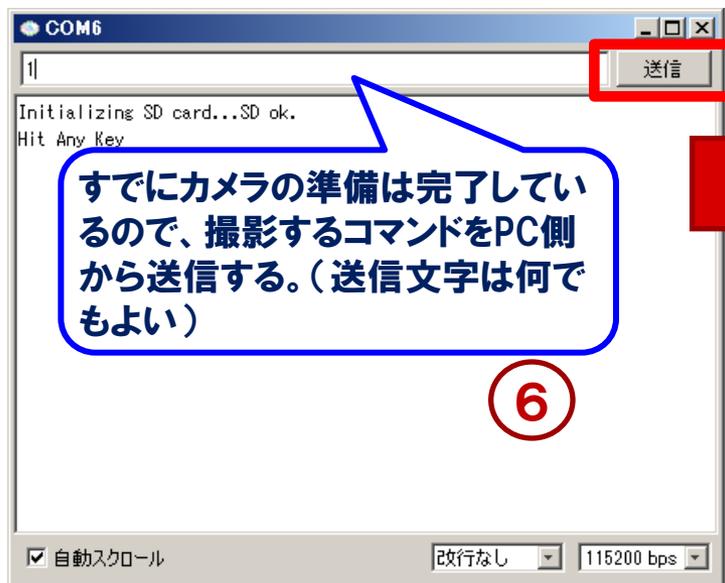
Initializing SD card...SD ok.
Hit Any Key

