WHITE PAPER

Wie erweitert man ein Embedded System mit einem Kameramodul?

Es gibt viele Gründe dafür, sein Embedded System „mit Augen auszustatten“ und somit in ein Embedded Vision System zu verwandeln. Viele Produkte werden durch Bilderfassung ‚smart‘ oder lassen ganz neue Möglichkeiten zu. Das ist nicht nur ein Trend für Systeme, die autonomer arbeiten sollen, sondern oft auch ein sinnvoller Schritt, Systeme intelligenter zu machen bzw. künstliche Intelligez (KI) zu integrieren.

In der Praxis heißt das, ein oder mehrere Kameramodul(e) in das System zu integrieren, damit eine Bildaufnahme und -auswertung möglich ist. Die dafür nötigen Kameramodule sind als Komponenten in vielfältigen Ausprägungen verfügbar. Doch wie integriert man sie, welche Möglichkeiten bieten sich und welche Schwierigkeiten können auftreten, insbesondere angesichts der individuellen Anforderungen jedes Anwenders?

Auch wenn man ein komplettes System mit bereits integrierter Kamera, also eine fertige Lösung, einkaufen möchte oder auch entwickeln lässt, gibt es viele Dinge zu spezifizieren. Neben den rein optischen Eigenschaften gehören z.B. Softwareschnittstellen dazu, aber auch die angestrebte Performance der Bildverarbeitung, die wiederum unmittelbare Auswirkungen auf die Wahl der Hardware hat.

In diesem White Paper möchten wir Entwicklern sowohl einen kurzen Überblick über relevante Aspekte als auch Empfehlungen geben, über welche Gesichtspunkte man sich Gedanken machen sollte, um eine geeignete, aber auch schlanke und kostengünstige Lösung zu erhalten.

Inhalt

Abbildung 1: Embedded Vision System bestehend aus Snapdragon 820 Processing Board und Kameramodul mit BCON for MIPI-Schnittstelle

1. Was ist ein Kameramodul?

1.1 Mehr als ein Bildsensor

Als Kernkomponente steht der Bildsensor, also das Stück Silizium, das Lichteinfall in elektronische Signale und schließlich Bildinformation umwandelt. In einem Kameramodul ist dieser Bildsensor bereits fest integriert und angebunden. Daneben enthält ein Kameramodul noch mehr, nämlich die nötige Elektronik zum Betreiben des Sensors, genauso wie ein Gehäuse / eine Halterung und ein Objektiv oder eine Objektivhalterung. Weitere Komponenten, die nicht unmittelbar auffallen, sind die Schnittstelle, die Treibersoftware und auch das Image Signal Processing (ISP), das hardwareseitig (HW) und/oder softwareseitig (SW) realisiert wird. Die letzten zwei Punkte werden im Folgenden genauer beschrieben:

1.2 Interface und Treiber

Um die Bildsignale im Embedded System verwenden zu können, müssen sie im Prozessor dieses Systems in geeigneter Form ankommen. Dazu ist eine HW-Anbindung nötig, d.h. ein Stecker, ggf. mit Kabel. Bei USB-Verbindungen ist diese Anbindung standardisiert, bei anderen Schnittstellen ist dem nicht so. Im Embedded Bereich ist eine LVDS-Datenübertragung oder eine MIPI CSI-2- Anbindung gängig. Weitere Informationen finden Sie im White Paper „Die MIPI CSI-2 Schnittstelle für Embedded Vision Anwendungen“. Die Interfaces unterschieden sich u.a. durch ihre Bandbreite und ihre maximalen Kabellängen. Um die Daten an eine geeignete SW-Schnittstelle weiterzugegeben, ist ebenfalls noch ein Treiber nötig. Lesen Sie hierzu Genaueres im Abschnitt „Softwareanbindung“

1.3 Image Signal Processing (ISP)

Was oft nicht beachtet wird: zu einer Kamera gehört auch eine Bilddaten(vor-)verarbeitung. Diese enthält z.B. bei Farbkameras die Farbwertberechnung aus den übertragenen Grün-, Rot- und Blauwerten (das sog. De-Bayering) für jeden Pixel. Aber auch Interpolationen und Korrekturberechnungen (z.B. für eine Rauschverminderung oder dem Eliminieren von Randeffekten) zählen dazu. Dazu befindet sich auf dem Kameraboard entweder ein passender Logikbaustein oder im Embedded System ein speziell ausgelegter Prozessor (der dann als Image Signal Processor bezeichnet wird) und durch entsprechende Treibersoftware unterstützt werden muss.

Einfache Kameramodule enthalten diese Vorverarbeitung nicht, so dass sie (weniger effizient) in der GPU oder CPU des Embedded Systems stattfinden muss. Das bindet dort ggf. nötige Ressourcen, so dass diese Komponenten ggf. leistungsstärker ausgelegt werden müssen.

