

# Kamera-Shield auf Arduino-Basis für ein SSTV-Sendemodul

RICHARD PRINZ – OE1RIB

Als Weiterentwicklung des bereits in [1] und [2] vorgestellten autonomen SSTV-Systems stellt dieser Beitrag die nächste Generation eines Kamera-moduls vor. Es basiert auf einem Mikrocontroller-Board des Arduino-Projekts und wurde als eine „Shield“ genannte Erweiterung realisiert.

Die Arduino-Boards wurden in den vergangenen Jahren aufgrund der vielen Variationsmöglichkeiten und Ausstattungen immer beliebter, wie u. a. Anwendungen in [3] und [4] zeigen. Ebenso wie seinerzeit der PC die Computerindustrie durch Modularisierung und Standardisierung der Erweiterungsmöglichkeiten in Form z. B.

an Arduino-Boards und deren Shields – es ist allerdings weit weniger flexibel als diese. Das *Arduino Camera Shield* stellt, wie auch das Original-Kameramodul vom Typ C3188A, eine Echtzeituhr (engl.: *Real Time Clock*, RTC) in Form des DS1307 mit passender Stützbatterie CR2032 und einem Temperatursensor TC74 zur Verfügung.



**Bild 1:**  
Ansicht des fertigen  
Arduino Camera  
Shields neben einem  
Arduino-Board

**Bild 3:**  
Das selbst gebaute  
Shield lässt sich auf  
beliebige Arduino-  
Boards, hier auf ein  
Arduino Duemilano-  
aufstecken.  
Fotos, Screenshot:  
OE1RIB

die dazugehörigen Widerstände und Kondensatoren, die der Aufbereitung des SSTV-Signals dienen, vom Original entfernt. Ein eigenes Audio Shield kann diese Funktionen übernehmen.

In den Speicherbaustein AL422B lässt sich ein 320 × 256 Pixel großes Bild ablegen, das später über die USB- oder V24-Schnittstelle bzw. ein Ethernet-Shield übertragen werden kann. Auf diesem gespeicherten Bild sind Texte, die mit dem MAX7456 in das Videosignal eingeblendet werden sollten, allerdings noch nicht zu sehen. Es handelt sich hier um ein Bild des CCD-Sensors der Kamera. Sichtbar sind diese Einblendungen erst im analogen S/W-Videosignal der Kamera, das über Jumper durch den MAX7456 geleitet werden kann. Das C3188A-Kameramodul verfügt leider nur über einen S/W-Videoaus-



von ISA- und PCI-Steckkarten revolutionierte, gelang dies im Mikrocontrollerbereich durch die Entwicklungen des Arduino-Projekts [5].

Jedes Basis-Board basiert auf einem Mikroprozessor von Atmel, wobei unterschiedliche Boards jeweils teilweise andere technische Daten besitzen. Die Basis-Boards stellen stets eine standardisierte Schnittstelle für Erweiterungsbaugruppen bereit. Diese als *Shields* bezeichneten Baugruppen ermöglichen es, ein und dieselbe Erweiterung mit unterschiedlichen Boards zu verwenden. Unter [6] finden Sie eine Liste der Shields sowie die entsprechenden Anwendungen.

Mittlerweile kann das *Shield Interface* auch auf Boards verwendet werden, die nicht auf Arduino-Basis arbeiten. Beispiele sind die ARM-Controller [7]. Sofern die Firmware angepasst wurde, sind die Shields auch plattformübergreifend einsetzbar.

## ■ Hardware

Das in [1] und [2] vorgestellte Original-SSTV-System verwendete eine eigene PIC-basierte Controllerplatine und eine Huckepack-Platine mit der eigentlichen Kamera. Das Konzept erinnert ein wenig

Neu hinzugekommen ist ein MAX7456 für Video-On-Screen-Anwendungen, der es ermöglicht, Texte in das Videosignal der C3188A-Kamera einzublenden. Möglich sind dabei z. B. die Uhrzeit oder die Temperatur.

Um dafür Platz zu schaffen, habe ich Bauteile, wie etwa das Filter MAX7419 und

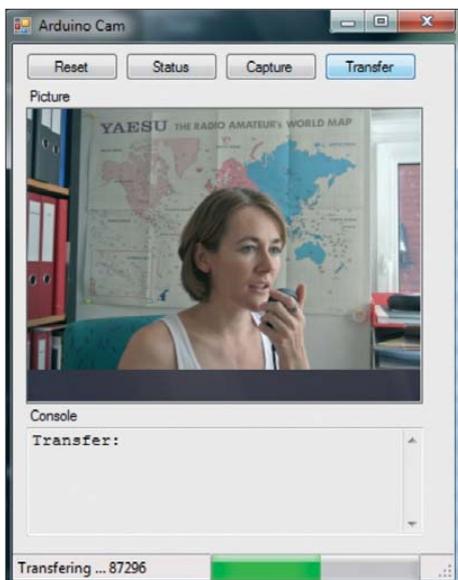
gang, am digitalen Bus steht hingegen ein Farbbild zur Verfügung.

