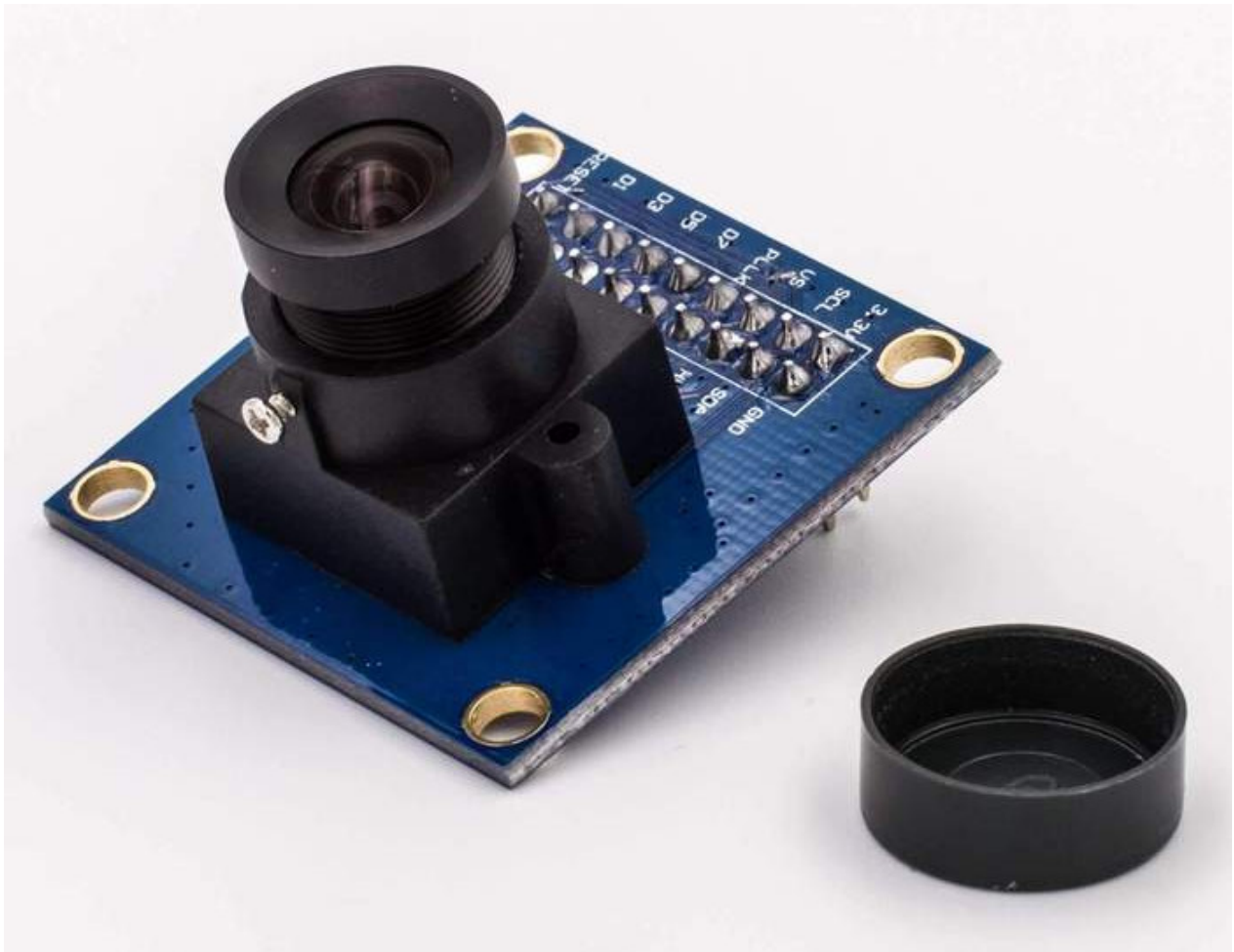




## **Benvenuto!**

Grazie per aver acquistato la nostra fotocamera OV7670 AZ-Delivery per Arduino. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

**Buon divertimento!**



# Az-Delivery

Il modulo fotocamera OV7670 è disponibile in due versioni. Una con memoria *First In First Out (FIFO)* e una senza *FIFO*. Questa descrizione si applica alla versione che vendiamo, cioè il modulo **SENZA FIFO**.