GR-KURUMIのカメラの準備が完了

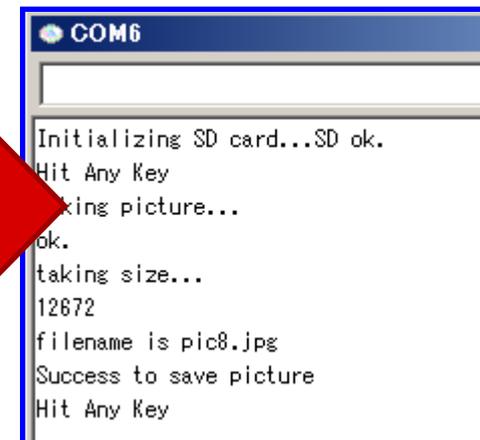
5

自動スクロール 改行なし 115200 bps

動作させてみる

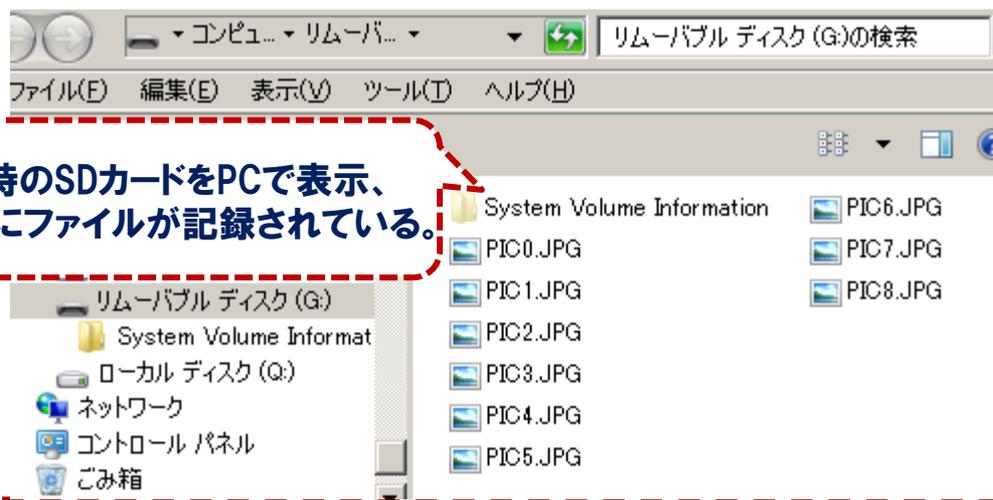


数秒で撮影完了。画像ファイルがSDカードへ記録された。



カメラ準備OK。次の撮影を待っている。

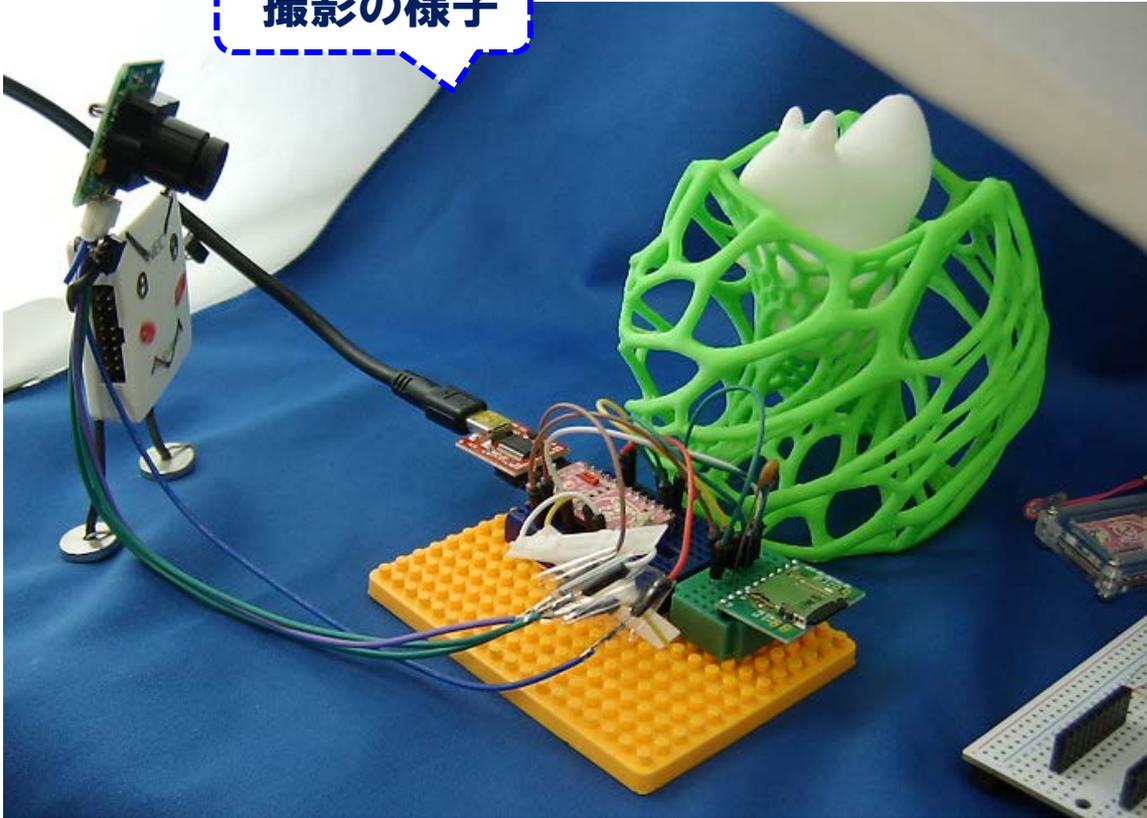
7



あとは⑥、⑦の繰り返し

撮影結果

撮影の様子



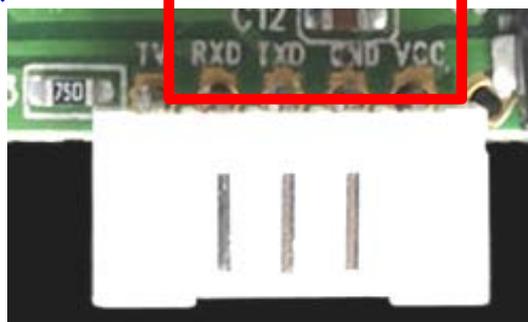
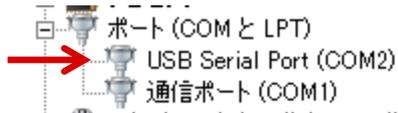
撮影の結果(データ加工なし)



