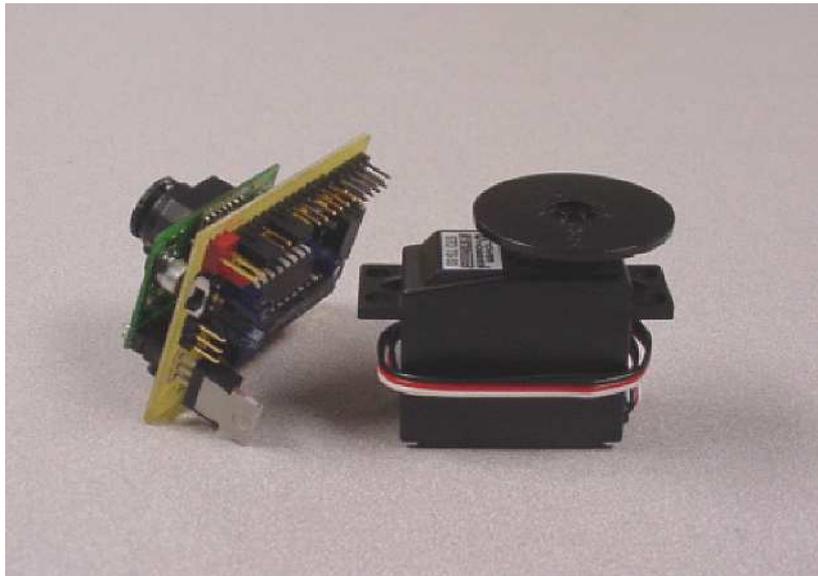


## CMUcam



# Benutzerdokumentation

Einführung	3
Allgemeine Informationen	
Allgemeines zur CMUcam	4
Inbetriebnahme	5
Überprüfung der CMUcam	6
Scharf stellen der CMUcam mit dem Java Demoprogramm	7
Demo Modus	8
Anmerkungen zur besseren Performance beim Verfolgen von Farben	9
Über die CMOS Kamera	10
Allgemeine Fehlerhinweise	29
Hardware	
Leiterplatten-Layout	11
Ansicht der bestückten CMUcam Leiterplatte	12
Schnittstellen	13-15
Beschreibung der Jumper	16
Bauteilliste	30
Schaltung	31
Kommunikation	
Befehlsbeschreibung	17-25
Beschreibung der Datenpakete	26-27
Firmware Upgrade	28

## Einführung

Die CMUcam, entwickelt von der Carnegie-Mellon-Universität in Pittsburgh, PA, enthält auf einer Platine einen 75MHz SX28 Microcontroller, der über einen privaten digitalen Bus von einem OV6620 Omnivision CMOS Kameramodul die Bilddaten erhält. Das Board kommuniziert über die RS.232 oder serielle TTL Schnittstelle und überträgt vorverarbeitete high-level Informationen aus dem Video-Datenstrom.

Folgende Funktionen sind realisiert:

- Verfolgen von benutzerdefinierten Farbobjekten mit 16,7 Bildern pro Sekunde
- Berechnung des Schwerpunktes eines Bildobjektes
- Ermitteln der durchschnittlichen Farbe und Varianz eines Bildbereichs
- Übermitteln eines Binärbildes der verfolgten Bildobjekte in Echtzeit (Line-Mode)
- Setzen beliebiger Bildausschnitte
- Einstellung der Bildparameter des Omnivision-Kameramoduls
- Übertragen eines Rohbildes mit Auflösung von 80x143 Pixel
- Schnittstellengeschwindigkeit: 115.200, 38.400, 19.200, 9.600 bps
- Möglichkeit der parallelen Bildverarbeitung mit einem untergeordnetem Prozessorboard über einen gemeinsamen Kamerabus
- Automatische Verfolgung einer Farbe mit Ansteuerung eines Modellbauservos
- Anschluss eines RC-Modellbauservos oder Nutzung des Ports als 1 Kanal Digital I/O

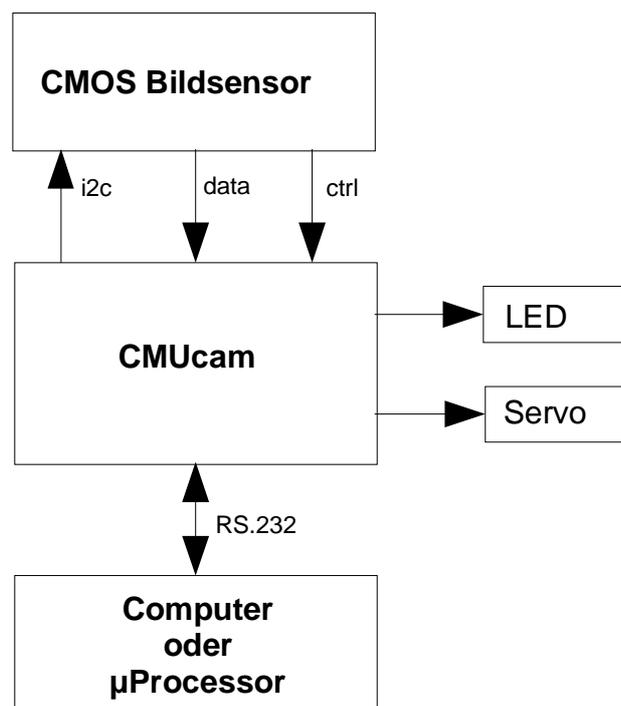
## Allgemeines zur CMUcam

Die CMUcam kann eingesetzt werden, um eine Farbe zu verfolgen oder zu überwachen. Die besten Ergebnisse erreicht man, bei hohem Kontrast und intensiven Farben. Zum Beispiel ist es einfach einen roten Ball vor einem weißen Hintergrund zu erkennen aber schwieriger bis unmöglich zwischen verschiedenen Nuancen einer Farbe bei wechselndem Licht zu unterscheiden.

Das verfolgen intensiver Farbobjekte kann für Robotikanwendungen sinnvoll sein um Landschaften zu erkennen, Linien zu verfolgen oder eine ortsveränderliche Bake zu suchen. Mit Statistiken über die Farbverteilung, ist es möglich eine Szene wieder zu erkennen, eine bestimmte Farbe zu finden oder eine einfache Bewegungserkennung durchzuführen. Mit der Betriebsart „Line Mode“ kann die Kamera niedrig auflösende Binärbilder in Echtzeit übertragen. Damit lässt sich anspruchsvolles verfolgen von Linien, mit Erkennung von Abzweigungen, bis hin zu einfacher Mustererkennung realisieren. Diese anspruchsvolleren Operationen erfordern eigene Algorithmen die die Nachbearbeitung der von der CMUcam gelieferten Binärbilder auf dem Hostprozessor durchführen. Diese Art von Nachbearbeitung erfordert als Hostprozessor einen Computer oder mindestens einen schnellen Microcontroller.

Die Standardanwendung für die CMUcam ist die Verbindung mit einem Hostprozessor über die serielle Schnittstelle. Dieser Hostprozessor kann ein Computer, PIC, Basic Stamp, Handy Board, Conrad C-Control, Brainstem oder ähnliches Microcontroller System sein. Die CMUcam kann das Microcontrollersystem um einfache Bildverarbeitungsaufgaben ergänzen, sie erreicht jedoch nicht die Leistungsfähigkeit von computergestützten Bildverarbeitungssystemen.

Das Kommunikationsprotokoll unterstützt auch die langsamsten Mikroprozessoren. Falls das System keine RS.232-Pegel unterstützt, kann auch mit TTL-Pegeln gearbeitet werden. Dabei findet die gleiche Kommunikation statt, jedoch mit Pegeln von 0 und 5 Volt. Die CMUcam unterstützt verschiedene Baudraten um auch langsamere Prozessoren anschließen zu können. Gerade für langsamere Prozessoren kann die Kamera in einem Abfragemodus betrieben werden. (siehe PM-Befehl auf Seite 22) In diesem Modus fordert der Prozessor ein einziges Datenpaket an. Das erleichtert bei langsameren Prozessoren Synchronisation mit der CMUcam. Auch ist es möglich, Pausen zwischen den übertragenen Bytes einzufügen (siehe DM-Befehl auf Seite 19). Allerdings verlangsamen sowohl der Abfragemodus als auch die Pausen die mögliche Bildrate der CMUcam.



## Inbetriebnahme

Die CMUcam wird normalerweise als Bausatz oder Fertigmodul geliefert. Sofern die Kamera als Bausatz geliefert wurde, so ist dieser zunächst aufzubauen. Das setzt Erfahrungen im Bestücken von elektronischen Leiterplatten voraus. Die Abbildung auf Seite 12 zeigt das fertig aufgebaute Modul. In den meisten Fällen ist der Scenix SX28 Chip, der mit der CMUcam geliefert wird, bereits programmiert. Es ist also nicht notwendig, die SX-Key Programmierumgebung zu erwerben, es sei denn Sie beabsichtigen selbst Firmware Updates durchzuführen oder den Programmcode zu modifizieren. Wenn das Board aufgebaut ist, wird noch ein serielles Kabel und ein Stromversorgungskabel benötigt. Die CMUcam kann mit einer Spannung von 6-9V betrieben werden, die Stromaufnahme beträgt bis 200mA. Geeignet sind Steckernetzgeräte mit Gleichspannungsausgang oder Batterien, die in jedem Elektronikladen zu erwerben sind.

Die Kamera besteht aus zwei Hauptkomponenten, der CMUcam Leiterplatte und dem CMOS Kameramodul. Das Kameramodul muss mit der Leiterplatte für die korrekte Funktion verbunden sein. Die Ausrichtung ist auf dem Deckblatt dieser Dokumentation zu sehen.

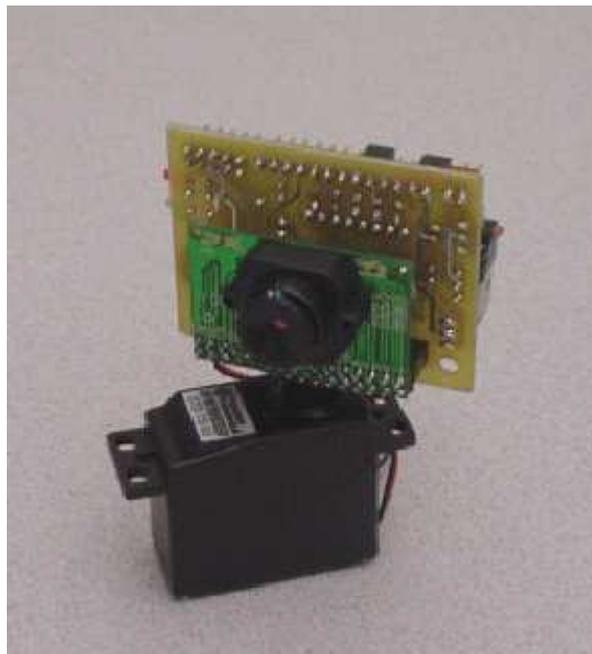
Nach dem zusammenbauen sollten nochmal alle Verbindungen sorgfältig überprüft werden. Stellen Sie sicher, dass der Pluspol der Stromversorgung von der Leiterplatte weg zeigt. Falls die Kamera mit einem Netzteil geliefert wird, stellen Sie sicher, dass der Stecker richtig herum eingesteckt wird. Dann stellen Sie die serielle Verbindung zu Ihrem Computer her. Diese Verbindung ist erforderlich, um die Kamera zu überprüfen und die Schärfe einzustellen (siehe auch der Abschnitt über die serielle Verbindung auf Seite 13). Überprüfen Sie, ob der Programmierschalter ausgeschaltet ist. Falls die CMUcam nur einen Schalter hat, wurde der Programmierschalter durch einen Jumper ersetzt. Der Schalter oder Jumper muss sich in der Aus-Position (rechte Position) befinden.