**I dettagli, il codice e le specifiche spiegati in questa descrizione non funzionano con il modulo FIFO.**

La presenza o assenza di un FIFO si riconosce dalle seguenti caratteristiche:

#### **Modulo senza FIFO**

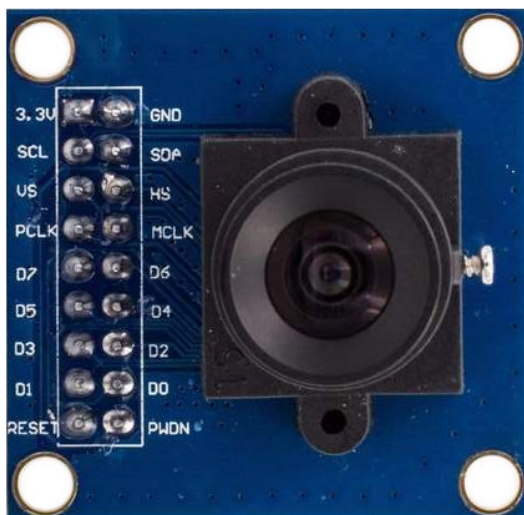
- Connettore a 18 Pin
- Retro senza chip SMD

#### **Modulo con FIFO**

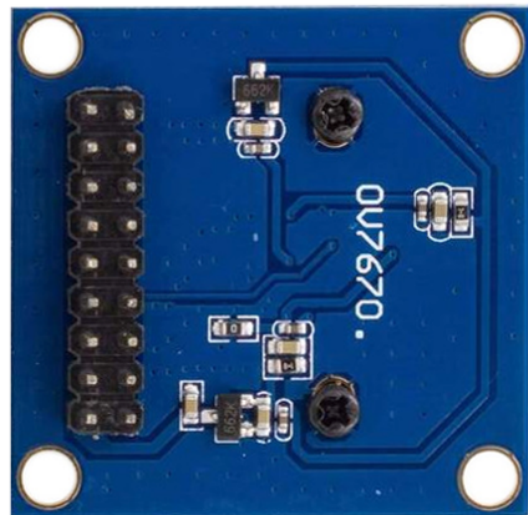
- Connettore a 20 Pin
- Retro con chip SMD (AVRERL)

Il nostro modulo (senza FIFO) ha un aspetto frontale e posteriore come quello della foto qui sotto:

**DAVANTI**



**DIETRO (Senza FiFo)**





**Il modulo fotocamera ha un'alimentazione e un voltaggio logico (dati) di 3.3V.**

**Voltaggi superiori a 3.3V possono distruggere la fotocamera!!!**

Il modulo fotocamera viene fornito con un terminale maschio 9x2, le funzioni dei pin sono indicate qui sotto (per l'assegnazione dei pin, vedi immagine qui sotto):

(Alcuni pin hanno nomi alternativi, a senza della documentazione, che vengono indicati tra parentesi)

<b>PIN</b>	<b>DIREZIONE</b>	<b>FUNZIONE</b>
3.3V		alimentazione: <b>+3.3V</b>
GND		terra: <b>0V</b>
SCL (SIOC)	input	due fili interfaccia seriale clock
SDA (SIOD)	input / output	due fili interfaccia seriale dati
VS (VSYNC)	output	output sincronizzazione verticale
HS (HREF)	output	output sincronizzazione orizzontale
PCLK	output	output pixel clock
MCLK (XCLK)	input	output system clock
D7	output	dati bit 7 MSB
D6	output	dati bit 6
D5	output	dati bit 5
D4	output	dati bit 4
D3	output	dati bit 3
D2	output	dati bit 2

# Az-Delivery

D1	output	dati bit 1
D0	output	dati bit 0 LSB
RESET	input	reset (attivo <b>LOW</b> , GND)
PWDN	input	spegnimento (attivo <b>HIGH</b> , 3.3V)

La fotocamera supporta le seguenti risoluzioni:

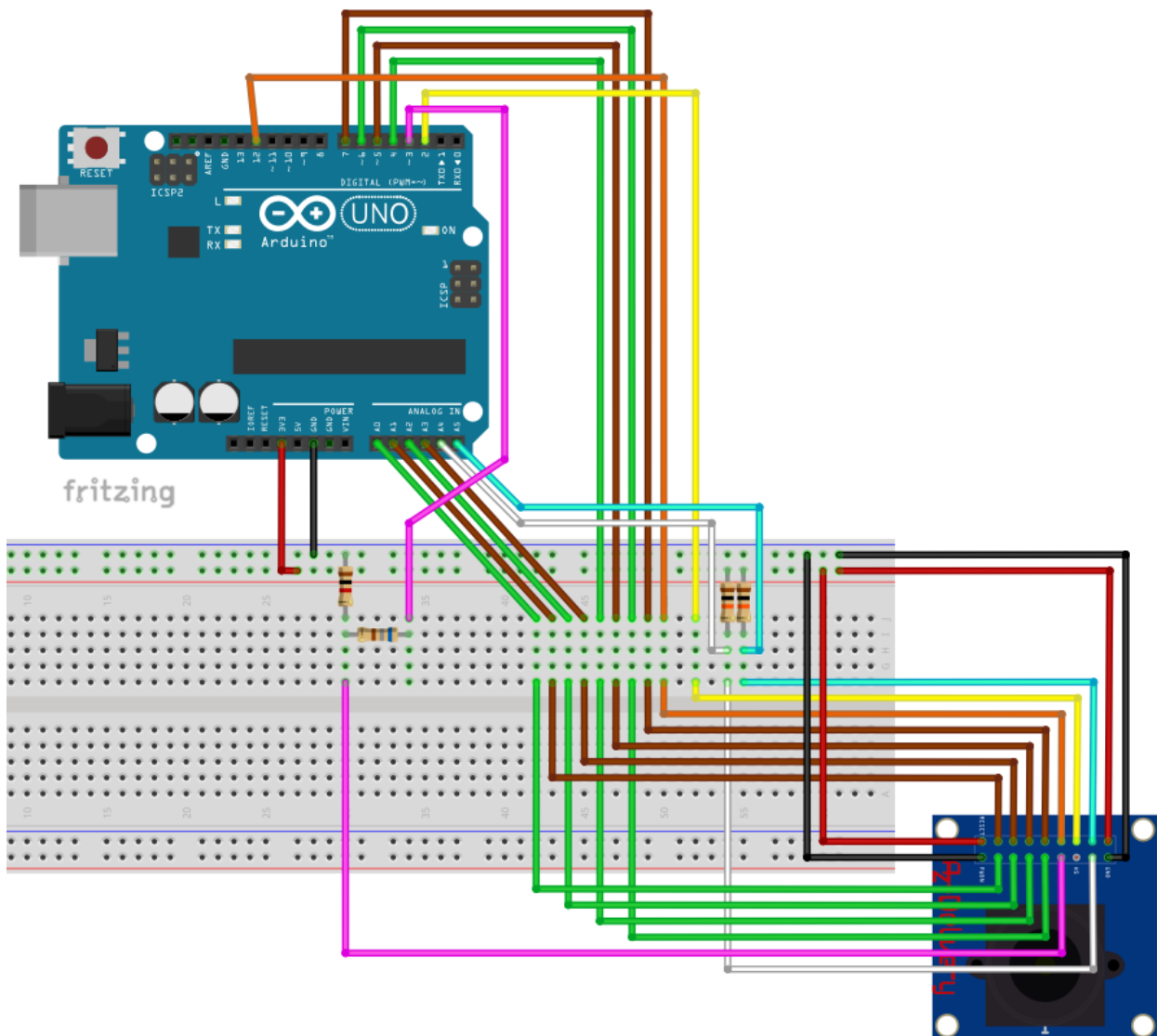
- » VGA (640x480) (modalità standard)
- » QVGA (320x240)
- » CIF (352x240)
- » QCIF (176x144)
- » Risoluzione definibile liberamente fino a massimo VGA

La trasmissione dei dati avviene byte dopo byte in parallelo con uno dei seguenti possibili formati di trasmissione dei dati:

- » YCbCr422 (standard)
- » RGB565
- » RGB555
- » RGB444

# Az-Delivery

## Collegamento della fotocamera con Uno



# Az-Delivery

Pin camera	>	Pin Uno	
3.3	>	3.3V	Filo rosso
GND	>	GND	Filo nero
SIOC	>	A5 (SCL)	Filo ciano
SIOD	>	A4 (SDA)	Filo bianco
VSYN	>	Pin digitale 2	Filo giallo
HREF	>	Non connesso a nulla	
PCLK	>	Pin digitale 12	Filo arancio
XCLK (via partitore tensione)		Pin digitale 3	Filo rosa
Data 7	>	Pin digitale 7	Filo marrone
Data 6	>	Pin digitale 6	Filo verde
Data 5	>	Pin digitale 5	Filo marrone
Data 4	>	Pin digitale 4	Filo verde
Data 3	>	A3	Filo marrone
Data 2	>	A2	Filo verde
Data 1	>	A1	Filo marrone
Data 0	>	A0	Filo verde
RESET	>	3.3V	Filo rosso
PWDN	>	GND	Filo nero

Sia le linee SDA che SCL hanno una **resistenza pull-up di 10kΩ**.