Alle Komponenten, bis auf den MAX7456, werden über den I<sup>2</sup>C-Bus angesteuert. Der MAX7456 verwendet einen SPI-Bus und ist auf dem Arduino Camera Shield analog zum MAX7456-Breakout-Board von [8] realisiert. Deshalb sind auch die im Internet verfügbaren Software-Bibliotheken für den MAX7456 verwendbar.

Alle Bauteile des Arduino Camera Shields sind in SMD-Bauform. Bis auf den MAX7456, die beiden Gatter vom Typ SN74AHCT1G00 und den Temperatursensor TC74 sollte es allerdings keine großen Probleme beim Löten geben. Bei den genannten Bauteilen, vor allem beim MAX7456, sind jedoch dünnes Lötzinn, eine blendfreie Beleuchtung sowie ein mit ruhiger Hand geführter LötKolben mit sehr feiner Spitze von Vorteil. Wer nicht alle Funktionen benötigt, kann auf den TC74, den DS1307 oder den MAX7456 und die entsprechenden Bauteile außen herum verzichten. Dadurch vereinfacht sich der Aufbau.

## ■ Firmware

Die Firmware ist sehr schlicht gehalten und dient zum Testen der Funktionen der zu



**Bild 2:** PC-Anwendung *Arduino Cam* zur Steuerung des *Arduino Camera Shields*

diesem Shield passenden Software-Bibliothek. Analog zum Konzept der standardisierten Hardware-Erweiterung via Shield wird im Arduino-Projekt [5] zum Shield eine passende Software-Bibliothek bereitgestellt. Sie stellt die Funktionen zum Betrieb des Shields zur Verfügung, ohne dass sich der Entwickler mit Details wie etwa Kameraregistern herumschlagen muss. Lediglich ein einfacher Aufruf der Bibliotheksfunktion *Capture* mit den notwendigen Parametern ist nötig und schon ist das Bild im Kasten, sprich im Speicher AL422B.

Der Funktionsumfang der Firmware orientiert sich am Atmel168 des *Arduino Duemilanove Boards* [9]. Da dieser Controller im Vergleich zum originalen SSTV-System

weniger Ressourcen zur Verfügung stellt, habe ich auch hier einige Funktionen eingespart. Auf dem Duemilanove-Board entfallen alle SSTV-Funktionen. Sie lassen sich auf einem besser ausgestatteten Board mit mehr Speicher problemlos implementieren.

Das Kameramodul, auf dem sich ein OV7620 befindet, wird standardmäßig im Modus für 640 × 480 Pixel betrieben und nur beim Speichern eines Bildes in den Modus 320 × 256 Pixel umgeschaltet. Dies ist notwendig, da ein Bild mit 640 × 480 Pixeln nicht in den Speicher passen würde.

Damit bei der Aufnahme nichts schief geht, ist das Modul im Automatikmodus (Weißabgleich usw.) zu betreiben. Liegt das Bild im Speicher, kann es entweder in roher

