### Willkommen!

Vielen Dank, dass sie sich für unsere Kamera vom Typ "OV7670" für den Arduino von AZ-Delivery entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!



Das Kameramodul OV7670 ist in zwei Versionen erhältlich. Eine mit *First-In-First-Out-Speicher (FIFO)* und eine ohne *FIFO*. Diese Beschreibung gilt für das von uns verkaufte Modul, also das Modul **OHNE FIFO**.

### Die Details, der Code und die Spezifikationen, die in dieser Beschreibung erläutert werden, gelten nicht für das Modul mit FIFO.

Anhand folgender Charakteristika können Sie erkennen, ob ein FIFO-Speicher vorhanden ist:

#### Modul ohne FIFO

- 18-poliger Stecker
- Rückseite ohne SMD chip

#### Modul mit FIFO

- 20- poliger Stecker
- Rückseite mit SMD chip (AVRERL)

Unser Modul (ohne FIFO) sollte wie folgt aussehen:



FRONT

### BACK (without FiFo)



Das Kameramodul hat eine Versorgungs- und Logik-(Daten-)Spannung von 3,3V.

### Spannungen über 3,3V können die Kamera beschädigen!!!

Das Kameramodul wird mit einer 9x2 Stiftleiste geliefert, die Pin-Funktionen werden unten erläutert (siehe Bild oben für die Pin-Belegung):

(Einige Pins haben alternative Namen, je nach Dokumentation, in Klammern)

Pin	Ausrichtung	Funktion
3.3V		Stromversorgung: +3.3V
GND		Masse: <b>0V</b>
SCL (SIOC)	Eingang	Zweidraht-Serienschnittstelle (Takt)
SDA (SIOD)	Eingang / Ausgang	Zweidraht-Serienschnittstelle (Daten)
VS (VSYNC)	Ausgang	vertikale Synchronisation (Ausgang)
HS (HREF)	Ausgang	horizontale Synchronisation (Ausgang)
PCLK	Ausgang	Pixeltakt (Ausgang)
MCLK (XCLK)	Eingang	Systemtakt (Eingang)
D7	Ausgang	Datenbit 7 MSB
D6	Ausgang	Datenbit 6
D5	Ausgang	Datenbit 5
D4	Ausgang	Datenbit 4
D3	Ausgang	Datenbit 3
D2	Ausgang	Datenbit 2
D1	Ausgang	Datenbit 1
D0	Ausgang	Datenbit 0 LSB
RESET	Eingang	reset (aktiv <b>LOW</b> , GND)
PWDN	Eingang	Ausschalten (aktiv HIGH, 3.3V)

Die Kamera unterstützt folgende Auflösungen:

- » VGA (640x480) (Standartmodus)
- » QVGA (320x240)
- » CIF (352x240)
- » QCIF (176x144)
- » Frei wählbare Auflösung bis zu maximaler VGA

Die Datenübertragung erfolgt byteweise parallel in einem der folgenden möglichen Datenübertragungsformate:

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



### Verbindung der Kamera mit dem Uno



Kamera Pin	>	Uno Pin	
3.3	>	3.3V	Roter Draht
GND	>	GND	Schwarzer Draht
SIOC	>	A5 (SCL)	Cyaner Draht
SIOD	>	A4 (SDA)	Weißer Draht
VSYNC	>	Digitaler Pin 2	<b>Gelber Draht</b>
HREF	>	Nicht verbunden	
PCLK	>	Digitaler Pin 12	Oranger Draht
XCLK (Spannung	steiler)	Digitaler Pin 3	Pinker Draht
Data 7	>	Digitaler Pin 7	Brauner Draht
Data 6	>	Digitaler Pin 6	Grüner Draht
Data 5	>	Digitaler Pin 5	Brauner Draht
Data 4	>	Digitaler Pin 4	Grüner Draht
Data 3	>	A3	Brauner Draht
Data 2	>	A2	Grüner Draht
Data 1	>	A1	Brauner Draht
Data 0	>	A0	Grüner Draht
RESET	>	3.3V	Roter Draht
PWDN	>	GND	Schwarzer Draht

Die SDA- und die SCL-Leitung haben einen Pullup-Widerstand von  $10k\Omega$ .

#### Verbinden Sie den XCLK-Pin nicht direkt mit dem UNO-DIGITAL-Pin!!!

Der XCLK-Pin ist der Eingangspin der Kamera und arbeitet mit 3.3V Logik. Daher verwenden wir einen Spannungsteiler, um die 5V Logik der UNO auf 3.3V zu senken. Wir verwenden einen 1k $\Omega$  Widerstand (ein Ende an GND und das zweite Ende an XCLK) und einen 680 $\Omega$  Widerstand (ein Ende an D3 von Uno und das zweite Ende an XCLK).

### Sketch

Dies ist eine Beispielskizze von **indrekluuk.** Wir werden Ihnen zeigen, wie man sie verwendet.

Gehen Sie zu <u>https://github.com/indrekluuk/LiveOV7670</u> und laden Sie es auf Ihren lokalen Rechner herunter. Extrahieren Sie die Datei und öffnen Sie diese Skizze:

*src/LiveOV7670/LiveOV7670.ino* in Ihrem Arduino IDE.

Gehen Sie auf Datei > Öffnen und suchen Sie nach Ihrem extrahierten Ordner und öffnen Sie ihn. Es werden mehrere Tabs (Skizzen) in einem Arduino IDE Fenster geöffnet. Gehen Sie zu *ExampleUard.cpp*, indem Sie STRG + TAB drücken oder auf den Schalter (rotes Rechteck auf dem Bild unten) klicken.