## NON COLLEGARE PIN XCLK DIRETTAMENTE AL PIN DIGITALE UNO!!!

Il pin XCLK è un pin di ingresso della fotocamera e funziona su una logica da 3.3V, usiamo quindi un **partitore di tensione** per abbassare la logica UNO da 5V a 3.3V. Usiamo una **resistenza da 1kΩ** (un'estremità al GND e la seconda all'XCLK), e una **resistenza da 680Ω** (un'estremità al D3 dell'Uno e la seconda al XCLK).

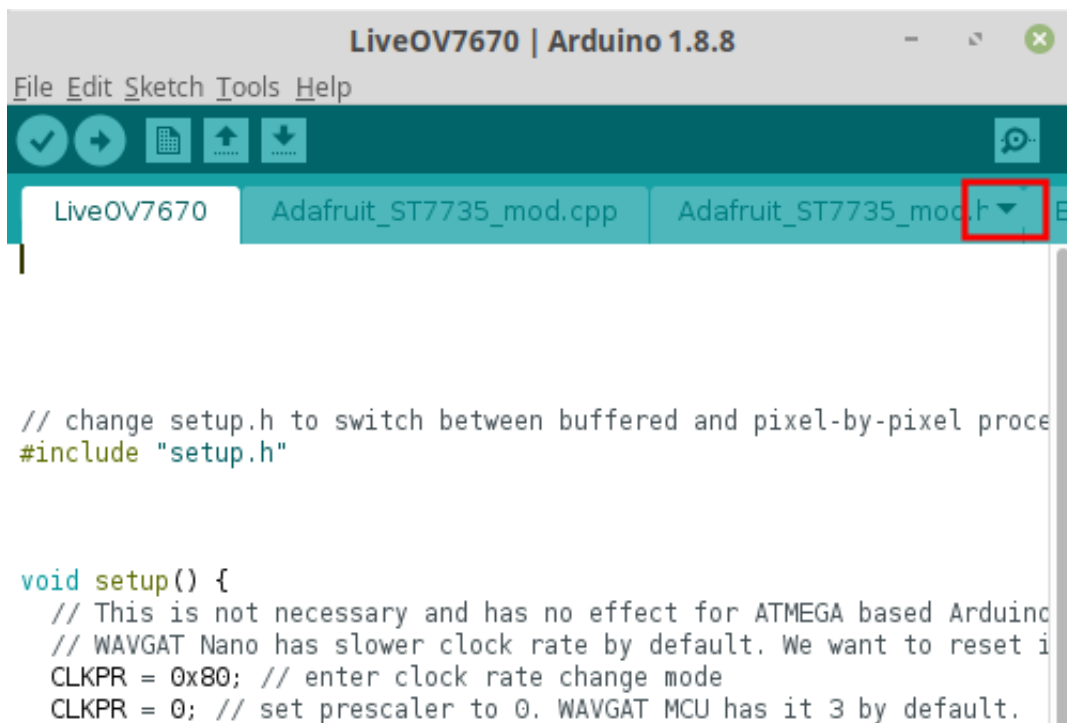
# Az-Delivery

## Sketch

Questo sketch di esempio è fatto da **indrekluuk**, e ti mostreremo come usarlo. Vai su <https://github.com/indrekluuk/LiveOV7670> e scaricalo per la tua macchina locale. Estrailo, ed apri questo sketch:

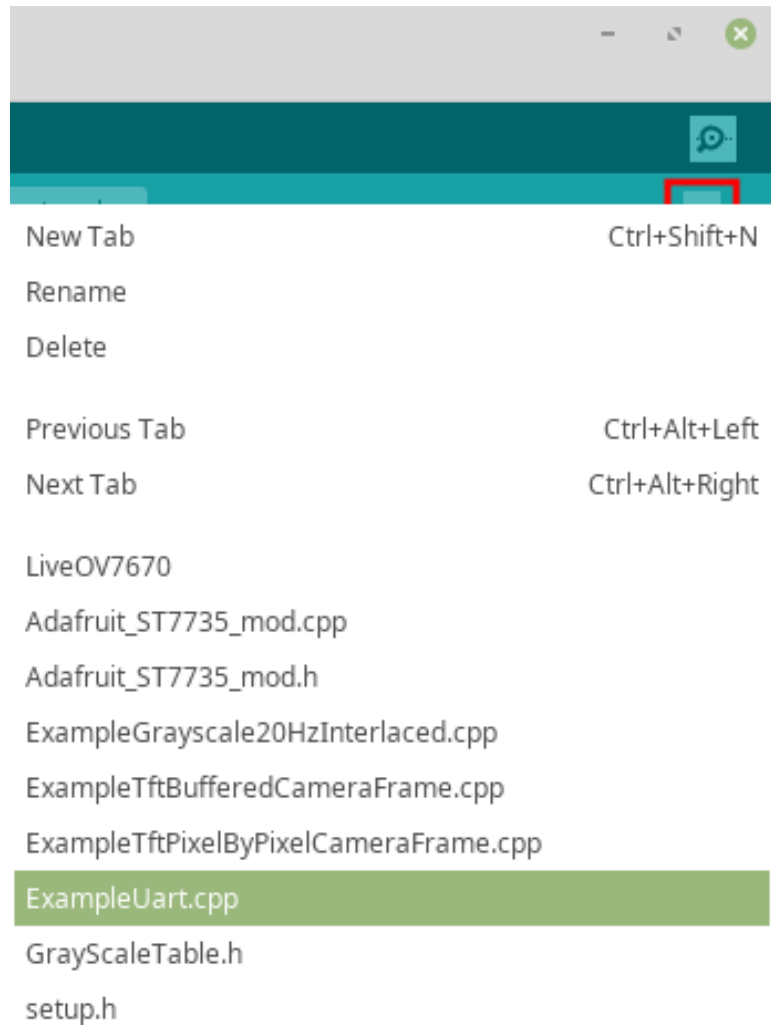
***src/LiveOV7670/LiveOV7670.ino***

nella tua Arduino IDE. Via su *File > Open* e cerca la cartella che è stata estratta, e aprila. Si apriranno diverse schede (sketch) in una finestra Arduino IDE. Naviga su *ExampleUard.cpp* premendo CTRL + TAB o facendo clic sul pulsante di commutazione (nel rettangolo rosso sull'immagine in basso)



# Az-Delivery

Si apriranno dei nuovi menu a discesa, seleziona quindi *ExampleUart.cpp*





# Az-Delivery

Dalla riga 22 alla riga 28 ci sono diversi commenti, che spiegano le modalità operative di questo sketch, seguite da una riga `#define` in cui specifichiamo la modalità operativa. Ai fini del presente manuale utilizzeremo la modalità 3, come mostrato di seguito.

```
// select resolution and communication speed:
// 1 - 320x240 with 2M baud (may be unreliable!)
// 2 - 320x240 with 1M baud
// 3 - 160x120 with 1M baud
// 4 - 160x120 with 115200 baud
// 5 - 320x240 grayscale with 1M baud
// 6 - 160x120 grayscale with 1M baud
#define UART_MODE 3
```

# Az-Delivery

Naviga poi su *Setup.h*. In *Setup.h* dobbiamo configurare l'esempio che useremo. Dalle righe 11 a 30 ci sono dei commenti che spiegano cosa fa ogni esempio, seguiti da una riga `#define` in cui specifichiamo l'esempio. Ai fini del presente manuale utilizzeremo l'esempio 3, come mostrato di seguito.

```
/*
 * EXAMPLE = 1
 * Use LiveOV7670Library class that reads line into buffer and
 * sends data to TFT over SPI during blank lines.
 *
 * EXAMPLE = 2
 * Use LiveOV7670Library class that processes data pixel by pixel
 * sends data to TFT during each pixel reading loop.
 * VGA can not be used with line buffer since there is no
 * time (no blank lines) to do something useful with a buffered line
 *
 * EXAMPLE = 3
 * Reads data from LiveOV7670Library and send it over UART to computer
 * Java application for receiving picture from UART
 * https://github.com/indrekluk/ArduinoImageCapture
 *
 * EXAMPLE = 4
 * Gray scale image @20Hz. Interlaced image.
 *
 */
#define EXAMPLE 3
```

Dopo aver effettuato queste modifiche, collega il tuo Arduino Uno al computer tramite cavo USB e carica questo sketch sulla scheda Arduino.