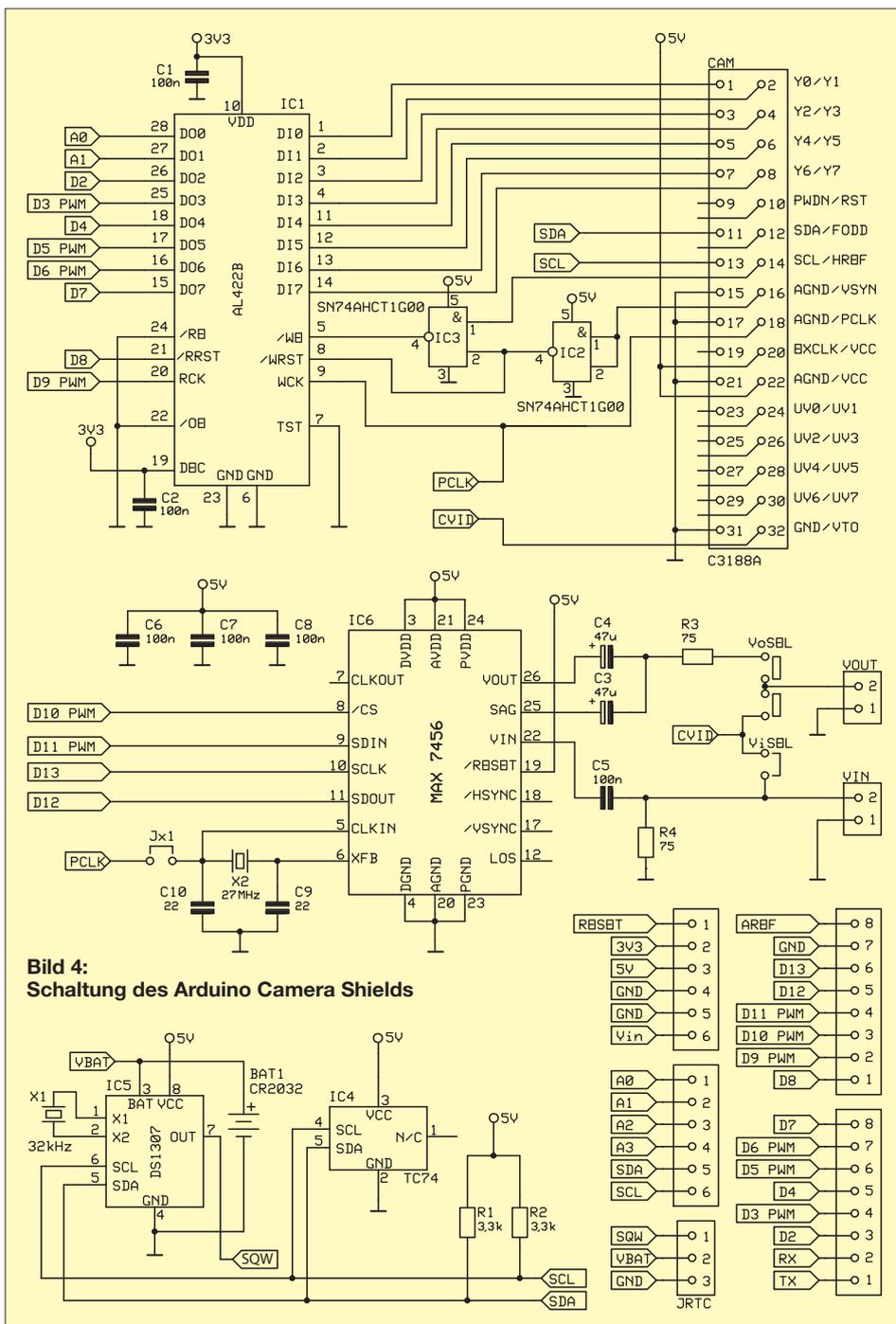
Form oder bereits als Windows-Bitmap sowohl in Farbe als auch in Schwarz-Weiß auf den PC übertragen werden.

Die Software stellt darüber hinaus einfache Funktionen zum Abfragen der Temperatur, der Uhrzeit und des Kamerastatus sowie zum Setzen der Uhrzeit bereit. Da die Bilddaten entweder direkt in Binärfeldform oder als ASCII-Werte codiert zum PC übertragen werden und man auf diese Art vom übertragenen Bild nichts sieht, habe ich eine sehr einfach gehaltene PC-Anwendung entwickelt, mit der sich das Shield steuern lässt. Außerdem ermöglicht es das sofortige Betrachten der Bilder.

## Erweiterungen und Ideen

Durch Kombinieren des Kamera-Shields mit einem oder mehreren Shields aus [6] ergeben sich damit viele weitere Möglichkeiten. So entsteht beispielsweise mit einem zusätzlichen Shield, das einen Speicherkartensteckplatz zur Verfügung stellt, eine Kamera, die für Ballonmissionen geeignet ist und zum Beispiel in einem festgelegten Intervall ein Bild auf der Speicherkarte ablegt.