#### Es werden Dropdown-Menüs geöffnet. Wählen Sie ExampleUart.cpp

	-	ø	8
		3	Ð-
New Tab	Ctr	l+Shi	ft+N
Rename			
Delete			
Previous Tab	Ctr	+Alt+	-l eft
i i cvioda i da			Len
Next Tab	Ctrl+	Alt+F	Right
LiveOV7670			
Adafruit_ST7735_mod.cpp			
Adafruit_ST7735_mod.h			
ExampleGrayscale20HzInterlaced.cpp			
ExampleTftBufferedCameraFrame.cpp			
ExampleTftPixelByPixelCameraFrame.cpp			
ExampleUart.cpp			
GrayScaleTable.h			
setup.h			

In den Zeilen 22 bis 28 gibt es mehrere Kommentare, die die Funktionsweise dieser Skizze erläutern. Darauf gefolgt eine Zeile #define, in der wir die Funktionsweise festlegen. Zum Zwecke dieses Handbuchs verwenden wir Modus 3, wie unten dargestellt.

- // 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

Dann gehen Sie zu *Setup.h.* In *Setup.h* müssen wir ein Beispiel setzen, das wir verwenden werden. In den Zeilen 11 bis 30 gibt es Kommentare, die erklären, was jedes Beispiel macht, gefolgt von einer Zeile #define, in der wir das Beispiel angeben. Zum Zwecke dieses Handbuchs werden wir Beispiel 3 verwenden, wie unten gezeigt.

```
/*
 *
   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 buffeRoter line
 *
 *
 *
   EXAMPLE = 3
 *
   Reads data from LiveOV7670Library and send it over UART to computer
 *
   Java application for receiving picture from UART
 *
   https://github.com/indrekluuk/ArduinoImageCapture
 *
 * EXAMPLE = 4
 *
   Gray scale image @20Hz. Interlaced image.
 */
#define EXAMPLE 3
```

Nach diesen Änderungen, verbinden Sie Ihren Arduino Uno via USB-Kabel mit dem Computer und laden sie die Skizze auf Ihr Arduino Board.

### Lesen von Bildern auf Ihrem Computer

Dafür benötigen Sie *IntelliJ Idea*. Um es herunterzuladen gehen Sie auf: <u>https://www.jetbrains.com/idea/download</u>

Nach der Installation müssen wir das Projekt importieren. Eine App, die, die von dem Arduino Uno via USB-Kabel (serielle Schnittstelle) gesendeten Daten, ausliest. Aber zuerst müssen wir es herunterladen. Auch dieses Projekt wurde von **indrekluuk** erstellt, also gehen Sie für den Download zu: <u>https://github.com/indrekluuk/ArduinolmageCapture</u>

Folgen Sie den Anweisungen aus dem obigen Link, um die App zu importieren, zu erstellen und auszuführen.

Es gibt eine Schraube an der Kamera (Rotes Rechteck auf dem unteren Bild), die als Sperre dient, wenn der Fokus der Kamera eingestellt wird. Wenn sie gelöst wird, können wir den Kamerafokus ändern. Wenn wir den Kamerafokus einstellen, dann schrauben wir die Schraube zu und verriegeln die Kamera.



Wenn Sie eine Anwendung in "IntelliJ Idea" starten, können Sie folgendes sehen. Sie müssen Ihr Arduino Uno Board mit dem PC verbinden, bevor Sie diese Anwendung starten.



Wie Sie sehen können, hat im Arduino Image Capture Fenster, oben links, eine App unser Arduino Uno auf dem Com-Port erkannt: COM8.

Die Nummer "1000000" neben COM8 ist die 1MB Baudrate, die wir verwenden. Sie können mit dem Pfeil, rechts, andere Betriebsarten einstellen.

🛃 Arduino Image Cap	ture	- 🗆 X
СОМ8	▼ 1000000	✓ Listen
COM Port	Baud rate	Start
		button
Select save folder		

Sie können, indem Sie auf Listen (Startknopf) drücken, anfangen Bilder zu machen. Dann sollte ein grünes Rechteck angezeigt werden.

🕌 Arduino Image Capture		-	
COM8	▼ 1000000		▼ Listen
Select save folder			

Sollte ein rotes Rechteck angezeigt werden, heißt das, dass die App zwar eine Kamera **erkannt** hat, aber der Datenaustausch **fehlerhaft** war. Überprüfen Sie erneut die Verbindungen zwischen Uno und Kamera.



Bei korrekter Ausführung sollte nach einem grünen Rechteck ein Bild der Kamera angezeigt werden.



Auf dem Kamerabild befindet sich unsere Visitenkarte. Auf dem unteren Bild sehen Sie die Kameraeinstellung.



Das ist das Bild der Kamera:



Es gibt noch eine weitere Schaltfläche im Arduino Image Capture Fenster. Links unten ist die Schaltfläche **Select save folder**. Sie wird verwendet, um einen Ordner zu wählen, in dem die empfangenen Bilder gespeichert werden sollen.

🛃 Arduino Image Capture 🛛 —		×
▼ 1000000	-	Listen
Susimumt.	1	
Folder name:		
Files of <u>Type</u> : All Files ▼		
Open Cancel		
	]	
Select save folder		

### Sie haben es geschafft. Sie können jetzt unser Modul nun für Ihre Projekte nutzen.

Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

Falls Sie nach noch weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.

https://az-delivery.de

Have Fun! Impressum https://az-delivery.de/pages/about-us