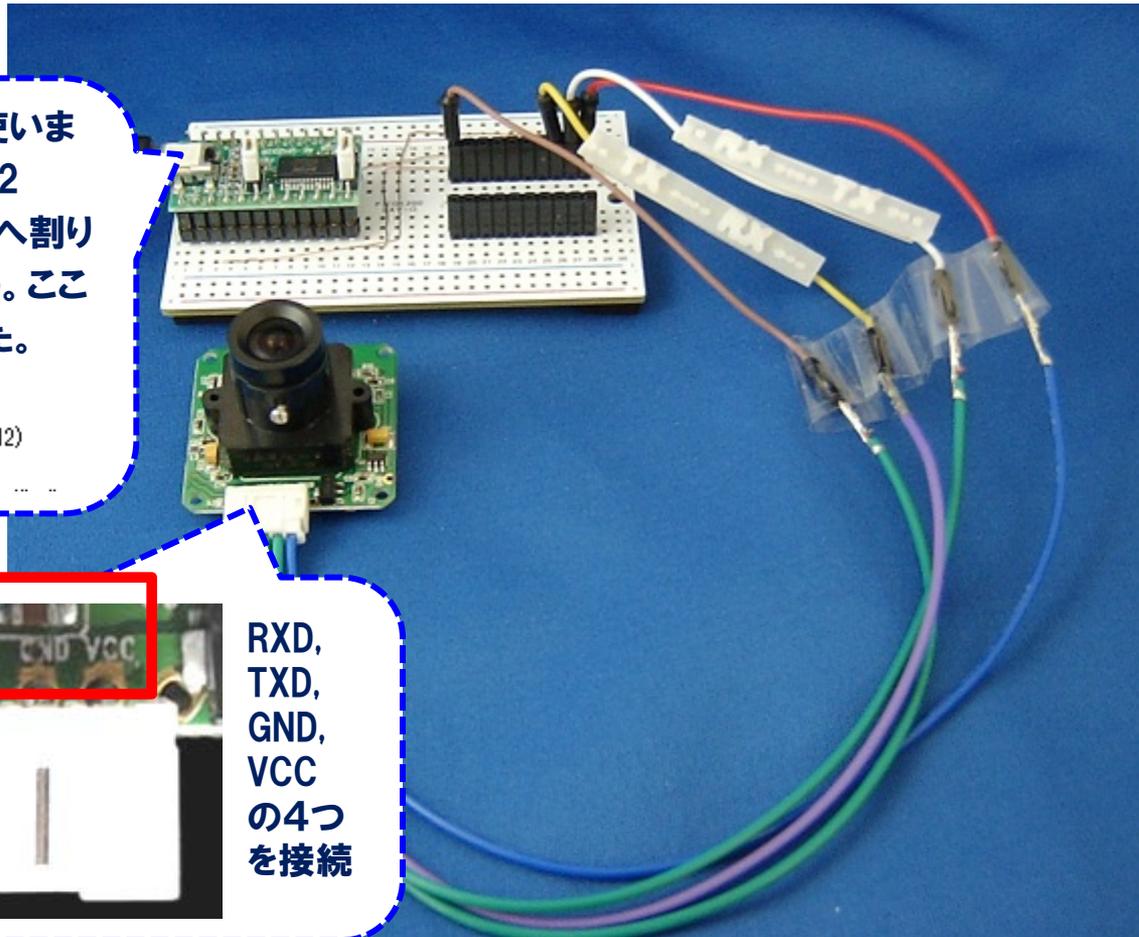
補足資料:カメラのテスト

- PCとカメラを接続してカメラの電源がONになった時の起動メッセージをPC上のターミナルソフトで確認します。

USBシリアル変換ボードを使います。USB Serial PortをCOM2 (COM1~COM4のどれか)へ割り当てるように変更しましょう。ここではCOM2へ割り当てました。