Findet ein *Audio Shield* Verwendung, lässt sich eine SSTV-Kamera realisieren. Und bei Verwendung einer leistungsstarken Prozessorplatine ist anschließend sogar eine Bearbeitung des Bildes bis zu einem gewissen Grad möglich. Denkbar wäre auch eine Auswertung des Bildes, wie sie zur Be-



**Tabelle 1: Brücken**

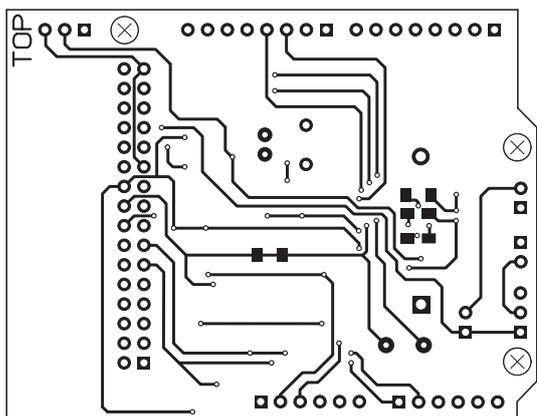
Brücke	Funktion; Standard
Jx1	Takt für MAX7456 von Kamera (SMD) beziehen; Standard: geschlossen
ViSEL (Jumper)	Kamera-Video oder externes Videosignal für MAX7456 verwenden; Standard: 1-2 geschlossen
VoSel (Jumper)	Kamera-Video oder Videosignal des MAX7456 ausgeben; Standard: 2-3 geschlossen

**Tabelle 2: Anschlüsse**

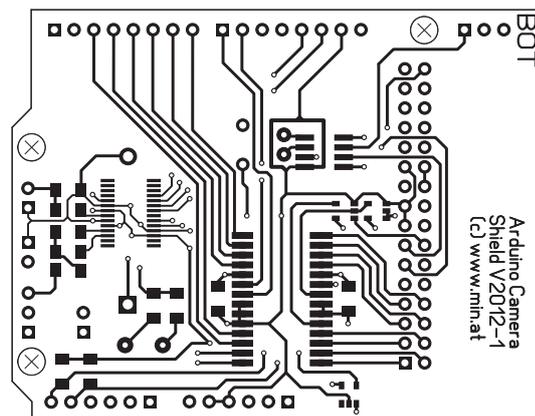
Pin	Beschreibung
VIN	Video-Eingang, zweipolig
VOU	Video-Ausgang, zweipolig
JRTC	RTC, externe Batterie und Takt, dreipolig

**Tabelle 3: Stückliste**

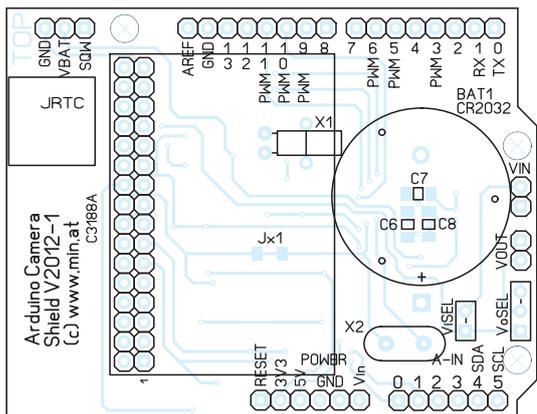
Bauteil	Wert/Bezeichnung
BAT1	3 V, CR2032
C1, C2, C5...C8	100 nF
C3, C4	47 µF
C9, C10	22 pF
CAM	Kameramodul OV7620 (CAM-C3188A)
IC1	AL422B
IC2, IC3	SN74AHCT1G00
IC4	TC74
IC5	DS1307
IC6	MAX7456
R1, R2	3,3 kΩ
R3, R4	75 Ω
X1	32,768 kHz
X2	27 MHz



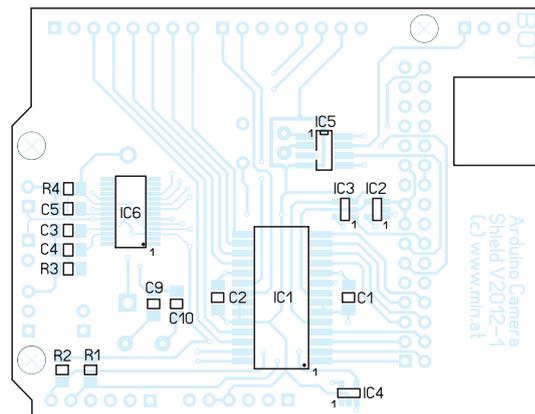
**Bild 5:**  
Layout der  
Platinoberseite  
(Bestückungsseite für  
bedrahtete Bauteile) des  
Arduino Camera Shields;  
Abmessungen 68 mm x 53 mm,  
M 1:1



**Bild 7:**  
Layout der Platinenunterseite  
(Bestückungsseite für  
SMD-Bauteile) des  
Arduino Camera Shields



**Bild 6:**  
Bestückungsplan des  
Arduino Camera Shields  
für bedrahtete Bauteile;  
vor dem Einlöten des runden  
Halters für die Knopfzelle sind  
noch drei SMD-Kondensatoren  
aufzulöten.