Wenn alle Verbindungen hergestellt sind, schalten Sie die CMUcam ein. Wenn alles OK ist, sollte die Power-LED (rot) leuchten. Jetzt sind Sie bereit, die CMUcam zu überprüfen (siehe nächste Seite). Wenn die LED nicht leuchtet, überprüfen Sie nochmals alle Verbindungen und stellen Sie fest, ob die Stromversorgung korrekt arbeitet.

## Überprüfung der CMUcam

Nachdem die CMUcam aufgebaut wurde, ist es eine gute Idee, das Modul zu überprüfen. Dazu muss die CMUcam mit der seriellen Schnittstelle eines PCs verbunden werden.

- Schritt 1: Falls noch nicht getan, stellen Sie Kabel für Stromversorgung und serielle Verbindung her (siehe auch voriges Kapitel).
- Schritt 2: Stellen Sie beide Verbindungen her.
- Schritt 3: Starten Sie auf dem PC ein Terminal Programm, z.B. Hyperterm (Windows) oder Minicom (Unix)
- Schritt 4: Setzen Sie im Terminal Programm die Kommunikation auf 115.200 Baud, 8 Datenbits, 1 Stopbit, keine Parität, Lokales Echo an, und falls möglich erlauben Sie "add line feed" (add \n to a received \r). Diese Einstellungen erscheinen normalerweise unter Einstellungen für seriellen Port oder ähnlicher Menüoption.
- Schritt 4: Schalten Sie das CMUcam Modul ein; die Power-LED sollte leuchten, die Track-LED sollte nicht leuchten.
- Schritt 5: Auf dem PC sollte im Terminalprogramm folgendes angezeigt werden  
: CMUcam v1.12  
:  
Wenn das angezeigt wird, konnte die Kamera vom CMUcam Modul richtig initialisiert werden und die Firmware wurde gestartet.
- Schritt 6: Tippen Sie gv ein, gefolgt von der Enter Taste. Es sollte folgendes angezeigt werden:  
:gv  
ACK  
CMUcam v1.12  
:  
Das ist die aktuelle Firmwareversion, weiterhin wird gezeigt, dass sowohl die Sende- als auch die Leserichtung der seriellen Schnittstelle funktionieren.
- Schritt 7: Stellen Sie die Kamera Optik scharf unter Nutzung der CMUcam java GUI oder eines eigenen Anzeigeprogramms. Normalerweise ist das Bild scharf wenn die Linse einige Umdrehungen von der nahesten Position am CMOS Sensor herausgedreht wurde. Fortwährendes Drehen der Linse und darstellen des Bildes führt zum Erfolg. (siehe auch Scharf stellen mit der CMUcam GUI auf der nächsten Seite)
- Schritt 8: Betrieb der CMUcam im Demo Modus (siehe Seite 9).



## Scharf stellen der CMUcam mit dem Java Demoprogramm

Bei der ersten Inbetriebnahme der CMUcam wird die Optik meistens noch nicht richtig scharf gestellt sein. Der einfachste Weg um das zu tun ist die Verwendung eines grafischen Benutzerinterfaces das die Bilder der CMUcam darstellen kann. Eine Möglichkeit ist die CMUcamGUI, ein java Programm, das auf der CMUcam-Webseite zur Verfügung gestellt wird.

### Schritt 1: Testen ob java installiert ist

Als erstes muss herausgefunden werden, ob java auf dem Computer bereits läuft. Gehen Sie auf den Start-Button unter Windows und wählen Sie Programme | Zubehör | Eingabeaufforderung. Unter Unix, Linux oder späteren Versionen von Mac OS öffnen Sie eine Shell.

Jetzt tippen Sie in die Kommandozeile: `java -version`

Wenn eine Ausschrift erscheint mit `java version 1.x.xx` dann ist java auf Ihrem PC bereits installiert! Falls dagegen der Befehl nicht gefunden wird („not found“) oder eine ähnliche Ausschrift erscheint, dann ist es erforderlich, von `java.sun.com` eine Kopie der Java VM zu laden (J2SE, JDK, JRE sind verschiedene Java-Produkte, die alle die Voraussetzungen zum Betrieb der CMUcamGUI mitbringen). Sun hat verschiedene plattformspezifische Anweisungen, wie java zu installieren ist. Auch ist es erforderlich, dass Ihre Version von java neuer ist als v1.3. Falls nicht, müssen Sie eine neuere installieren.

### Schritt 2: Ausführen der CMUcamGUI

Nachdem java installiert ist, benötigen Sie eine aktuelle Kopie des CMUcamGUI java Programms, die Sie z.B. von der CMUcam Webseite downloaden können. Packen Sie die CMUcamGUI.zip Datei aus. Jetzt können Sie zu der in Schritt 1 beschriebenen Shell zurückkehren. Navigieren Sie zum CMUcamGUI Verzeichnis, in das Sie die Dateien entpackt haben. Sie können mit den Befehlen `dir` (DOS) oder `ls` (UNIX) den Inhalt des aktuellen Verzeichnisses anzeigen lassen. Sind Sie im gewünschten Verzeichnis angekommen überprüfen Sie das Vorhandensein der Datei `CMUcamGUI.class`. Falls es nicht vorhanden sein sollte, dann wurde die CMUcamGUI.zip nicht richtig entpackt oder Sie sind im falschen Verzeichnis. Wenn die Datei vorhanden ist, tippen Sie `java CMUcamGUI` (hierbei auf Groß-/Kleinschreibung achten!).

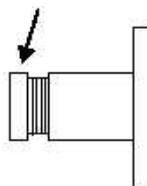
### Schritt 3: Ein Bild herunterladen

Als erstes wird nach der seriellen Schnittstelle gefragt. Unter Windows geben Sie die Nummer des COM-Ports an dem die CMUcam angeschlossen ist ein und klicken Sie OK. Unter Unix (Linux), überprüfen bzw. korrigieren Sie den angegebenen Pfad und klicken Sie OK. Die CMUcamGUI Oberfläche sollte sich nun öffnen, das Textfenster (in der Mitte des Schirms) sollte die Meldung ausgeben: *Camera OK and idle* .

Das heißt das die Kamera gefunden wurde und die Kommunikation funktioniert. Wenn das funktioniert, gehen Sie in das *Commands* Menü und wählen *Dump Frame* . Nach ein paar Sekunden sollte ein Bild angezeigt werden.

### Schritt 4: Scharf einstellen

Hier drehen



CMUcam Objektiv

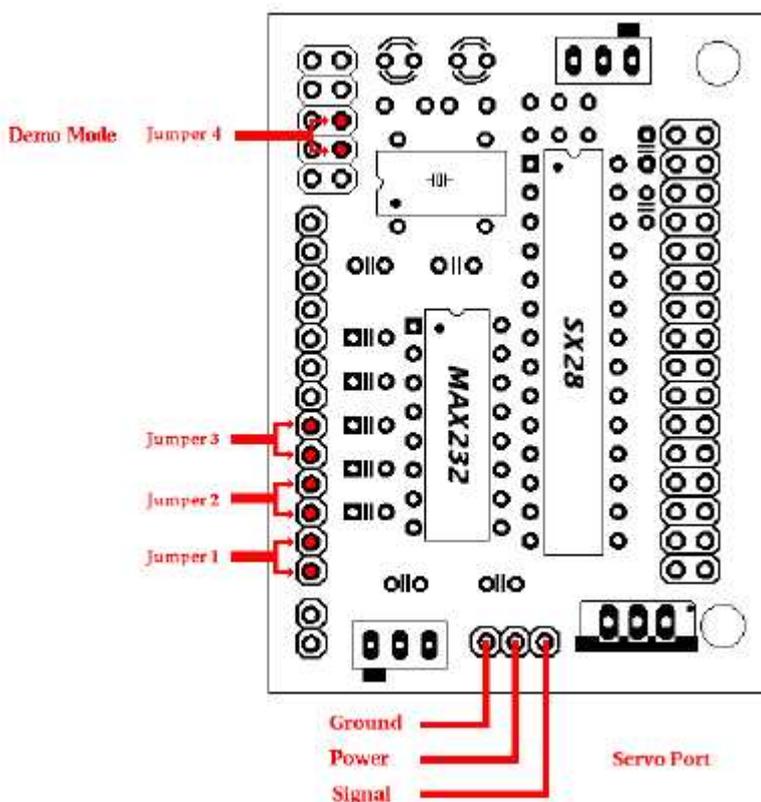
Nachdem Sie in der Lage sind, Bilder von der Kamera zu laden, können Sie die Schärfe des Bildes durch drehen am Frontring des Objektivs verändern. Versuchen Sie die Schärfe durch kleine Änderungen und nachladen des Bildes so gut wie möglich einzustellen. Normalerweise ist das Bild scharf wenn die Linse einige Umdrehungen von der nahesten Position am CMOS Sensor herausgedreht wurde.

## Demo Modus

Jumper 4 bringt die Kamera in den Demo Modus. Nach dem Einschalten wird der TW (Track Window) Befehl aufgerufen und ein Standard RC-Hobby-Servo angesteuert. Das Servo versucht die Kamera auf den Mittelpunkt der beim einschalten erkannten Farbe nachzuführen. Es ist nur erforderlich das Servo gemäß Skizze anzuschließen. Sinnvoll ist es das Servo vor dem Einschalten auf Mittelstellung auszurichten.

Beachten Sie bitte, dass nach dem Einschalten 5 Sekunden gewartet wird, bevor die Farbe bestimmt und das Servo angesteuert wird. Beim ersten einschalten sind Auto-Verstärkung und Auto-Weißabgleich aktiv. Wenn sich die Werte nach 5 Sekunden stabilisiert haben, schaltet die Kamera die Automaten ab, wechselt in den YCrCb Modus und führt den TW-Befehl aus. (siehe Seite 25 für TW Details)

Diese automatische Servoansteuerung kann auch manuell (ohne Jumper) gewählt werden. Dazu geben Sie über die serielle Schnittstelle den Befehl MM 2 und rufen anschließend TW auf. Wenn das Servo in die falsche Richtung zu laufen scheint, setzen Sie zusätzlich Jumper 1. Wenn Jumper 1 und Jumper 4 gesetzt sind, lässt sich das Board nicht im Slave Modus betreiben; es läuft im Demo Modus mit umgekehrter Servodrehrichtung. Jumper 4 benutzt die Anschlüsse der seriellen Schnittstelle, deshalb ist im Demo Modus keine Kommunikation mit der Kamera möglich.



Folgende Schritte werden beim Einschalten im Demo Modus ausgeführt:

1. Kamera wird über RS-Befehl zurückgesetzt.
2. 5 Sekunden Pause um der Kamera Zeit für die automatischen Korrekturen zu geben.
3. Befehl "CR 18 32 19 32" wird ausgeführt. Das setzt den YCrCb Modus und schaltet die automatische Verstärkungsregelung aus.
4. Der "TW" Befehl wird ausgeführt.