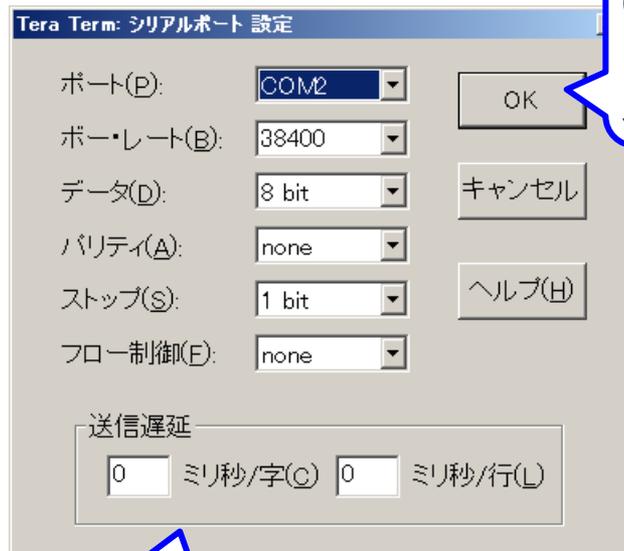


RXD,
TXD,
GND,
VCC
の4つ
を接続



補足資料:カメラのテスト

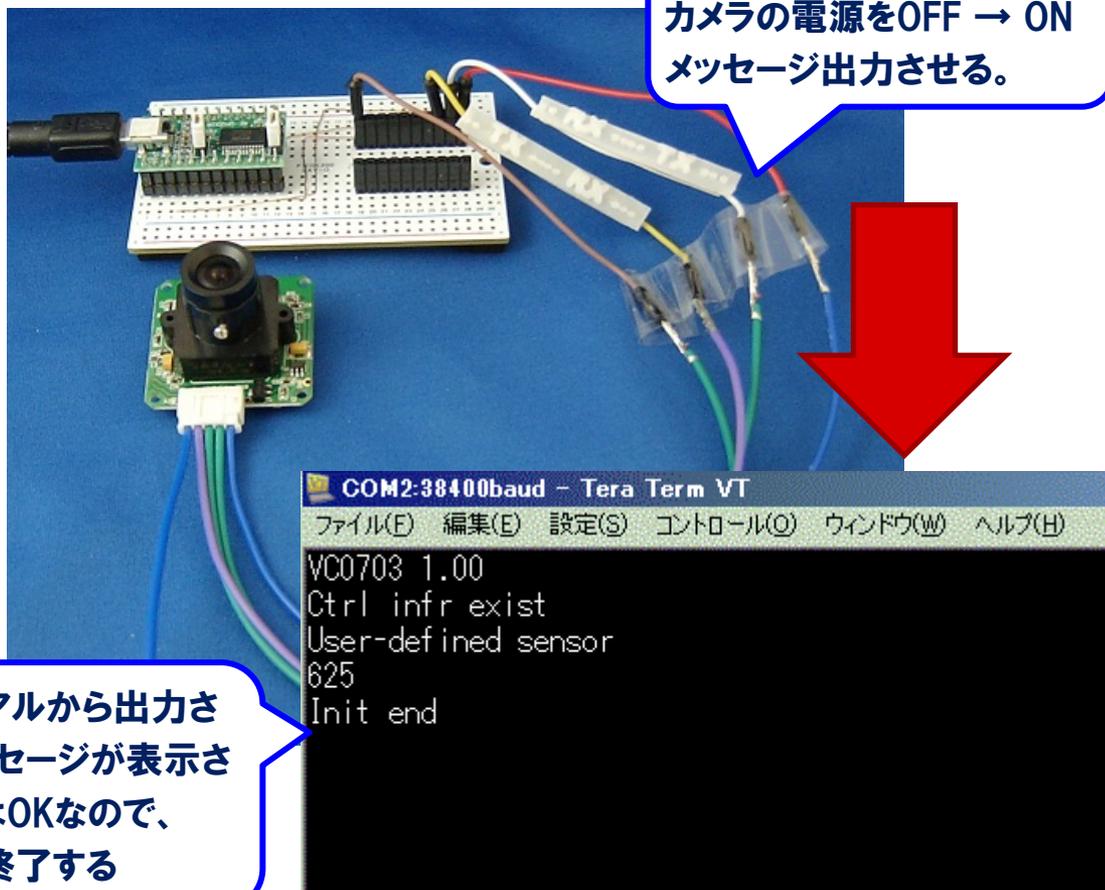
■ PCとカメラの接続確認



1 TeraTermを起動する。

2 COM2に割り当てたUSB Serial Portを38400bps,データ長8bit、パリティなし、ストップビット1、フロー無しに設定する。

3 TeraTermを起動した状態でカメラの電源をOFF → ONメッセージ出力させる。



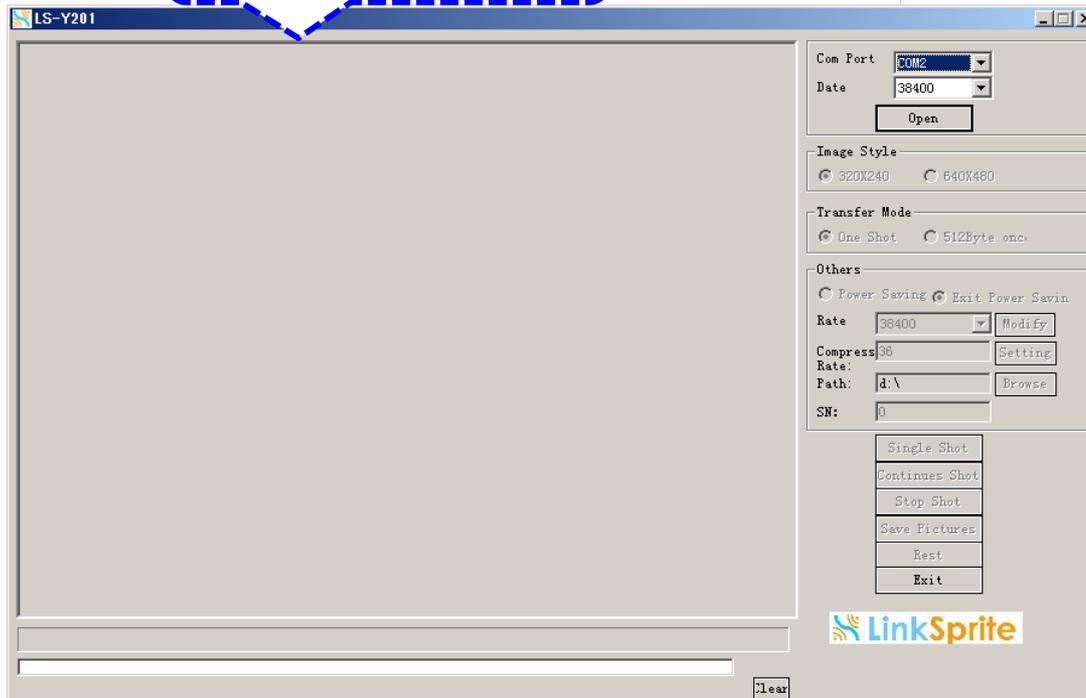
補足資料:カメラのテスト

■ PCでカメラのツールを使う

<https://www.sparkfun.com/products/retired/10061>

LSY201.exeをダウンロードして実行する

カメラツール起動画面



Electronics [US] <https://www.sparkfun.com/products/retired/10061>



Description: The LinkSprite JPEG color camera allows you to capture and output JPEG images through UART, making it easy to integrate into an existing design.