**Bild 8:**  
Bestückungsplan  
des Arduino Camera Shields  
für die restlichen SMD-Bauteile

wegungs- oder Objekterkennung erforderlich ist.

Das Arduino Camera Shield ermöglicht es, auch bei eher leistungsschwachen Mikroprozessoren mit Bildern und Videodaten umzugehen. Allerdings gilt: Je leistungsfähiger der verwendete Controller, desto mehr Funktionalität ist möglich.

Weitere Informationen sowie die aktuelle Firmware für das Kamera-Shield sind bei [10] zu finden.

oe1rib@min-at

**Literatur und Bezugsquellen**

- [1] Prinz, R., OE1RIB: Universelles PIC-Board als Grundlage für SSTV-Modul. FUNKAMATEUR 56 (2007) H. 7, S. 748–750
- [2] Prinz, R., OE1RIB: SSTV-Sendemodul auf Basis eines universellen PIC-Boards. FUNKAMATEUR 56 (2007) H. 8, S. 868–870
- [3] Sander, K.: Sonnenverfolger mit Arduino zur Nachführung von Solarzellen. FUNKAMATEUR 58 (2009) H. 10, S. 1078–1080
- [4] Seidenberg, C.: MEMS – Mikroelektromechanik im Dienste der Messtechnik. FUNKAMATEUR 60 (2011) H. 7, S. 720–721

- [5] Arduino: [www.arduino.cc](http://www.arduino.cc)
- [6] Adafruit: Arduino Shields. [www.adafruit.com](http://www.adafruit.com) → Products → Arduino → Shields
- [7] GHI Electronics: FEZ Panda II Board. [www.ghielelectronics.com](http://www.ghielelectronics.com) → Catalog → .NET Micro Framework → Other
- [8] Sparkfun: Breakout Board for MAX7456 On Screen Display. [www.sparkfun.com](http://www.sparkfun.com) → Categories → Breakout Boards
- [9] Arduino: Arduino Duemilanove: [www.arduino.cc/en/Main/arduinoBoardDuemilanove](http://www.arduino.cc/en/Main/arduinoBoardDuemilanove)
- [10] Prinz, R., OE1RIB: A Universal Arduino Camera Shield: [www.min.at/prinz/oe1rib/ArduinoCam](http://www.min.at/prinz/oe1rib/ArduinoCam)

## CAT-Interface für Lowe HF-150

Auf der Suche nach Informationen über den Empfänger Lowe HF-150 fand ich einen interessanten Link [1]. Nick Bailey hat sich des leidigen Themas der PC-Schnittstelle dieses Geräts angenommen und eine eigene Lösung entwickelt. Der nicht gerade kostengünstige HF-150 wird meines Wissens schon lange nicht mehr produziert. Es handelt sich beim IF-150 um ein Interface mit USB-Schnittstelle mit einem

Arduino Uno R3 als Basis. Dieses Controllermodul ist sehr preiswert z. B. bei eBay ([www.ebay.de](http://www.ebay.de)) für etwa 30 € oder weniger erhältlich.

Nick Bailey hat eine Firmware erstellt, die nach einer Registrierung zum Herunterladen bereitsteht. Ursprünglich wollte er das Interface mit Firmware selbst produzieren und kommerziell vertreiben, doch die EU-Vorgaben (CE und UL) stellen zu hohe An-

forderungen, sodass er seine Entwicklung als Selbstbau vorstellt.

Das Arduino-Board ist nach der Beschreibung auf [1] nur um einige Bauteile für die galvanische Trennung und Senkung der Störeinflüsse durch den PC zu ergänzen, siehe Bild. Das sollte jedem halbwegs geübten Bastler gelingen.

Das Interface ist eine erfreuliche Entwicklung für jeden SWL, der noch kein PC-Interface für den HF-150 besitzt. Eine Anmerkung: Das Original-Interface IF-150 von Lowe ist dafür bekannt, dass es sehr langsam arbeitet. Das ist auch bei Nick Baileys Entwicklung so. Der Grund dafür ist das Konzept der Schnittstelle im HF-150, die mit einer Pulsphasenmodulation arbeitet und das Timing vorgibt.

**Erwin Fileschi**  
erwin.fileschi@t-online.de

**Literatur und Bezugsquelle**

- [1] Bailey, N.: IF-150 USB-Serial Interface. [www.nick-bailey.co.uk/if150/index.html](http://www.nick-bailey.co.uk/if150/index.html)