## Anmerkungen zur besseren Performance beim Verfolgen von Farben

Die automatische Verstärkungsregelung (Auto-Verstärkung) ist eine interne Steuerung, die den Helligkeitspegel des Bildes entsprechend der Aufnahmesituation anpasst. Dabei wird versucht die hellen und dunklen Stellen des Bildes so anzupassen dass sich ein Bild mit einer mittleren Helligkeit ergibt. Dieser iterative Prozess läuft ständig, so dass, wenn sich die Umgebungshelligkeit ändert, weil z.B. ein Licht eingeschaltet wurde, die Kamera die Bildhelligkeit fortwährend anpasst.

Auf der anderen Seite versucht der automatische Weißabgleich (Auto-Weißabgleich) die Verstärkung der einzelnen Farbkanäle anzupassen. Die Beleuchtung erfolgt in der realen Welt niemals mit reinem weißem Licht. Das Kamerabild wird dementsprechend die Beleuchtungsfarbe hervorheben, es gibt einen Farbstich. Die Kamera beginnt mit einem Anfangswert für die Verstärkung jedes Farbkanals. Wenn der Auto-Weißabgleich aktiv ist, werden diese Werte von Bild zu Bild so angepasst, dass der Mittelwert aller Farben über das gesamte Bild einen Grauwert ergibt. Diese Methode arbeitet relativ gut. Ein Problem damit ergibt sich jedoch bei einfarbigen Flächen, es wird nicht der reale Farbwert zurückgeliefert, sondern nach Anpassung irgendein Grauwert. Entfernt man die einfarbige Fläche, so gehen die Verstärkungswerte der Farbkanäle langsam wieder auf ihre richtigen Werte zurück.

Beim Verfolgen von Farben, wie z.B. im Demo-Modus, empfiehlt es sich die Auto-Verstärkung und den Auto-Weißabgleich für eine kurze Zeit anschalten und dann wieder auszuschalten. In einer Zeitspanne von z.B. 5 Sekunden, kann die Kamera mit Hilfe der Automaten die Verstärkungswerte bestimmen. Das folgende Ausschalten der Kamera verhindert dann das unnötige ändern der Werte wenn z.B. ein Objekt zu nah an das Objektiv kommt oder Schatten entstehen usw.. Wenn Auto-Verstärkung und Auto-Weißabgleich nicht ausgeschaltet werden und die Kamera die Einstellungen für die RGB-Kanäle ändert, dann können die aktuellen Werte der zu verfolgenden Farbe außerhalb des Bereiches der ursprünglich spezifizierten Farbe liegen.

YCrCb (Helligkeit – Farbdifferenz Rot – Farbdifferenz blau) ist eine abweichende Farbdefinition im Gegensatz zur bekannteren RGB-Definition (Rot – Grün – Blau). Bei YCrCb werden die Helligkeitsdaten in einem separaten Kanal gehalten (entspricht einem Schwarz/Weiß-Bild). Wegen dieser Eigenschaft ist der YCrCb Farbmodus der Kamera weniger anfällig auf Helligkeitsänderungen. Wegen der unterschiedlichen Farbdefinition, sehen die Bilder in YCrCb nicht wie Standard RGB Bilder aus, wenn sie direkt als Bild dargestellt werden (z.B. mit dem CMUcamGUI Programm). Die RGB Kanäle werden als CrYCb dargestellt. So entspricht im YCrCb Modus der Rotanteil R dem roten Farbdifferenzsignal Cr, der Grün Wert G entspricht der Helligkeit Y und der Blauwert B entspricht dem blauen Farbdifferenzsignal Cb. Wenn z.B. ein rotes Objekt verfolgt werden soll, schaut man sich das Bild am besten in YCrCb Darstellung an. Es können dann die Grenzen für Cr und Cb bestimmt werden, den Y-Wert gibt man relativ großzügig an, um das Ergebnis unempfindlich gegen Helligkeitsdifferenzen zu machen.

Folgende Transformationen werden von der Kamera genutzt, um RGB nach YCrCb zu konvertieren:

RGB -> CrYCb

$$Y = 0,59G + 0,31R + 0,11B$$

$$Cr = R - Y$$

$$Cb = B - Y$$

## Über die CMOS Kamera

Nach dem Einschalten kann die Kamera bis zu 15 Sekunden benötigen, um sich automatisch an die Lichtverhältnisse anzupassen. Gravierende Änderungen der Lichtverhältnisse, wie z.B. An- und Ausschalten von Lichtquellen, kann eine ähnliche Anpassungszeit bewirken. Bei Nutzung der Kamera im Außen-Bereich, wird es wegen der kräftigen IR Emission der Sonne, auch bei bewölkten Tagen, unter Umständen notwendig sein ein zusätzliches IR-Filter oder ein Graufilter zu benutzen um den Helligkeitspegel zu reduzieren.

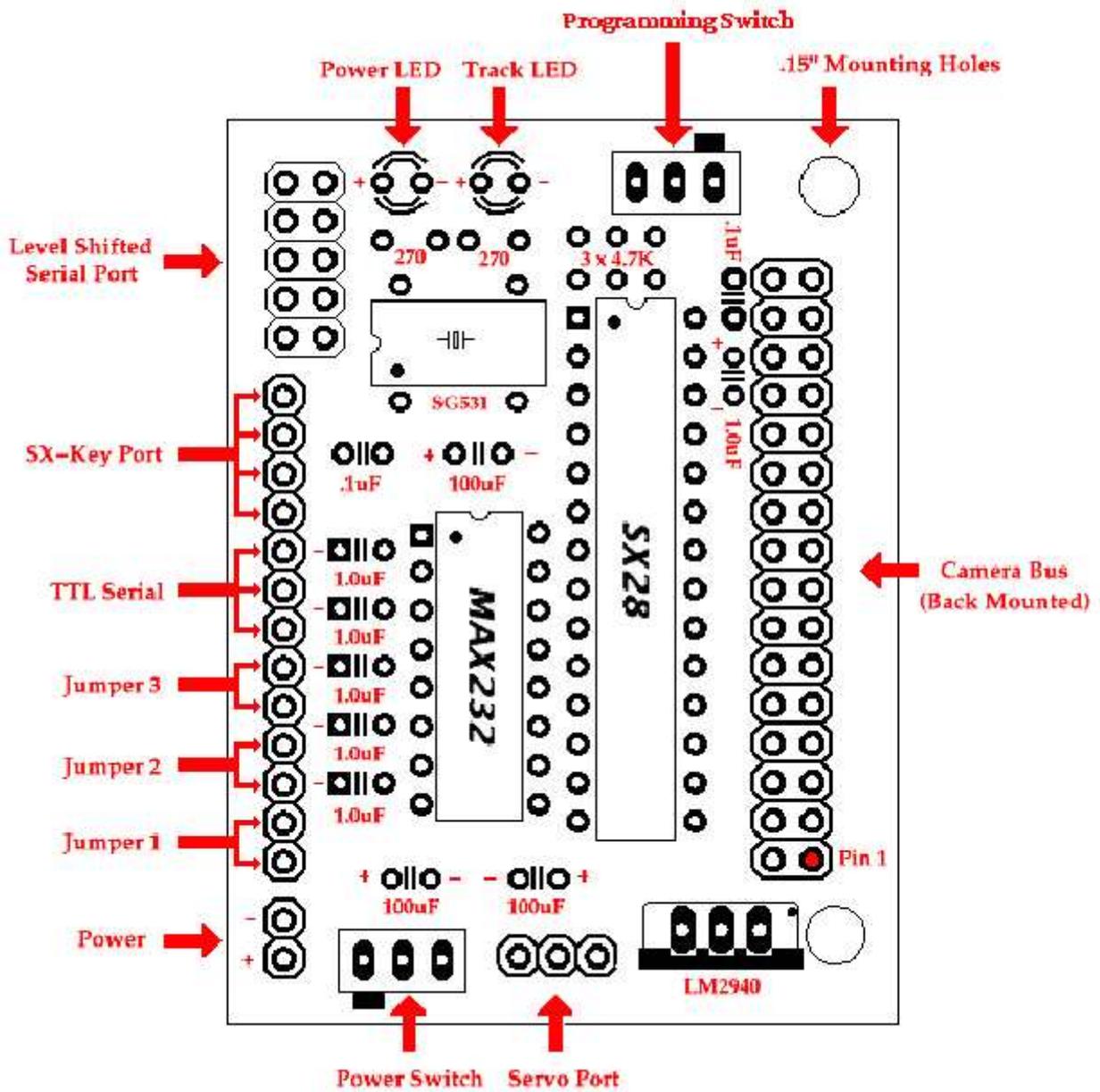
Die Funktionen, die von der Kamera bereitgestellt werden, sind eine Auswahl von verschiedenen Bildverarbeitungsfunktionen. Aktuelle Anwendungen können stark variieren und hängen von der Phantasie des Benutzers ab. Die Möglichkeit den Fensterausschnitt zu verändern, eine Farbe, Farb- oder Lichtstatistiken anzufordern und Farben zu verfolgen können ergänzt werden vom Host Prozessorsystem um weiter high-level Funktionen zu realisieren.

Eine wichtige Eigenschaft des CMOS Bildsensors ist der Rückgabewert von 16 bis 240 für jedes Pixel. Dieser Effekt ist zu beachten beim Verfolgen von Farben, holen von Farbmittelwerten oder holen von Bildern. Dieser eingeschränkte Wertebereich ist unabhängig vom Farbmodus der Kamera und besteht auch für den YCrCb Modus.