2. Welche Eigenschaften muss das Kameramodul haben?

2.1 Auflösung und Pixelgröße

Die Auflösung beschreibt die Anzahl der Pixel des Bildes. Wie viele Pixel man benötigt, ist stark von der Anwendung abhängig und kann z.B. über die Geometrie berechnet werden. Um eine Gesicht (20 cm breit) auf einem Bild, das einen Bereich von 10 Meter erfasst, noch mit 40 Pixeln (horizontal) zu erfassen, benötige ich 2000 Pixel in der Horizontalen. Für ein quadratisches Bild wären das 4.000.000 Pixel, also 4 MP.

Wer jedoch denkt, je mehr Pixel, desto besser, der irrt leider. Mehr Pixel sind nicht immer vorteilhaft und führen auch nicht zwangsläufig dazu, dass man auch mehr Details aufnehmen kann. Das hat zwei Gründe: Zum einen muss jedes Pixel prozessiert werden Für eine schnelle, effiziente und energiesparende Bildberechnung sind möglichst wenig Pixel also immer besser. Der andere Grund hat eine physikalische Erklärung, die mit dem natürlichen Bildrauschen zusammenhängt. Das kennt man von Bildern, die man bei wenig Licht, z.B. Kerzenschein, aufnimmt. Es entsteht durch zufällige statistische Schwankungen der Menge des einfallenden Lichts, aber auch durch elektronische Effekte („Dunkelrauschen“). Bei wenig Licht macht sich diese Schwankung besonders bemerkbar:

Hat man wenig Licht, z.B. 50 Lichtteilchen, könnte man mit einer Schwankung von ±10 Lichtteilchen rechnen. Das sind 20 % und damit eine relativ große Schwankung. Das führt dazu, dass eine gleichhelle Fläche im Bild gar nicht mehr so gleichförmig aussieht. Bei viel Licht (z.B. 20000 Lichtteilchen) ist die Schwankung (z.B. ±145 ≈ 0,7 %) klein und das Bild „sauber“.

Bei hohem Bildrauschen sinkt aber der effektive Informationsgehalt des Bildes, auch wenn man viele Pixel hat. Tatsächlich zeigt das ein praktisches Beispiel: Bei einem Test war im Vergleich die Erkennungsrate von Gesichtern mit einer 8 MP-Auflösung schlechter als die einer 2 MP-Kamera. Auch wenn das zweite Bild weniger Pixel pro Gesicht enthielt, waren diese weniger verrauscht (besseres Signal-Rausch-Verhältnis) und damit „aussagekräftiger“. Der Erkennungsalgorithmus konnte damit jedenfalls mehr anfangen.

Daher sollte man im Idealfall, sofern möglich, für gute Lichtverhältnisse z.B. mit externen Lichtquellen sorgen, um das Bildrauschen zu minimieren. Außerdem sollten die Pixel auf dem Bildsensor nicht zu klein sein, damit möglichst viele Lichtteilchen auf einen Pixel landen und so das Rauschen gering bleibt. Das aber bedeutet, dass weniger Pixel auf die Sensorfläche passen, d.h. der Sensor eine geringere Anzahl an Pixeln aufweist. Die ideale Auflösung eines Sensors ist also immer ein Kompromiss aus Auflösung und Signalqualität bzw. Informationsgehalt.

Weitere Infos finden Sie auch in unserem White Paper „Wie beurteilt man Bildqualität?“

Abbildung 2: Bild mit wenig (links) und viel Bildrauschen (rechts)

2.2 Bildrate und Sensortyp

Entscheidend für die zeitliche Auflösung ist die Framerate oder Bildwiederholrate. Sie bezeichnet, wie viele Bilder pro Sekunde aufgenommen werden und kann bei Kameramodulen sehr unterschiedlich sein. Auch hier muss man wieder berechnen, welcher Wert für die jeweilige Anwendung erforderlich ist. Für einen Stream reichen i.d.R. 30 Frames/Sekunde. Ansonsten muss man sich überlegen, wie schnell sich die abgelichteten Objekte (oder die Kamera) bewegen.

Hier zeigt sich aber auch noch ein anderer Punkt, über den sich jeder Entwickler Gedanken machen sollte: Soll die Kamera einen Global - oder Rolling Shutter-Sensor haben? Ein Global Shutter nimmt alle Bildzeilen gleichzeitig auf, der Rolling Shutter nacheinander. Wenn sich das Objekt während der Bildaufnahme bewegt, zeigt sich im Bild der sog. Rolling Shutter-Effekt.