Note: These only work reliably with a 5V power supply, even though in manual states they can work at 3.3V. If you want to use it at 3.3V, need a lower baud rate. Check the comments for more information.

Features:

- VGA/QVGA/160x120 resolution
- Support capture JPEG from serial port
- Default baud rate of serial port is 38400
- 5V power supply
- Size 32X32mm
- Current consumption: 80-100mA

Documents:

- [Manual](#)
- [Software](#)
- [Tutorial](#)
- [Library](#)
- [Product Page](#)
- [mbed Example](#)

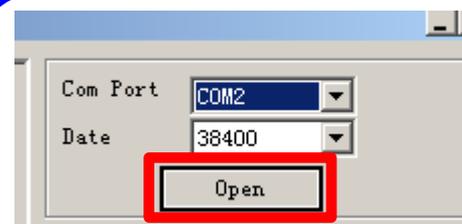
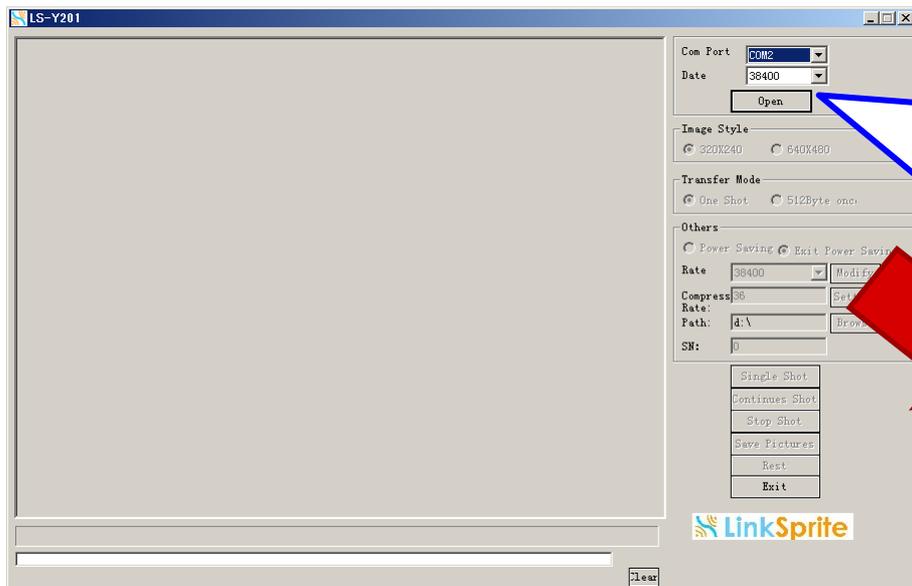
Images are CC BY-NC-SA 3.0