## Lettura delle immagini sul computer

Per farlo avari bisogno di **IntelliJ Idea**. Per scaricarlo, vai su:

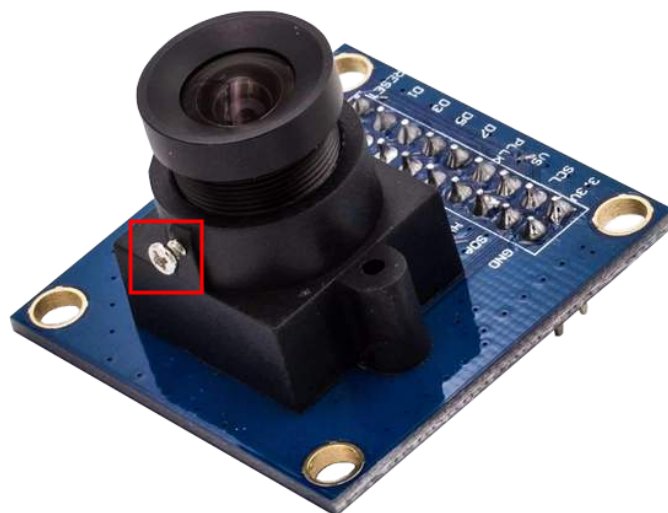
<https://www.jetbrains.com/idea/download>

Dopo l'installazione, è necessario importare il progetto, un'app che leggerà i dati inviati da Arduino Uno tramite cavo USB (linea seriale). Prima dobbiamo scaricarla. Questo progetto è stato creato da **indrekluk** , per scaricarlo quindi vai su:

<https://github.com/indrekluk/ArduinoImageCapture>

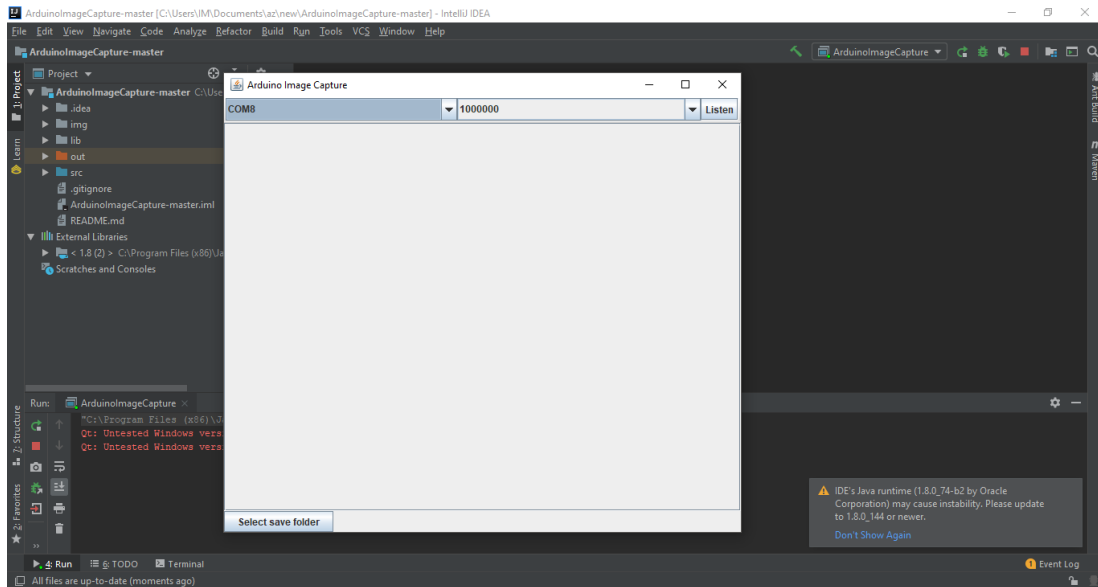
Per eseguire l'importazione, compilare ed eseguire l'app segui le istruzioni sul link sopra.

C'è una vite sulla fotocamera (rettangolo rosso nell'immagine in basso), che viene utilizzata come blocco, regolando la messa a fuoco della fotocamera. Svitandola si cambia la messa a fuoco della fotocamera, e quando impostiamo la messa a fuoco della fotocamera, dobbiamo svitare la vite e bloccare la fotocamera.