Bewegen sich die Objekte also, ist ein Sensor mit Global Shutter zunächst sinnvoller. Ein solcher Sensor ist aber in der Regel auch teurer als ein Rolling Shutter-Sensor. Jedoch ist er nicht immer nötig: Bei einer Gesichtserkennung kann das Gesicht im Bild eines Rolling Shutter Sensors zwar minimal schräg wirken, wenn sich die Person während der Aufnahme bewegt. Diese Verzerrung ist aber so gering, dass ein Gesichtserkennungsalgorithmus damit meist gleich hohe Erkennungsraten erreicht wie bei Verwendung eines Sensors mit Global Shutter. Hier wäre also ein Rolling Shutter ausreichend. Des Weiteren ist bei modernen Sensoren dieser Effekt auch nicht mehr so stark ausgeprägt, d.h. die Verzerrungen sind oftmals nur recht klein. Daher ist ein Global Shutter meist nur bei sehr schnell bewegten Objekten notwendig. Entwickler mit wenig Erfahrung sollten deshalb einfach testen, welcher Sensortyp für ihre Anwendung Sinn macht.

Weitere Infos zum Thema Global- und Rolling Shutter, finden Sie in unserem White Paper „Global Shutter, Rolling Shutter - Funktionsweise und Merkmale zweier Belichtungsverfahren“.

Abbildung 3: Der Rolling Shutter-Effekt. Bewegen sich die aufgenommenen Objekte, zeigen sich bei Sensoren mit Rolling Shutter teils merkwürdige Bewegungsartefakte.

3. Wie kann die Kamera in das Embedded System eingebunden werden?

3.1 Die Hardwareanbindung

Bis auf USB gibt es hardwareseitig keine Standardisierungen – weder für LVDS- noch MIPI-basierte Anbindungen. Eine USB-Anbindung ist dagegen einfacher; nachteilig ist jedoch, dass sie in der Regel nicht die günstigste Möglichkeit darstellt und auch vom Systemaufbau (inkl. der Datenübertragung) her nicht die schlankste Lösung verwendet.

Andere Anschlüsse sind jedoch immer herstellerspezifisch. Das heißt: Entweder es gibt passende Adapter vom Kamera- oder SQM-Hersteller, oder eine eigene Hardwareanpassung ist nötig. Diese kann dann über eine Adapterlösung (Kabel oder Platine) oder eine Anpassung des Carrier- oder Processing Boards geschehen.

3.2 Die Softwareanbindung

Ähnlich wie bei der Hardware ist es auch bei der Software: generische Treiber bietet nur USB in Form von USB3Vision. Für andere Interfaces bieten die Kamerahersteller in der Regel eigene, proprietäre Treiber an. Die Treiber knüpfen dann an herstellerspezifische SDKs an, die Softwareschnittstellen (APIs) für die gängigsten Programmiersprachen anbieten. Diese APIs dienen zum einen dazu, die Bilddaten von der Kamera zu übertragen, aber auch – und dieser Punkt wird oft übersehen – um Steuersignale zur Kamera zu transferieren und diese so zu steuern bzw. zu parametrisieren.
Bei einem LVDS- oder MIPI basierten Interface können die Treiber auch an die Video4Linux - Schnittstelle angebunden werden. Diese bietet wiederum eine Schnittstelle zu Programmiersprachen und auch zu diverser anderer Software. Video4Linux ist eine verbreitete Schnittstelle, die allerdings für viele speziellere Anwendungen nicht immer optimal geeignet ist, da sie in ihren Möglichkeiten beschränkt ist (z.B. in der Steuerung der Kamera).

3.3 Softwarebibliotheken

Der Zugriff auf die Bilddaten kann direkt im Programmcode erfolgen. Eine weitere Bildanalyse (z.B. eine Gesichtserkennung oder Kantendetektion) kann dann z.B. über fertige Softwarebibliotheken und -tools erfolgen. Solche Produkte gibt es als open source (z. B. OpenCV), aber auch als kommerzielle Software. Letztere ist häufig auf bestimmte Applikationen zugeschnitten oder bietet besondere Features.

4. Was sollte man sonst noch beachten?

4.1 An das Objektiv denken

Ein Kameramodul braucht ein Objektiv, um zu funktionieren. Weitere Informationen finden Sie im White Paper „Welche Objektive gibt es und wie wählen Sie das geeignete Objektiv für eine Kamera?“

Etliche Module haben ein fest eingebautes Objektiv, andere bieten eine Objektivhalterung. Dazu sollte man zwei Aspekte bedenken. Erstens: Größere Objektive lassen mehr Licht ein. Dies ist ein Vorteil für die Bildqualität (siehe Kasten oben). Und zweitens: Die Genauigkeit der Objektive lässt auch nur eine bestimmte maximale Auflösung zu. Billigere Objektive haben z.B. eine Auflösung von 5 MP, teurere von 20 MP. Wer also ein Kameramodul mit vielen Pixeln verwenden möchte, braucht konsequenterweise ein entsprechendes (ggf. teures) Objektiv. Dies ist ein weiterer Grund, warum hohe Auflösungen des Sensors nicht per se wünschenswert sind.