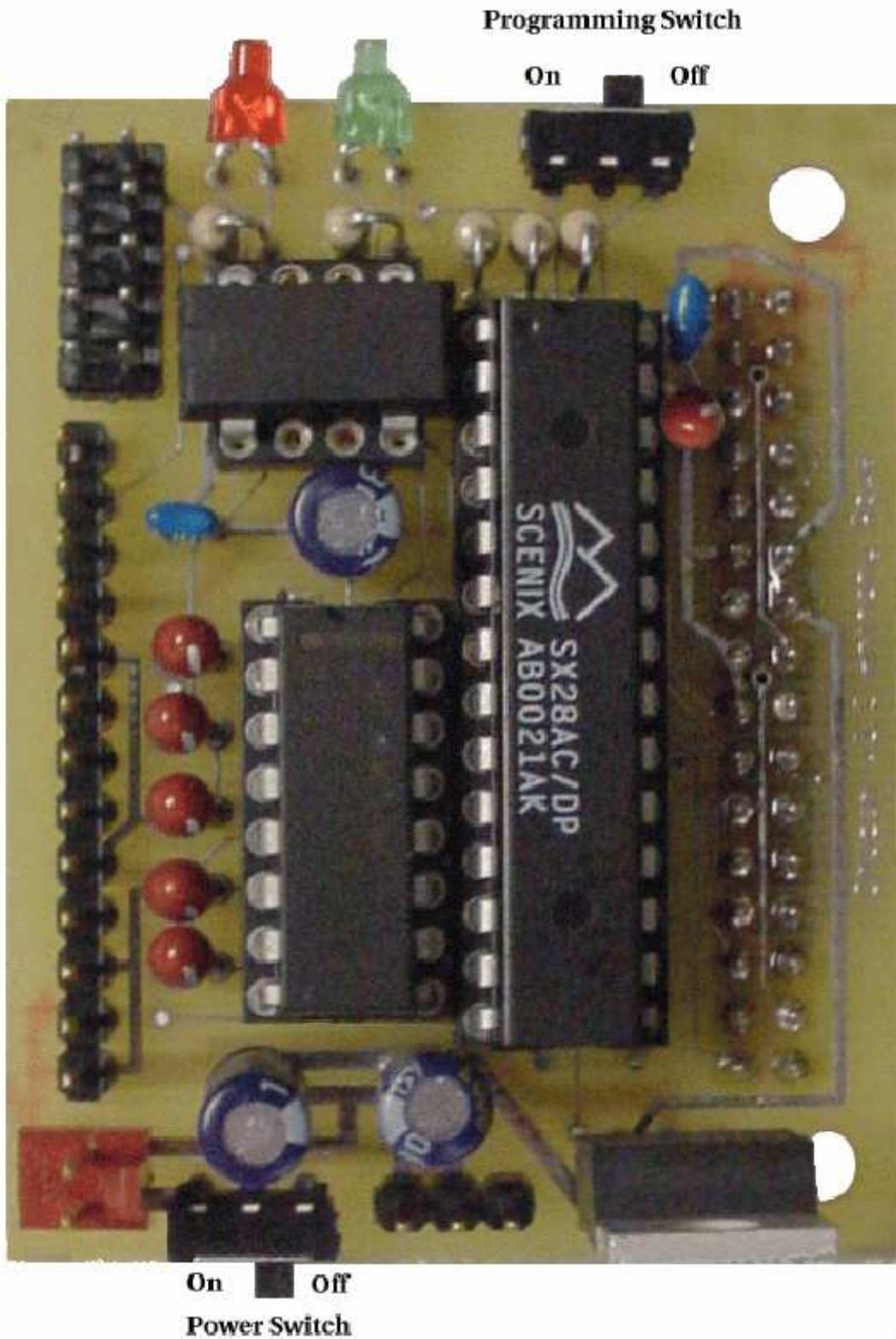
## Analog Video Ausgang

Pin 32 auf dem Kamerabus ist ein schwarz/weiß Analog Ausgang. Es ist möglich an diesem Ausgang einen TV oder Multi-Sync Monitor zu betreiben. Wegen des Systemtakts der Kamera, kann der Analogausgang nicht 100%ig mit PAL oder NTSC-Systemen synchronisieren. Wenn Sie einen Monitor zur Verfügung haben, der auch mit Nicht-Standard Signalen arbeitet, ist es möglich, parallel zur Bildverarbeitung das Kamerabild auf einem Monitor zu beobachten. Bei CMUcam-Leiterplattenversion 1.22 oder früher ist es notwendig, die Verbindung zwischen Pin 31 und 32 des Kamerabusses zu unterbrechen (das wurde in Version 1.23 behoben). Dann kann der Ausgang (Pin 32) mit einem 75 Ohm Koaxialeingang verbunden werden. Der Außenleiter des Koaxialkabels muss mit der Kameramasse (Pin 31) verbunden werden. Nach dem Einschalten passen Sie die Bildrate der Kamera so an, dass sich das beste Bild ergibt. Die meisten Standard-TV zeigen zumindest ein (vielleicht) flackerndes Bild bei einer Bildrateneinstellung von 0 (CR 17 0). Bitte beachten Sie, Bildrateneinstellungen größer als 17 fps schalten die Bildverarbeitungsroutinen der CMUcam aus. Auf diese Weise können zwar Bilder am Monitor dargestellt werden, aber es kann keine digitale Bildverarbeitung stattfinden.

# Leiterplatten-Layout



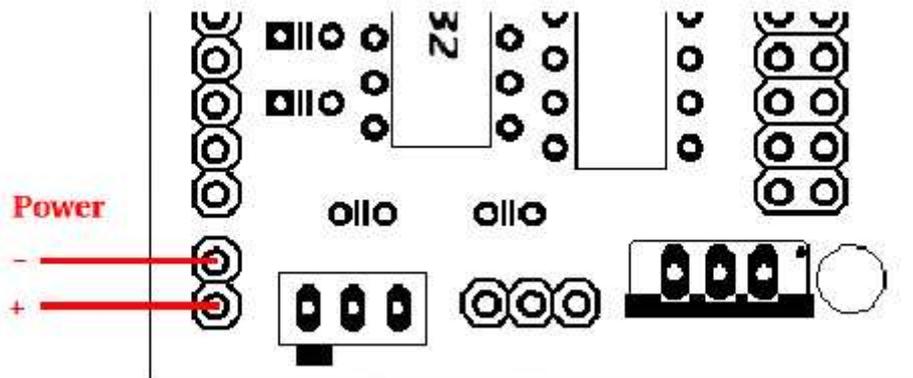
### Ansicht der bestückten CMUcam Leiterplatte



## Schnittstellen

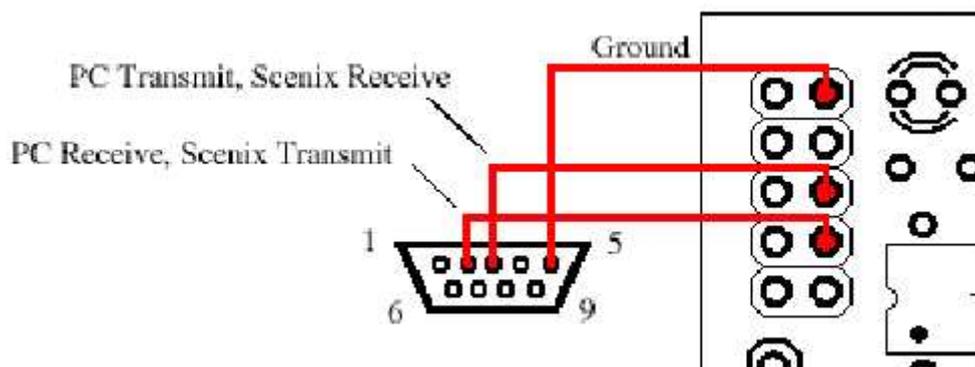
### Stromversorgung

Die Stromversorgung der CMUcam erfolgt über einen 5V Regler mit niedriger Dropout-Spannung (LM2940). Die Versorgungsspannung kann deshalb zwischen 6 und 9 Volt liegen. Je höher die Spannung, desto wärmer wird der Spannungsregler. Die Stromaufnahme beträgt ca. 200mA.



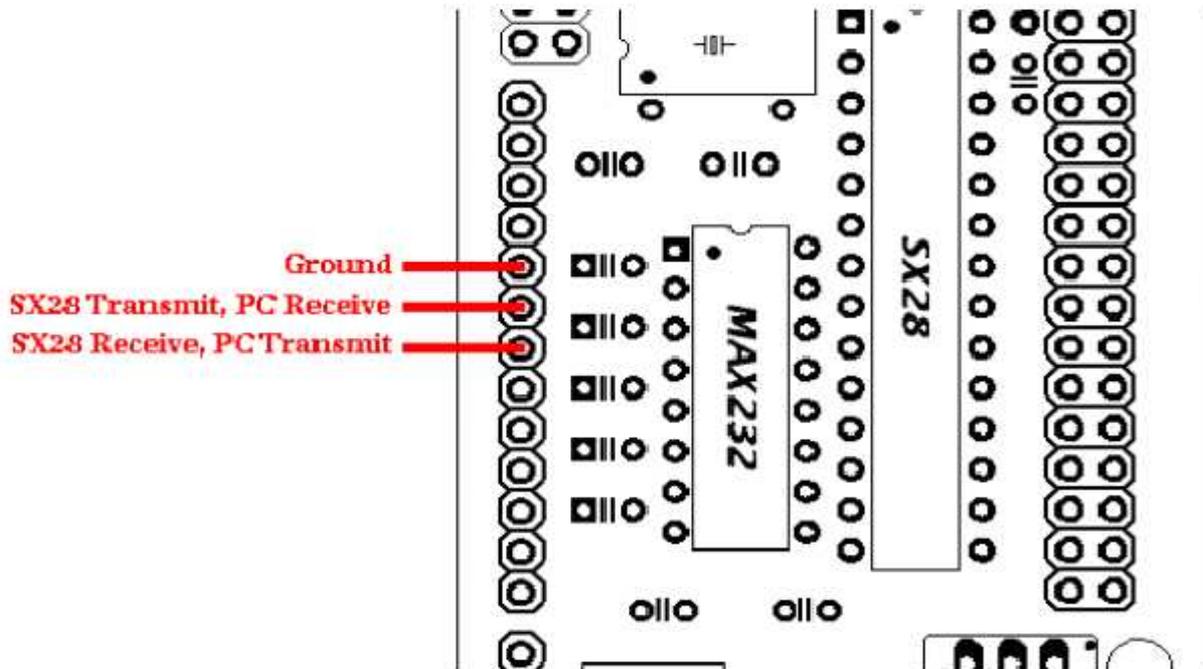
### Serielle Schnittstelle mit Pegelwandler (RS.232)

Dieser Port bietet eine serielle Schnittstelle mit Pegelwandlung (RS.232) zum Anschluss an einen Computer. Es werden nur 3 von 10 Leitungen genutzt, der 2 x 5 Pin Anschluss ist so konfiguriert, dass ein 9poliges Flachbandkabel mit Crimpverbindungen genutzt werden kann. Wenn das auf Anhieb nicht funktioniert, probieren Sie die Richtung, mit der der D-Stecker aufgecrimpt wird, zu vertauschen. Die Anschlussbelegung ist nochmal auf dem Bild dargestellt und kann mit einem Multimeter überprüft werden.



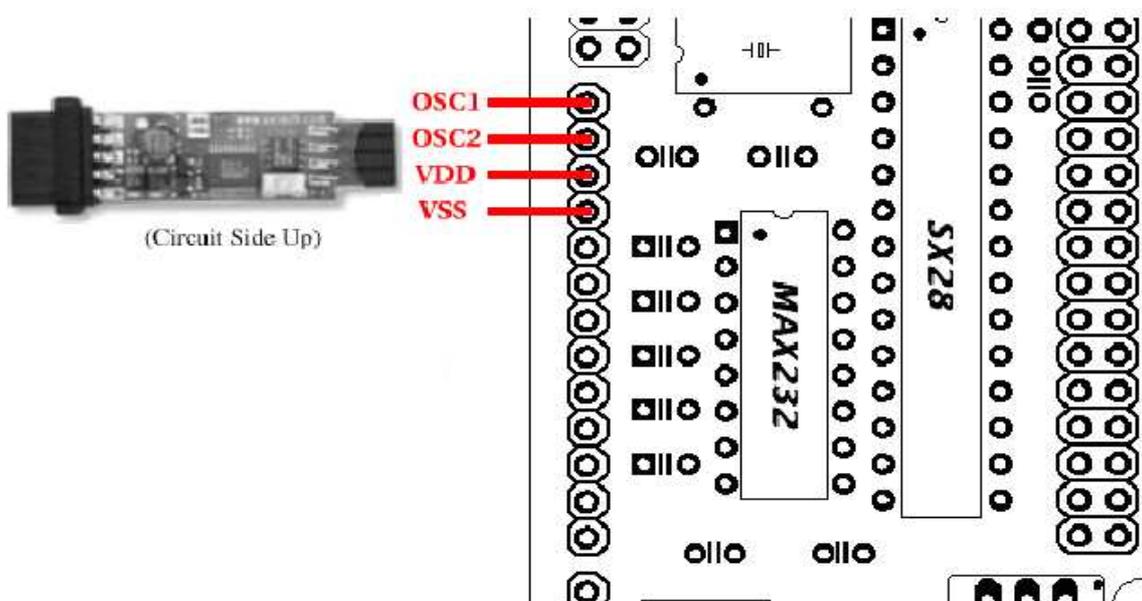
**Serielle Schnittstelle (TTL-Pegel)**

Für diese Schnittstelle ist die Verbindung SX28 – MAX232 herausgeführt. Dieser Port lässt sich sehr gut für die direkte Kommunikation mit Mikrocontrollern, mit TTL-Pegel, nutzen. Damit entfällt die mehrfache Pegelwandlung. Um diese Schnittstelle zu nutzen muss der MAX232 aus der Fassung entfernt werden.