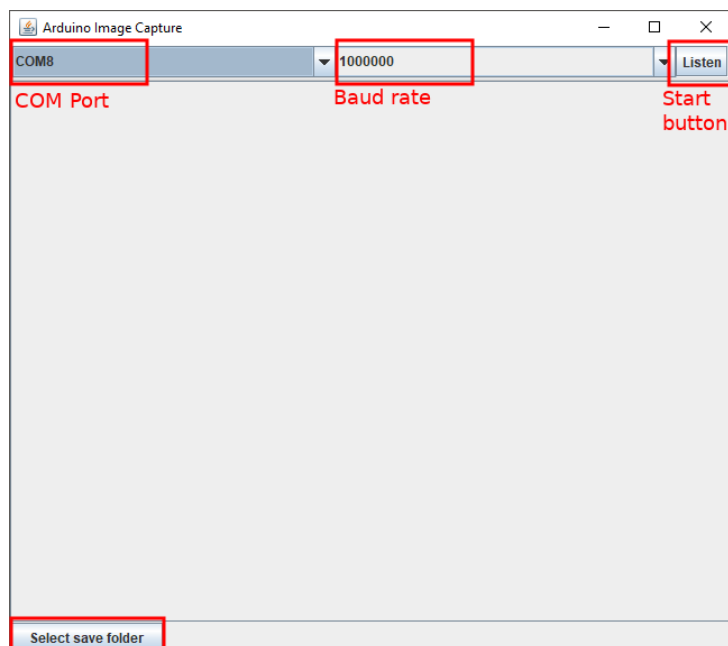
# Az-Delivery

Quando avvii un'app in IntelliJ Idea, questo è ciò che puoi vedere. Devi connettere la tua scheda Arduino Uno al PC prima di avviare questa app.



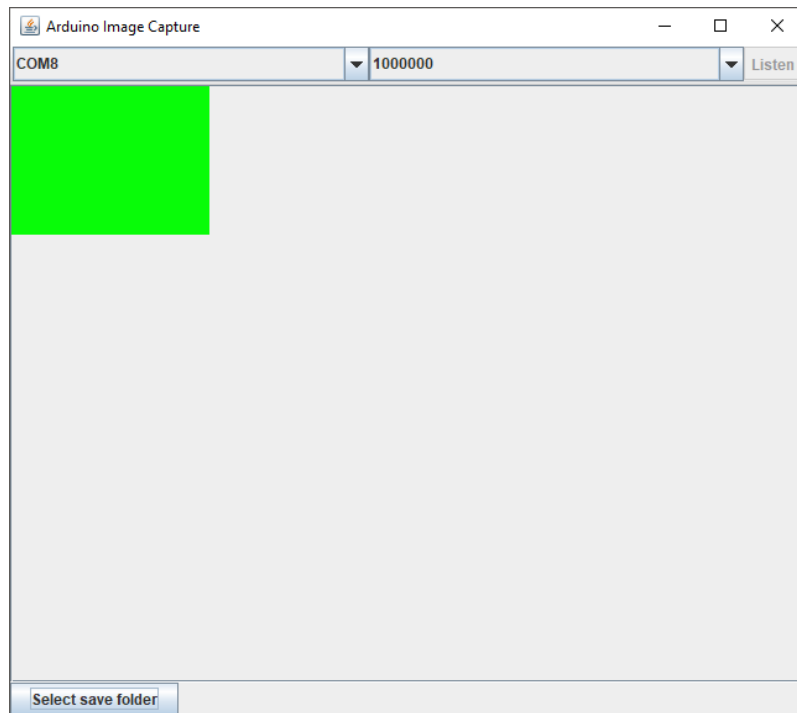
Come puoi vedere, nella finestra Arduino Image Capture, in alto a sinistra, un'app ha rilevato il nostro Arduino Uno sulla porta com: COM8.

Il numero 1000000 accanto a COM8 è un baud rate di 1 MB che stiamo utilizzando. Puoi modificare con una freccia, a destra per altre modalità operative.