4.2 Das Triggern („auslösen“)

Wer nicht kontinuierlich filmt, sondern gezielt einzelne Bilder aufnimmt, muss dafür zum richtigen Zeitpunkt ein Triggersignal an die Kamera senden. Dies kann durch die SW passieren (per „Software-Trigger“). Doch nicht alle Schnittstellen erlauben dies, z.B. ist es über die Video-4Linux-Schnittstelle nicht möglich, aus dem Programmcode einen Trigger zu senden. Andere SDKs dagegen bieten diese Möglichkeit. Neben dem Software-Trigger unterstützen einige Kameramodule auch die Möglichkeit, über einen Steuereingang an der Kamera direkt durch ein externes Triggersignal („Hardware-Trigger“) den Bildeinzug auszulösen. Ein solches Signal kann z.B. von einer Lichtschranke oder einem Schalter kommen.

5. Fazit

Wer ein Embedded System mit Kameramodul entwickelt oder spezifiziert, sollte sich über einige Aspekte Gedanken machen, um eine klarere Vorstellung der Lösung zu bekommen und nicht mitten in der Entwicklung zu stagnieren.

Natürlich gibt es hier Experten, die genauere Antworten geben können und geeigneten Lösungen im Gespräch vorschlagen können. Wichtige Aspekte sind dabei, wie einfach die Integration eines Kameramoduls ist, wie es in der Applikation angewendet werden kann, welche Features es wirklich bietet und natürlich, wie gut die Bildqualität und damit der Informationsgehalt des Bildes tatsächlich ist.

Basler bietet als Experte für Vision-Lösungen neben den einzelnen Komponenten wie Kameramodul und Software auch die gesamte Bandbreite an Services an: Neben der Beratung zu geeigneten Komponenten und Unterstützung bei deren Integration in Embedded Systeme hilft Basler auch bei Anpassung oder Erstellung von System-Software bis hin zur Entwicklung von Komplettlösungen.

Autor

Dr. Thomas Rademacher
Product Manager

Dr. Thomas Rademacher ist seit 2015 als Produktmanager bei Basler tätig. In dieser Funktion ist er verantwortlich für die Basler dart Kamera Serie und für Embedded Vision Systeme.

Nach Abschluss seiner Promotion im Fach Physik an der Universität Göttingen arbeitete er zunächst im Produktmanagement in einem führenden Unternehmen für industrielle Messtechnik in der Halbleiterbranche. Dort lag sein Tätigkeitsschwerpunkt im Bereich der optischen Metrologie und automatisierter Bildverarbeitung und -analyse.

Kontakt

Dr. Thomas Rademacher – Product Market Manager - Factory & Traffic – Product Manager

Tel. +49 4102 463 487
Fax +49 4102 463 46487
E-Mail: thomas.rademacher@baslerweb.com

Basler AG
An der Strusbek 60-62
22926 Ahrensburg
Deutschland

Basler AG

Basler ist ein international führender Hersteller von hochwertigen Kameras und Kamerazubehör für Anwendungen in Fabrikautomation, Medizin, Verkehr und einer Vielzahl von weiteren Märkten. Das Produktportfolio umfasst Flächen- und Zeilenkameras in kompakten Gehäusegrößen, Kameramodule als Boardlevel-Varianten für Embedded Vision-Lösungen sowie 3D-Kameras. Abgerundet wird das Angebot durch unser bedienerfreundliches pylon SDK sowie ein breites Spektrum von teils eigens entwickeltem Zubehör, das optimal auf unsere Kameras abgestimmt ist. Basler verfügt über drei Jahrzehnte Erfahrung im Bereich der Computer Vision. Das Unternehmen beschäftigt rund 600 Mitarbeiter an seinem Hauptsitz in Ahrensburg sowie in Niederlassungen und Vertriebsbüros in Europa, Asien und Nordamerika.

Basler AG
Germany, Headquarters
Tel. +49 4102 463 500
Fax +49 4102 463 599
sales.europe@baslerweb.com
www.baslerweb.com

Basler, Inc.
USA
Tel. +1 610 280 0171
Fax +1 610 280 7608
sales.usa@baslerweb.com

Basler Asia Pte Ltd.
Singapore
Tel. +65 6367 1355
Fax +65 6367 1255
sales.asia@baslerweb.com

BASLER
the power of sight