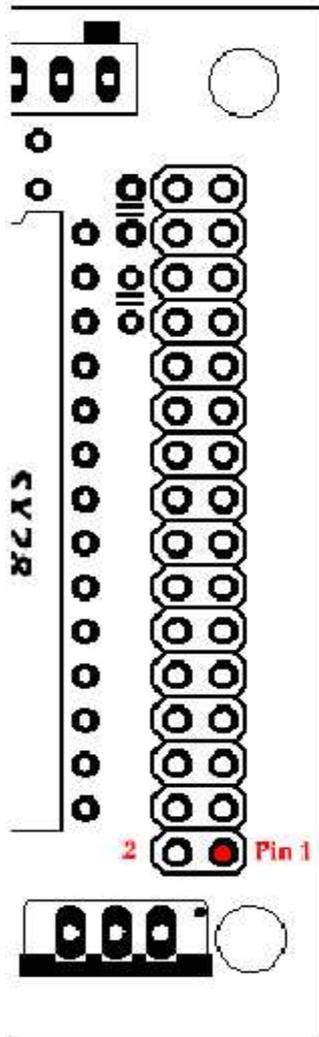
**Programmier-Port**

Über diesen Port kann die Firmware auf den SX28 gebracht werden (für Updates oder Änderungen). Voraussetzung ist ein SX-Key / SX-Blitz o.ä. Programmierinterface. Dieses kann von Parallax Inc ([www.parallaxinc.com](http://www.parallaxinc.com)) erworben werden. Der SX28 wird gewöhnlich vorprogrammiert ausgeliefert.



### Kamera-Bus

Dieser Bus ist die Schnittstelle zur CMOS Kamera. Die CMOS-Kamera ist parallel zur CMUcam Leiterplatte montiert und über einen 34poligen Steckverbinder JP5 verbunden. Dieser Steckverbinder (weiblich) sollte auf der Leiterseite der CMUcam bestückt sein. (siehe Ansicht der bestückten Leiterplatte) Für Informationen über den Video Analog Ausgang siehe Seite 10.

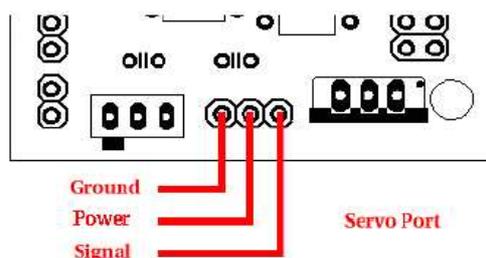


### Pin Description

1-8	Y0~Y7	Digital Output Y Bus
9	PWDN	Power Down Mode
10	RST	Reset
11	SDA	I2C Serial Data
12	FODD	Odd Field Flag
13	SCL	I2C Serial Clock
14	HREF	Horizontal Window Reference
15	AGND	Analog Ground
16	VSYN	Vertical Sync
17	AGND	Analog Ground
18	PCLK	Pixel Clock
19	EXCLK	External Clock
20	VCC	+5 VDC
21	AGND	Analog Ground
22	VCC	+5 VDC
23-30	UV0~UV7	Digital output UV bus
31	GND	Common Ground
32	VTO	Video Analog Output (75 Ohm)

### Servo Port

Das ist der Ausgang für einen RC-Modellbauservo. Die Stromversorgung für den Servo geht nicht über den 5V Regler. Deshalb sollte bei Versorgungsspannungen über 6V kein Servo angeschlossen werden.



## Beschreibung der Jumper

### Parallelverarbeitung im Slave-Modus (Jumper 1)

Die CMUcam unterstützt eine Betriebsart die es erlaubt, mehrere CMUcam Prozessorboards einzusetzen, die die Bilddaten eines CMOS-Kamerasensors gleichzeitig bearbeiten können. Dazu ist anstatt des doppelreihigen Steckverbinders JP5 ein Mehrfachsteckverbinder in PC104-Technik zu verwenden. Damit lassen sich mehrere Prozessorboards am gleichen Kamerabus betreiben. Wenn beim Einschalten des Prozessorboards Jumper 1 gesetzt ist, wird dieses Board zum Slave. Der Slave Modus bewirkt, dass keine Konfiguration oder andere Interaktion mit der CMOS-Kamera mehr durchgeführt wird. Stattdessen wird das über den Kamerabus gelieferte Bild bearbeitet. Am Bus muss es genau einen Master geben, alle anderen Boards müssen als Slave konfiguriert werden. In der momentanen Softwareversion gibt es bis auf die Bilddaten noch keine Kommunikation zwischen den Boards. Das heißt, zu jedem Board muss eine serielle Verbindung bestehen. Jedes Board ist wie eine einzeln betriebene CMUcam anzusprechen. So ist es z.B. möglich, über ein Board eine Farbe zu verfolgen, während das andere Board den Farbmittelwert eines Fensters zurück gibt. Jedes Board läuft autonom und nur das Master-Board kann die Kameraregister steuern.

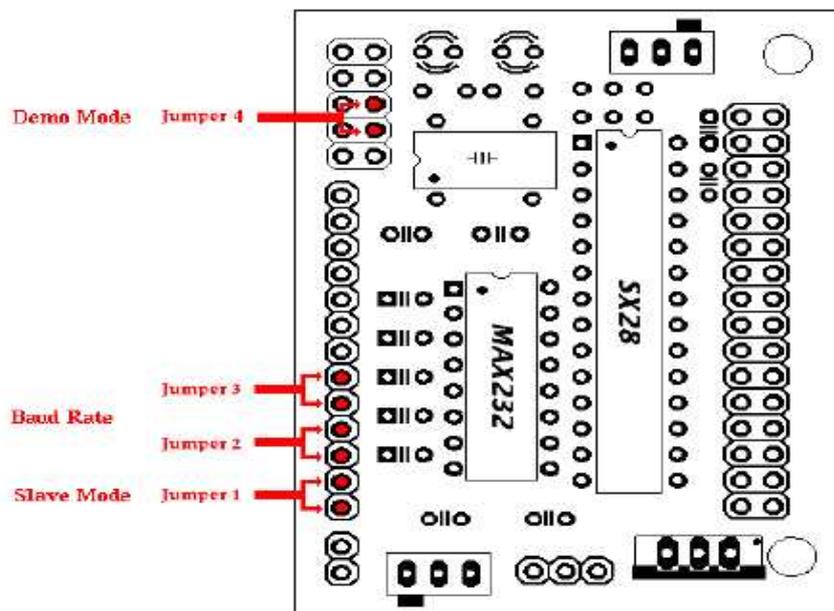
### Schnittstellengeschwindigkeit (Jumper 2 und 3)

115,200 Baud	Jumper 2 Offen	Jumper 3 Offen
38,400 Baud	Jumper 2 Gesetzt	Jumper 3 Offen
19,200 Baud	Jumper 2 Offen	Jumper 3 Gesetzt
9,600 Baud	Jumper 2 Gesetzt	Jumper 3 Gesetzt

Wegen der zusätzlichen Zeit, die die Übertragung mit den niedrigeren Baudraten benötigt können Bilder verloren gehen, was letztendlich zu einer niedrigeren Bildrate führt. Die niedrigere Baudrate führt ebenfalls zu einer niedrigeren Auflösung beim holen eines Bildes.

### Demo Modus (Jumper 4)

Jumper 4 bringt die Kamera in den Demo Modus. Nach dem Einschalten wird der TW (Track Window) Befehl aufgerufen und ein Standard RC-Hobby-Servo angesteuert. Das Servo versucht die Kamera auf den Mittelpunkt der beim einschalten erkannte Farbe nachzuführen. Für mehr Informationen siehe Demo-Modus Kapitel auf Seite 8.



## Befehlsbeschreibung

Die seriellen Schnittstellenparameter sind wie folgt:

- 115.200 Baud
- 8 Datenbits
- 1 Stopbit
- Keine Parität
- Keine Flow-Control (Kein Xon/Xoff oder Hardware)

Alle Befehle werden mit lesbaren ASCII-Zeichen übermittelt (123 sind 3 Bytes "123" ). Nach der Übertragung eines Befehls wird ACK an den Host-Prozessor zurückgesendet. Wenn es ein Syntax- oder Übertragungsproblem gab, wird NCK zurückgesendet. Nach einem ACK oder NCK wird ein Zeilenende \r zurückgesendet. Wenn ein Prompt ('\r' gefolgt von einem ':' ) gesendet wird heißt das, das die Kamera auf einen weiteren Befehl wartet (Idle State). Befehle und Argumente werden mit Leerzeichen getrennt. Das Zeilenende \r (ASCII 13 Carriage Return) wird benutzt um die Befehlszeile abzuschließen.

Wenn die Übertragung von lesbaren ASCII-Zeichen zu viel Zeit beansprucht (bei niedrigen Übertragungsraten) ist es möglich, verschieden Arten von Binärdatentransfer zu nutzen. (Siehe Binär-Modus Befehl, RM)

**\r**

Dieser Befehl wird benutzt um das Kameraboard in den Leerlauf-Zustand (idle state) zu setzen. Wie bei allen anderen Befehlen auch, wird entweder „ACK“ oder im Fehlerfall „NCK“ zurückgegeben. Nach dem Bestätigen des Befehls wartet die Kamera auf weitere Befehle, was durch den ':' Prompt angezeigt wird. Ein \r Befehl, während die Kamera im Leerlauf ist liefert ein "ACK" gefolgt von '\r' und einem ':' Prompt. Auf diese Weise kann die Kamera auch aus dem fortlaufenden Sendemodus (streaming) gestoppt werden.

Beispiel: Überprüfen ob die Kamera aktiv ist, während sie sich im Leerlauf befindet

```
: \r
ACK
:
```

**CR [ Reg1 Wert1 [Reg2 Wert2 ... Reg16 Wert16] ]\r**

Mit diesem Befehl werden die internen Register des CMOS Kameramoduls gesetzt. Die Registernummern und mögliche Werte sind in der Dokumentation des CMOS Bildsensors beschrieben. Alle Werte die mit diesem Befehl übertragen werden sind in dezimaler Form lesbar als ASCII Zeichen zu notieren sofern die Kamera nicht vorab in den Binär-Modus gesetzt wurde. Bis zu 16 Register - Werte Kombinationen können in einem Befehl übertragen werden. Vorherige Schreibbefehle werden nicht auf ihren Standardwert zurückgesetzt, jedoch werden die angegebenen Register mit den neuen Werten überschrieben. Der Befehl CR ohne Parameter setzt die Kamera zurück und bringt alle Registerwerte auf ihren Standardwert. Dieser Befehl kann benutzt werden um z.B. Verstärkungswerte direkt zu setzen oder andere Low-Level Eigenschaften zu verändern.