# Az-Delivery

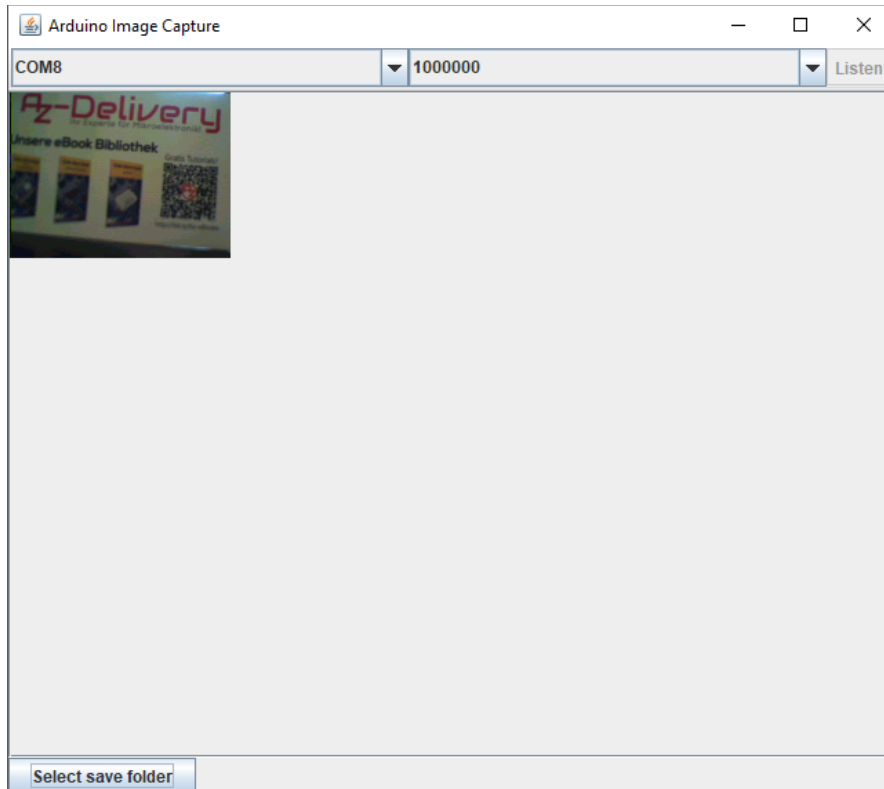
Per iniziare a scattare le foto clicca sul pulsante ***Listen*** (pulsante Start). Se tutto è andato bene, dovrebbe apparire un rettangolo verde.



Ma se appare un rettangolo rosso, l'app e l'Uno hanno **rilevato** la fotocamera, ma l'invio/ricezione di dati **non è avvenuto in modo corretto**. Controlla nuovamente il collegamento tra l'Uno e la fotocamera.

# Az-Delivery

Se tutto è andato bene, dopo il rettangolo nero dovrebbe apparire un'immagine dalla fotocamera.



Nell'immagine della fotocamera c'è il nostro biglietto da visita, puoi vedere la configurazione della fotocamera nell'immagine qui sotto.

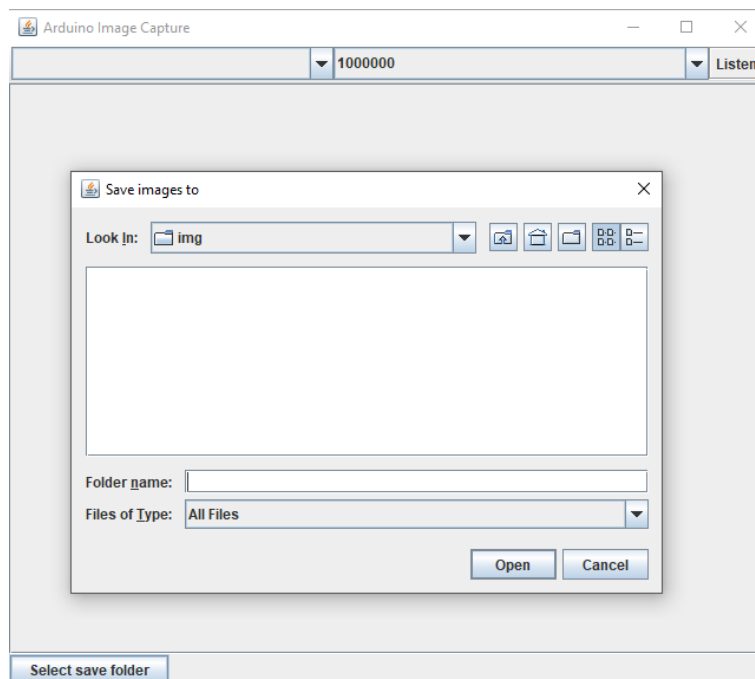


# Az-Delivery

E l'immagine scattata dalla fotocamera è:



C'è un pulsante aggiuntivo nella finestra Arduino Image Capture, in basso a sinistra, il pulsante **Select save folder**. Viene usato per selezionare la cartella in cui verranno salvate le immagini ricevute.



**Ce l'hai fatta, ora puoi usare il tuo modulo per i tuoi progetti.**



E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet.

**Se stai cercando dei prodotti di alta qualità per il tuo Arduino e Raspberry Pi, AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.**

<https://az-delivery.de>

Buon divertimento!

Impressum

<https://az-delivery.de/pages/about-us>