Documents:

- [Manual](#)
- [Software](#)
- [Tutorial](#)
- [Library](#)
- [Product Page](#)
- [mbed Example](#)

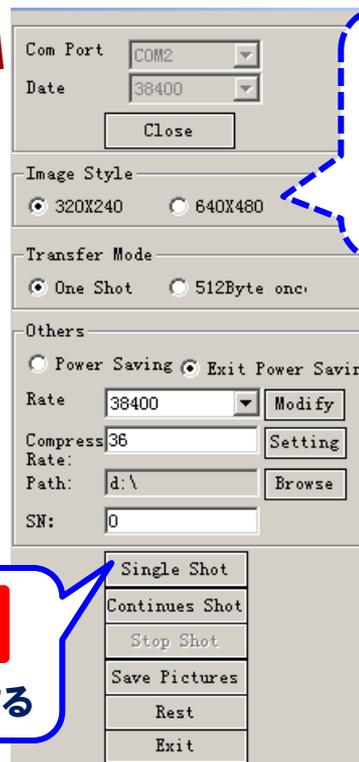
補足資料:カメラのテスト

- ツール上でcom2を選択して開く



5

Com PortがCOM1～COM4しか選択できないが、USBシリアルはCOM2に割り当て済みなのでオープンする。



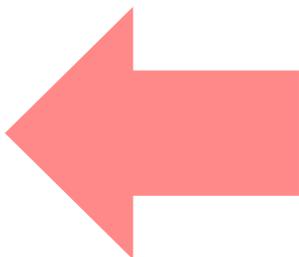
下記のメニューが選択できない(エラーの)場合はカメラの電源をOFF→ONし、カメラを再起動して試してみる。

6

Single Shot

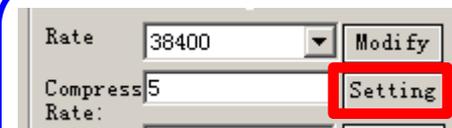
押下して撮影する

次ページ

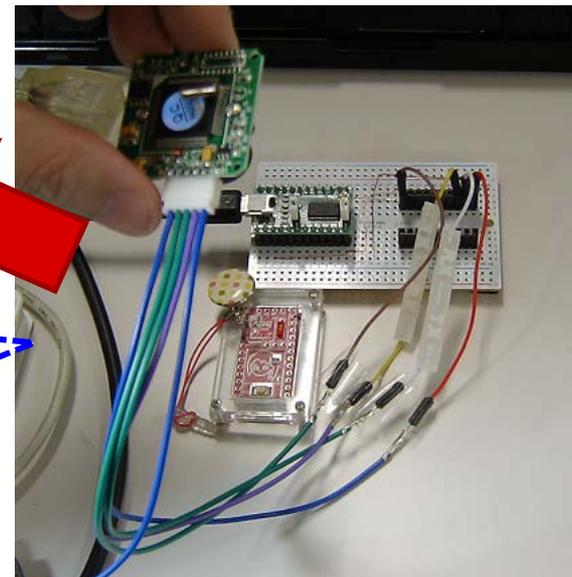


補足資料:カメラのテスト

- PC上にキャプチャされた画像が表示される

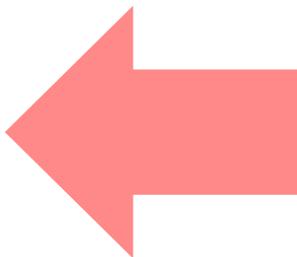


画像の圧縮率も変更可能
小さいほど圧縮率が低くなる



撮影の様子

カメラテスト
完了です。



最後に

- **これをベースとして、インターバル撮影など応用範囲を広げてみてください。
応用作品は、Facebookでも紹介します。**

以上