Beispiel: senken der internen Bildrate auf 9fps (Standard Bildrate ist 2 (17fps))

```
: CR 17 5\r
ACK
:
```

Häufig benutzte Einstellungen:

Register	Werte	Effekt
5 Kontrast	0-255	
6 Helligkeit	0-255	
18 Farb-Modus	36	YCrCb* Auto-Weißabgleich An
	32	YCrCb* Auto-Weißabgleich Aus
	44	RGB Auto-Weißabgleich An
	40	RGB Auto-Weißabgleich Aus (Standard)
	17 Bildrate	2
	3	13 fps
	4	11 fps
	5	9 fps
	6	8 fps
	7	7 fps
	8	6 fps
	10	5 fps
	12	4 fps
19 Auto Belichtung	32	Auto-Verstärkung Aus
	33	Auto-Verstärkung An (Standard)

- \* Aus dem roten Kanal R wird das rote Farbdifferenzsignal Cr, aus dem grünen Kanal G wird die Helligkeit Y und aus dem blauen Kanal B das blaue Farbdifferenzsignal Cb  
 RGB -> CrYCb  
 $Y = 0.59G + 0.31R + 0.11B$   
 $Cr = R - Y$   
 $Cb = B - Y$

**DF\r**

Dieser Befehl holt ein komplettes Bild von der Kamera über die serielle Schnittstelle (Dump Frame). Das ist der einzige Befehl, der immer ein Binärdatenpaket zurück liefert (Typ-F Paket). Dieses Paket enthält die binären Videodaten Spalte für Spalte mit einem Bild- und einem Spaltensynchronisationsbyte. (Das Paket wird z.B. gelesen und angezeigt von der CMUcamGUI Java Anwendung.) Weil die eigentlich notwendige Datenrate für ein Videobild weit höher ist, als die max. Geschwindigkeit der seriellen Schnittstelle, wird immer genau eine Spalte pro aktuellem Bild übertragen. Das bewirkt, dass sich bei bewegten Bildern der Inhalt von Spalte zu Spalte ändert. Um ein richtiges Seitenverhältnis zu bekommen, sollte bei der Darstellung jede Spalte verdoppelt werden. Die Kamera ist in der Lage ein Bild mit voller Auflösung bei max. Bildrate (17fps) nur bei 115.200 bps zu übertragen. Bei niedrigeren Baudraten, oder bei 115.200 bps mit zusätzlichen Verzögerungen muss die Bildrate herabgesetzt werden um die volle Auflösung übertragen zu können. Mit eingeschalteter Auto-Verstärkung kann bei niedrigen Bildraten das Bild am Anfang heller als gewöhnlich sein. Das ist eine Folge der längeren Adaptionzeit bei niedrigerer Bildrate. (Ein möglicher Ausweg ist die manuelle Korrektur von Helligkeit und Kontrast.)

Aufbau Typ-F Datenpaket

1 – Neues Bild; 2 – Neue Spalte; 3 – Bildende

RGB (CrYCb) Bereich von 16 - 240

1 2 r g b r g b ... r g b r g b 2 r g b r g b r ... r g b r g b ...

Beispiel: Holen eines Bildes mit einem Terminalprogramm (Achtung, die Binärdaten können beim Terminalprogramm zu temporären Darstellungsproblemen wegen undefinierten Zeichen führen.)

:DF\r

ACK

maKP(U A\$IU AL<>U A\$L\*YL%\*L L (G AUsonthAYA(KMAy098a34ymawvk....

**DM Wert\r**

Dieser Befehl setzt eine Pause vor jedem zu übertragenden Zeichen um die Synchronisation mit langsameren Host-Prozessoren zu vereinfachen. Der Wertebereich geht von 0 bis 255. Ein Wert von 0 (Standard) verursacht keine Verzögerung, 255 setzt die max. Verzögerungszeit. Die Einheit entspricht einer Übertragungszeit für 1bit bei der eingestellten Baudrate.

**GM\r**

Der Befehl gibt den Farbmittelwert des aktuellen Bildes zurück. Falls vorher ein Fenster definiert wurde, wirkt der Befehl nur auf dieses Fenster. Die Werte liegen zwischen 16 und 240 (siehe Seite 10). Weiterhin wird eine mittlere absolute Abweichung vom Farbmittelwert mit zurückgegeben. Der Mittelwert mit der Abweichung kann hilfreich sein für das automatische Verfolgung oder Erfassen einer Veränderung einer Szene. Im YCrCb-Modus werden die RGB Kanäle auf CrYCb abgebildet.

Aufbau Type-S Datenpaket

S Rmean Gmean Bmean Rdeviation Gdeviation Bdeviation\r

Beispiel: Mittelwert eines bestimmten Fensters bestimmen

:SW 1 1 40 143\r

ACK

:GM\r

ACK

S 89 90 67 5 6 3

S 89 91 67 5 6 2

**GV\r**

Dieser Befehl liefert die aktuelle Firmwareversion. Es wird ein ACK gesendet, gefolgt vom Firmware Versions-Text.

Beispiel: Abfrage Firmwareversion  
:GV\r  
ACK  
CMUcam v1.12

**HM Aktiv\r**

Dieser Befehl setzt die Kamera in einen Modus mit halber horizontaler Auflösung für den DF und LM Befehl. Ein Aktiv Wert von 1 bewirkt, dass nur jede ungerade Spalte ausgewertet wird. Der Standardwert 0 unterdrückt diesen Modus.

**I1\r**

Dieser Befehl nutzt den Servo-Port als digitalen Eingang. Ein Aufruf von I1 kehrt mit einer 0 oder 1 zurück, abhängig vom Spannungspegel des Servo-Ports. Die Leitung hat einen internen Pull-Up, so dass sie im Bedarfsfall nur gegen GND geschaltet werden braucht. Der Servo-Port kann ebenfalls als digitaler Ausgang genutzt werden. (siehe S1 Befehl)

Beispiel: lesen eines Wertes über den Servo-Port  
:I1\r  
ACK  
1

**L1 Wert\r**

Mit diesem Befehl wird die Tracking-LED (grün) gesteuert. Die möglichen Werte sind 0, 1 und 2 (Standard). Der Wert 0 schaltet die LED aus, während 1 die LED einschaltet. Der Wert 2 bringt die LED in einen Automatik-Modus. In dieser Betriebsart leuchtet die LED, wenn während des Verfolgens das Objekt im Bildbereich gefunden wird. Der Befehl lässt sich für Debug-Zwecke nutzen.

Beispiel: Ein und Ausschalten Tracking-LED  
:L1 1\r  
ACK  
:L1 0\r  
ACK

**LM Aktiv\r**

Dieser Befehl schaltet den Line-Modus ein, mit dem zusätzliche Infos über das Kamerabild übertragen werden. Im Line-Modus wird ein Binärdaten-Vorspann vor die C, M oder S Pakete gesendet. Dieser Modus ist für Anwender gedacht, die eigene Bildverarbeitungsalgorithmen über das Binärbild ausführen möchten. Weil die Übertragung der Binärdaten entsprechend der Bildrate erfolgt, muss die Verarbeitung mit einem leistungsfähigeren Host-Prozessorsystem erfolgen, als das für die anderen Befehle erforderlich ist. Der Line-Modus funktioniert mit den Befehlen TC und TW: wenn der Line-Modus aktiv ist und TC oder TW ausgeführt wird, wird eine aktuelle Binär-Bitmap mit übertragen. Die Bitmap startet mit einem 0xAA Flag (kein ASCII!). Der Wert 0xAA taucht auch nicht im Datenstrom auf. Dann werden die Pixelwerte (8 Pixel = 1 Byte) als Binärwerte von links oben nach rechts unten (zeilenweise) übertragen. Pixel == 1 bedeutet, das Pixel liegt innerhalb der durch TC, oder TW definierten Grenzen. Die vertikale Auflösung wird beschränkt durch die Übertragungszeit der horizontalen Daten, so dass Zeilen abgeschnitten werden können. Bei voller Auflösung ergibt sich ein Binärbild von 80x48 Pixel. Das Binärbild wird beendet mit 2 folgenden 0xAA. Anschließend wird das Standard C oder M Datenpaket übertragen (berechnet mit niedrigerer Auflösung).

Beispiel: TC mit aktiven Line-Modus

:LM 1\r

:TC\r

(Binärdaten: AA XX XX XX &. XX XX XX AA AA) C 45 72 65 106 18 51

(Binärdaten: AA XX XX XX &. XX XX XX AA AA) C 46 72 65 106 18 52

Auswirkungen des Line-Modus auf den Befehl GM: Wenn bei aktive Line Modus der Befehl GM ausgeführt wird, wird zusätzlich ein binärer (kein ASCII) Mittelwert jeder Zeile übermittelt. Dieses Datenpaket startet mit einem 0xFE und endet mit 0xFD. Jedes Datenbyte zwischen diesen Werten repräsentiert jeweils eine Zeile. Wegen der seriellen Übertragungszeit ist ähnlich wie beim oben beschriebenen Binär-Bitmap die vertikale Auflösung halbiert. Bei 17fps und 115.200 bps wird jede zweite Zeile unterdrückt. Bei niedrigeren Bildraten (und 115.200 bps) werden keine Zeilen unterdrückt.

Beispiel: GM mit aktiven Line-Modus

:LM 1\r

:GM\r

(Binärdaten: FE XX XX XX & XX XX XX FD) M 45 56 34 10 15 8

(Binärdaten: FE XX XX XX & XX XX XX FD) M 45 56 34 10 15 8

**MM Modus\r**

Der Befehl steuert den Massenmittelpunkt-Modus welcher Schwerpunkt-Koordinaten zu den normalen Tracking-Daten hinzufügt. Ein Wert 0 schaltet den Modus aus, der Wert 1 (Standard) schaltet den Modus ein, Wert 2 schaltet ebenfalls ein und erlaubt die direkte Servo-Ansteuerung am Servo-Port. Hierbei wird versucht, die Kamera mit dem Servo auf den Farb-Schwerpunkt nachzuführen (siehe auch Demo Modus Beschreibung). Der Modus ist gut geeignet für die Einzelpunkt-Repräsentation des zu folgenden Objektes oder wenn das Farbrechteck wegen zu viel Hintergrundrauschens zu groß wird. Der Modus Wert beim Funktionsaufruf wird als Bitfeld interpretiert.

Modus = B3 B2 B1 B0

Bits

B0 – Massenmittelpunkt-Modus ein / Massenmittelpunkt-Modus aus

B1 – Massenmittelpunkt-Modus + Servoausgang ein / Servoausgang aus

B2 – Inverse Servodrehrichtung

B3 – gibt N-Datenpaket mit Servoposition zurück (siehe Seite 27)

Beispiel: Massenmittelpunkt-Modus aus

:MM 0\r

ACK

:TC\r

ACK

C 38 82 53 128 35 98

C 38 82 53 128 35 98

C 38 82 53 128 35 98

**NF Aktiv\r**

Dieser Befehl schaltet den Rauschfilter an oder aus. Mögliche Werte sind 1 (Standard) oder 0. Eine 1 aktiviert den Filter, 0 schaltet ihn aus. Bei aktivem Filter werden beim Verfolgen von Farben zwei benachbarte Pixel statt einem ausgewertet.

Beispiel: Ausschalten Rauschfilter

:NF 0\r

ACK

:

**PM Modus\r**

Dieser Befehl setzt die CMUcam in den Abfragemodus. Eine 1 aktiviert den Abfragemodus, eine 0 (Standard) schaltet ihn aus. Bei aktiven Abfragemodus wird nach einem Verarbeitungsbefehl immer nur ein Paket zurückgesendet. Das ist nützlich bei schnell zu ändernden Parametern oder bei langsamen Prozessoren, die der Bildrate und damit der resultierenden Paketrate nicht folgen können.

Beispiel: Abfragemodus

:PM 1\r

ACK

:TC 50 20 90 130 70 255\r

ACK

C 38 82 53 128 35 98

:

**RM bit\_flags\r**

Der Befehl setzt die CMUcam in den Binärtransfermodus. Der Übergabeparameter `bit_flags` wird bitweise ausgewertet, wenn alle Bits gelöscht sind (Standard), findet die Übertragung in beiden Richtungen mit lesbaren ASCII-Zeichen statt. Bei gesetztem Bit 0 werden die von der Kamera gesendeten Bytes als Binärdaten gesendet. Das Format des Datenpaketes ändert sich, es gibt keine Leerzeichen oder lesbare ASCII-Zeichen mehr. Stattdessen wird als Startbyte ein `0xFF` Byte gesendet, gefolgt von dem beschreibenden Buchstaben des Paketes (z.B. C für das Farb-Datenpaket). Danach folgt das Paket ohne abschließendes `\r`. Da es kein Endebyte gibt, muss die Synchronisation über die `0xFF` Startbytes erfolgen. Alle `0xFF` Bytes im Datenpaket werden auf `0xFE` abgeändert, um Verwechslungen zu vermeiden.

Bei gesetztem Bit 1 werden die `ACK\r` und `NCK\r` Bestätigungen unterdrückt.

Wenn Bit 3 gesetzt ist, müssen die Befehle an die CMUcam ebenfalls im Binärmodus gesendet werden. In dieser Betriebsart wird nach den 2 Befehlsbytes ein weiteres Byte gesendet, indem die Anzahl der folgenden Argumente angegeben werden. (z.B. DF gefolgt von einem Nullbyte für keine weiteren Argumente). Das abschließende `\r` entfällt.

`bit_flags = B2 B1 B0`

Bits

B0 – Daten von der Kamera zum Hostprozessor sind Binärdaten

B1 – `ACK\r` und `NCK\r` Bestätigungen werden unterdrückt

B2 – Daten zur Kamera sind Binärdaten

Beispiel: Neues Paket für TC (Track Color) mit Binärdaten von der Kamera (Achtung, die Binärdaten können beim Terminalprogramm zu temporären Darstellungsproblemen wegen undefinierten Zeichen führen.)

```
:RM 1\r
```

```
ACK
```

```
:TC 50 20 90 130 70 255\r
```

```
ACK
```

```
C>%k(ai Ck$&,.L
```

**RS\r**

Dieser Befehl setzt die CMUcam zurück. Das erste ausgegebene Zeichen ist `\r`.

Beispiel: Rücksetzen der Kamera

```
:rs\r
```

```
ACK
```

```
CMUcam v1.12
```

```
:
```

**S1 Wert\r**

Mit diesem Befehl lässt sich die Position des angeschlossenen Servos setzen. Wert 0 schaltet den Servo ab und hält den Ausgangspegel auf Low. Positionswerte 1-127 setzen den Servo auf die vorgegebene Position (nur beim Tracking oder holen von Mittelwerten). Werte  $\geq 128$  setzen den Ausgang auf High. Damit der Servo angesteuert wird, muss sich die Kamera im Verfolgungsbetrieb oder beim holen von Mittelwerten befinden. Die Werte 0 und 128 können nützlich sein, wenn man den Port als digitalen Ausgang nutzen möchte. Auch ist die Nutzung als Digitaleingang möglich (siehe I1 Befehl). Der MM-Befehl kann die automatische Ansteuerung (automatisches Tracking) des Servos erlauben oder unterdrücken.

**SM Wert\r**

Dieser Befehl steuert die Ausgabe von zusätzlichen Statistikinformationen beim Verfolgen von Farben. Normalerweise (Wert: 0, Standard) wird beim Verfolgen von Farben ein C oder M-Datenpaket zurückgeliefert. Wenn der Wert 1 beträgt, werden die Ausgaben bei den Tracking-Befehlen zwischen Farb- und Statistikpaketen abgewechselt. Jedes Statistikpaket wird erzeugt von einem viertel der umgrenzten Fläche des gefundenen Farbrechtecks. Wenn kein Farbrechteck gefunden wird, wird auch kein S-Paket generiert. Das kann nützlich sein für dynamisches, adaptives Farb-Tracking oder anderes Tracking, wo man Informationen über das Farbrechteck benötigt. Nach dem ersten Paket wird die Fenstergröße auf den vollen Wert (1, 1, 80, 143) zurückgestellt. Es ist zu beachten, dass nur die Hälfte der Farbpakete pro Zeiteinheit erzeugt werden.

Beispiel: Farbverfolgung mit SM

```
:SM 1\r
ACK
:TC 200 255 0 30 0 30\r
ACK
C 2 40 12 60 10 70 S 225 20 16 2 3 1
C 5 60 20 30 12 100 S 225 19 17 1 2 1
C 0 0 0 0 0
C 0 0 0 0 0
C 0 0 0 0 0
C 5 60 20 30 12 100 S 225 19 17 1 2 1
```

**SW [x y x2 y2]\r**

Dieser Befehl setzt das Fenster (Ausschnitt aus dem Bild) der Kamera. Akzeptiert werden kartesische Koordinaten der oberen linken Ecke und der unteren rechten Ecke. Der Bildursprung ist oben links. SW kann vor einem Bildverarbeitungsbefehl aufgerufen werden, um die Bildgröße einzuschränken. Ohne Argumente stellt der Befehl die Standardfenstergröße (Vollbild: 1, 1, 80, 143) ein. Versuchen Sie nicht SW 0 0 80 144, das ist außerhalb der Grenzen.

Beispiel: Einstellen eines Ausschnittsfensters in der Mitte des Kamerabildes

```
:SW 35 65 45 75\r
ACK
:
```

**TC [Rmin Rmax Gmin Gmax Bmin Bmax]r**

Der Befehl TC startet das Verfolgen einer Farbe. Übergeben werden die Minimum und Maximum RGB (CrYCb) Werte, von der Kamera daraufhin erzeugt werden M- oder C- oder N-Datenpakete (gesetzt über MM-Befehl). Das kleinere C-Paket übermittelt die Koordinaten eines Farbrechtecks (Rechteck um Pixel mit der definierten Farbe), die skalierte Anzahl der gefundenen Pixel (zurückgelieferter Wert ist  $(\text{Pixel}+4) / 8$ ), die im Farbrechteck liegen und ein Gütewert (Pixel / Fläche). Das Standard-M-Datenpaket schließt ebenfalls die Koordinaten des Massenmittelpunktes mit ein. Die Auflösung des berechneten Bildes ist 80 x 143 Pixel. Der X-Wertebereich geht von 1 bis 80 und der Y-Wertebereich von 1 bis 143. Ein Paket mit Nullen bedeutet, dass die definierte Farbe im Kamerabild nicht gefunden wurde. Der Güte- oder Vertrauenswert ist das Verhältnis von Anzahl der gefundenen Pixel und der Fläche des umgebenen Farbrechtecks, der Wertebereich liegt zwischen 0 und 255. Unter normalen Umständen stellt ein Wert größer 50 schon ein sehr hohes Vertrauen in das zu verfolgende Objekt dar. Ein Wert kleiner oder gleich 8 ist eher schlecht einzuschätzen. Ein Aufruf ohne Argumente verwendet die zuletzt benutzten Werte. Die Farbwerte der CMOS Kamera liegen zwischen 16 und 240 (siehe Seite 10).

Aufbau Typ M Datenpaket

M mx my x1 y1 x2 y2 Pixel Güte\r

Aufbau Typ C Datenpaket

C x1 y1 x2 y2 Pixel Güte\r

Beispiel: Farbverfolgung mit Standardparametern

:TC 130 255 0 0 30 30\r

ACK

M 50 80 38 82 53 128 35 98

M 52 81 38 82 53 128 35 98

**TW \r**

Nach Aufruf dieses Befehls wird die Kamera die Farbe verfolgen, die sich in der Mitte des Bildes vor dem Objektiv befindet. Nachdem die Farbe bestimmt wurde, wird die TC-Funktion aufgerufen.

Das kann nützlich sein, wenn ein Objekt vor der Kamera verfolgt werden soll. Weil intern der TC-Befehl benutzt wird, werden die gleichen C- und M-Datenpakete zurückgegeben. Das eingestellte Fenster (Ausschnitt) wird nur benutzt, um die Farbe zu ermitteln. Zum Verfolgen dieser Farbe wird dann die volle Auflösung (80x143 Pixel) benutzt. Die folgenden internen Schritte werden ausgeführt, wenn der TW -Befehl bearbeitet wird:

1. Einschränken des Sichtfeldes auf 1/4 der Größe (in jeder Dimension) des aktuellen Fensters. Positionierung des neuen Fensters auf Bildmitte. (SW 30 54 50 90)
2. Bestimmen des Farbmittelwertes. (GM)
3. Fenster auf volle Bildgröße. (SW 1 1 80 143)
4. Setzen der Min- und Maxwerte für jede Farbe auf Mittelwert +/- 30.

Beispiel: Benutzung von Track Window (Schwerpunktberechnung nicht aktiv):

:TW\r

ACK

S 240 50 40 12 7 8

C 2 40 12 60 10 70

C 2 41 12 61 11 70

## Beschreibung der Datenpakete

Alle Datenpakete, die von der Kamera gesendet werden, bestehen aus lesbaren ASCII Zeichen, mit Ausnahmen des F-Paketes (DF-Befehl) und der Prefix Pakete.

### ACK

Das ist die Standard-Bestätigung, dass ein syntaktisch richtiger Befehl empfangen wurde

### NCK

Das ist die Standard-Fehlermeldung, wenn Daten zur Kamera übertragen wurden, und ein Fehler auftrat. Bei der Arbeit mit Binärpaketen kann diese Meldung auch generiert werden, wenn kein Fehler aufgetreten ist.

### Aufbau Typ-C-Datenpaket

C x1 y1 x2 y2 Pixel Güte\r

Das Paket wird als Antwort auf den TC-Befehl gesendet.

x1 – Linke obere Ecke X-Wert

y1 – Linke obere Ecke Y-Wert

x2 – Rechte obere Ecke X-Wert

y2 – Rechte obere Ecke Y-Wert

Pixel – Anzahl der übereinstimmenden Pixel in der verfolgten Region, skaliert und begrenzt auf 255:  $(\text{Pixel} + 4) / 8$

Güte – Wert zur Aussage über den Grad der Übereinstimmung  $(\text{Anz. Pixel} / \text{Fläche}) * 256$  des gefundenen Farbrechtecks, begrenzt auf 255

### Aufbau Typ-F-Datenpaket:

1 2 r g b r g b ... r g b r g b 2 r g b r g b ... r g b r g b ...

1 – Neues Bild; 2 – Neue Spalte; 3 – Bildende

RGB (CrYCb) Wertebereich von 16 - 240

RGB (CrYCb) repräsentiert einen Wert von jeweils 2 Pixeln. Jedes Pixel hat einen Rot- und Blauanteil. 176 Spalten von R G B (Cr Y Cb) Paketen (ergibt 352 Pixel x 144 Zeilen).

Um das richtige Seitenverhältnis anzuzeigen, sollte jede Spalte verdoppelt werden, so dass sich ein endgültiges Bild von 352x144 ergibt. Das Paket fängt im Gegensatz zu den anderen Paketen nicht mit dem Kennbuchstaben an (kein F!) und enthält ausschließlich Binärdaten

### Aufbau Typ-M-Datenpaket

M mx my x1 y1 x2 y2 Pixel Güte\r

Das Paket wird als Antwort auf den TC-Befehl bei eingeschalteten Massenmittelpunkt-Modus gesendet.

mx – Massenmittelpunkt X-Wert

my – Massenmittelpunkt Y-Wert

x1 – Linke obere Ecke X-Wert

y1 – Linke obere Ecke Y-Wert

x2 – Rechte obere Ecke X-Wert

y2 – Rechte obere Ecke Y-Wert

Pixel – Anzahl Pixel in der verfolgten Region, skaliert und begrenzt auf 255:  $(\text{Pixel} + 4) / 8$

Güte – Grad der Übereinstimmung  $(\text{Anz. Pixel} / \text{Fläche}) * 256$ , begrenzt auf 255

**Aufbau Typ-N-Datenpaket**

N SPos mx my x1 y1 x2 y2 Pixel Güte\r

Das Paket ist identisch zum M Paket, mit einem hinzugefügten Wert für die Servo-Position.  
SPos – die aktuelle Servoposition

**Aufbau Typ-S-Datenpaket**

S Rmean Gmean Bmean Rdeviation Gdeviation Bdeviation\r

Das ist ein Paket mit Statistikinformationen über das Bild

Rmean – Mittelwert Rot R (oder Farbdifferenzsignal Cr ) im aktuellen Fenster

Gmean - Mittelwert Grün G (oder Helligkeit Y) im aktuellen Fenster

Bmean – Mittelwert Blau B (oder Farbdifferenzsignal Cb) im aktuellen Fenster

Rdeviation – die Abweichung\* R oder Cr im aktuellen Fenster

Gdeviation – die Abweichung\* G oder Y im aktuellen Fenster

Bdeviation – die Abweichung\* B oder Cb im aktuellen Fenster

\*Abweichung: der Mittelwert der absoluten Differenz zwischen den Pixeln und dem Mittelwert

**Aufbau Binärbitmap Line-Modus Prefix Paket**

Das Paket ist ausschließlich in Binärform. Es startet mit dem Byte 0xAA gefolgt von Binärwerten, die jeweils den Werten für 8 Pixel entsprechen, von links oben nach rechts unten. (Jedes Bit im Byte entspricht einem Pixel) Die Bitmap wird abgeschlossen mit einem zweifachen 0xAA. 0xAA kommt im normalen Datenstrom nicht vor, kann also als Kennung für Bildende verwendet werden. Nach dem Binärpaket schließt sich ein normales Trackingpaket an.

(Binärdaten: AA XX XX XX &. XX XX XX AA AA) C 45 72 65 106 18 51

(Binärdaten: AA XX XX XX &. XX XX XX AA AA) C 46 72 65 106 18 52

**Aufbau Mittelwert Line-Modus Prefix Paket**

Dieses Paket enthält die Mittelwerte für jede bearbeitete Zeile. Das Paket startet mit 0xFE und endet mit einem 0xFD. Jedes Byte dazwischen repräsentiert den Mittelwert der korrespondierenden Zeile. Wegen der seriellen Übertragungszeit wird die vertikale Auflösung halbiert. Nach dem Binärpaket schließt sich ein normales Trackingpaket an.

(Binärdaten: FE XX XX XX & XX XX XX FD) M 45 56 34 10 15 8

(Binärdaten: FE XX XX XX & XX XX XX FD) M 45 56 34 10 15 8

## Firmware Upgrades

Achtung: Das Upgrade ist normalerweise nicht notwendig, weil der SX28AC-Microcontroller vorprogrammiert ausgeliefert wird!

Mit Hilfe des SX-Key Download-Programms (Bestandteil der Entwicklungsumgebung SX-Key der Firma Parallax) kann das Programm im eingebauten Zustand in den Controller gebracht werden.

Aktuelle Versionen der Firmware für die CMUcam werden bereitgestellt auf:  
<http://www.cs.cmu.edu/~cmucam/downloads.html>

Durchführen des Firmware-Downloads

Voraussetzung ist eine installierte SX-Key (oder SX-Blitz) Programmierumgebung.

- Schritt 1: Anschalten der CMUcam Stromversorgung.
- Schritt 2: Sicherstellen, dass sich der Programmierschalter in der Programmierstellung befindet (zeigt in Richtung LEDs, siehe auch Ansicht des bestückten Moduls auf Seite 12)
- Schritt 3: SX-Key einstecken, Textseite zeigt vom Board weg (nach hinten, für mehr Details siehe auch die Schnittstellenbeschreibung)
- Schritt 4: SX-Key Software starten
- Schritt 5: Wählen Sie "Device" unter dem Menüpunkt "Run"
- Schritt 6: Klicken Sie auf "Load Hex" Button
- Schritt 7: Gehen Sie in das Verzeichnis, wo die CMUcam.hex Datei abgelegt ist und geben Sie den Namen in das Dialogfeld ein.
- Schritt 8: Drücken Sie "Open" um es zu laden.
- Schritt 9: Drücken Sie den "Program" Button, um den SX28AC zu programmieren
- Schritt 10: Nachdem die Programmierung fertig ist, schalten Sie das Board aus
- Schritt 11: Ziehen Sie den SX-Key ab und stellen den Programmierschalter in die normale Position (ggf. setzen Sie den Jumper)
- Schritt 12: Nach dem Wiedereinschalten des Boards wird die neue Firmware gestartet und sich über die serielle Schnittstelle melden:

CMUcam v1.xx

:

### **Download Fehlerhinweise**

Wenn folgende Fehlermeldungen erscheinen:

SX-Key not found

-Sichern Sie, dass der richtige COM-Port selektiert ist und die Kamera mit Strom versorgt wird

Vpp Generation Failed

-Sichern Sie, dass der Programmierschalter in der richtigen Position ist

Programming Failed

- Versuchen Sie es noch einmal, wenn der Fehler wieder auftritt, konsultieren Sie das SX-Key Manual

No response

-Sichern Sie, dass der Programmierschalter in der richtigen Position ist, schalten Sie das Board aus und wieder ein.

## Allgemeine Fehlerhinweise

Im Demo-Modus leuchtet die LED leuchtet kurz auf und dann stoppt alles  
Wenn Kamera und Servo aktiv sind, ist der Strombedarf größer, versuchen Sie eine leistungsfähigere Stromversorgung.

Beim Start der Java CMUcamGUI erscheint: Exception in thread "main"  
java.lang.NoClassDefFoundError: CMUcamGUI  
Evtl. sind Sie im falschen Verzeichnis. Tippen Sie *dir* am Kommandozeilen-Prompt und stellen Sie sicher, dass die Datei CMUcamGUI.class vorhanden ist.

Im Verzeichnis steht CMUcamGUI.java aber es fehlt die CMUcamGUI.class Datei.  
Sie sollten eine neue Version des Programms herunterladen, die .class Dateien sollten auf jeden Fall enthalten sein. Wenn Sie die Dateien wirklich noch einmal übersetzen möchten, tippen Sie `javac *.java`.

Ich erhalte die Meldung: 'java' ist entweder falsch geschrieben oder konnte nicht gefunden werden (oder ähnlich):  
Das bedeutet java ist nicht oder nicht richtig auf Ihrem Computer installiert. Probieren Sie eine nochmalige Installation und schauen Sie in die Dokumentation von Sun.

Die Spannungsanzeige-LED leuchtet nicht: Entweder hat das Board einen Fehler, oder die Stromversorgung bringt nicht genug Leistung. Überprüfen Sie die Stromversorgung und die Anschlüsse. Ziehen Sie alle Kabel von der CMUcam ab, bis auf die Stromversorgung, und probieren Sie noch einmal.

Ich erhalte Müll von der Kamera  
Schalten Sie die Kamera für min. 10 Sekunden aus. Dann probieren Sie noch einmal.

Ich bekomme Wellenlinien oder ein gestörtes schwarz/weiß Bild beim holen des Bildes (DF)  
Dahinter steckt meist eine Störung in der Stromversorgung. Sichern Sie, dass die Spannung hoch genug ist und frei von Störungen. Ein Test der Kamera an frischen Batterien ist eine gute Möglichkeit, das zu testen.

Mein Prozessor kann nicht mit der seriellen Geschwindigkeit mithalten  
Setzen Sie die Kamera in den Abfrage-Modus und fügen Sie Verzögerungen ein. Siehe Seiten 19 und 22 für PM und DM Details.

Ich bekomme keine seriellen Daten  
Sichern Sie, dass das Kabel mit der Kamera richtig verbunden ist, drehen Sie es testweise herum.

Warum gibt der SW-Befehl ständig ein NCK zurück?  
Beachten Sie die SW 1 1 80 143 Grenzen.

Ich sehe die CMUcam Start-Message, aber dann passiert nichts weiter.  
Überprüfen Sie das Schnittstellenkabel.

## Bauteilliste

Hierbei handelt es sich um die Originalliste der CMU

Anz.	Beschreibung	Teilenummer (Digi-key)	Bezeichnung im Schaltplan	Bauteil-Preis (\$)
1	75 MIPS Uvicom SX28	SX28AC/DP-ND	IC1	5,18
1	OV6620 Eval Board *	BB048 *		57,95
1	Maxim 232 Level Shifter	MAX232CPE-ND	IC3	3,31
1	SG-531 75.00 MHz crystal osc	SE2911-ND	QG1	5,85
1	5 Volt Regulator	LM2940CT-5.0-ND	IC2	1,8
3	4.7 Kohm 1/4 Watt Resistor	4.7KQBK-ND	R2 R3 R4	0,06
2	220 ohm 1/4 Watt Resistor	220QBK-ND	R1	0,06
6	1.0 uF Capacitor	P2105-ND	C1 C2 C3 C4 C5 C10	0,42
2	SPST Switch	EG1847-ND	S1 S2	1,11
1	3mm red LED	67-1125-ND	LED3MM	0,16
1	3mm green LED	67-1127-ND	D1	0,16
2	0.1uF Capacitor	P4923-ND	C6 C9	0,55
3	100 uF Capacitor	P5138-ND	C7 C8 C11	0,28
2	14 pin IC socket (to form a 28 pin socket)	AE8914-ND	IC1	0,32
1	8 pin IC socket **	ED3308-ND	QG1	0,42
1	16 pin IC socket	ED3316-ND	IC3	0,83
1	Double row female header	929852-01-36	JP5	3,35
1	Single row male header	929647-09-36-ND	JP1x2 JP3 JP4 JP7	1,83
(1)	Polarized 2 pin terminal housing	WM2700-ND		0,25
(2)	Crimp terminals	WM2200-ND		0,13
(1)	Polarized 2 pin terminal header	WM2000-ND		0,25
(1)	Female serial ribbon cable head	AFS09G-ND		6,14
(1)	Serial ribbon cable socket connector	ASC10G-ND		1,11

\* Bestellbezeichnung BB048 bei <http://www.electronics123.com>. Fragen Sie nach einem Evaluations-Board mit dem OV6620 von Omnivision und einer Optik mit IR-Filter.

\*\*die 4 inneren Kontakte müssen entfernt werden

( ) Teile, die das anschließen der CMUcam erleichtern, die aber für die Funktion nicht notwendig sind

# CMUcam Schaltung

