HARDWARENAHE SOFTWARELÖSUNGEN FÜR MINIATURISIERTE STERNSENSOREN

W. Fischer, O. Balagurin, H. Kayal, H. Wojtkowiak
Julius-Maximilians-Universität Würzburg, Institut für Informatik
Am Hubland, 97074 Würzburg, Deutschland

Zusammenfassung

Zur genauen Lagebestimmung von Raumfahrtsystemen sind Sternsensoren meist unabdingbar. Da die Anforderungen an die Lagekenntnis auch im Kleinsatelliten- und Cubesatbereich immer weiter steigen, müssen vermehrt neue Möglichkeiten gefunden werden bestehende Konzepte an die Anforderungen dieser Satellitenklassen anzupassen oder neue zu entwickeln. Im folgenden Beitrag wird die Entwicklung eines neuartigen Sternerkennungsalgorithmus vorgestellt, der eine sehr viel schnellere und effizientere Lagebestimmung als herkömmlichen Verfahren ermöglicht. Hierdurch soll die präzise Lagebestimmung im Kleinsatellitenbereich weiter vorangebracht werden. Die Implementierung dieses Algorithmus auf dem neuen Sternsensor Nanostar wird ebenfalls beleuchtet. Dieser wird momentan an der Universität Würzburg entwickelt und soll bis 2016 vollständig qualifiziert sein.

1. STERNSENSOREN FÜR KLEINSATELLITEN UND CUBESATS

Eine genaue Kenntnis der Lage und die Möglichkeit ihrer präzisen Regelung ist für Raumfahrtsysteme, insbesondere solche mit optischen Nutzlasten beispielsweise zur Erdbeobachtung oder Astronomie, von existentieller Bedeutung. Da bisher keine anderen Sensoren verfügbar sind, die eine ebenso exakte Genauigkeit liefern wie Sternsensoren, werden diese bei den meisten Missionen zur Feinregelung genutzt. Aufgrund der beschränkten Ressourcen an Platz und Energie ist der Einsatz solcher Instrumente aber bis jetzt meist nur auf großen Satelliten möglich, während im Kleinsatellitenbereich häufig viel ungenauere Sonnen- oder Magnetfeldsensoren eingesetzt werden.

An der Universität Würzburg wurde deshalb im Zeitraum 2009 bis 2012 bereits ein erster Sternsensor, STELLA[1], entwickelt, welcher mit einer Gesamtmasse von 170 g und einer Leistungsaufnahme von durchschnittlich 250 mW auch Kleinsatelliten die Möglichkeit zur Lagebestimmung im Bogensekundenbereich gibt. Das System wurde im Rahmen mehrerer Testkampagnen sowohl funktional wie auch Thermal-Vakuum- und mechanisch qualifiziert [1] und wird voraussichtlich 2015 als Nutzlast im TechnoSat Satelliten der Technischen Universität Berlin im Weltraum verifiziert. [2]

Auf Basis der im STELLA Projekt gewonnenen Erfahrungen und dem Trend zu einer weiteren Miniaturisierung von Satellitenkomponenten folgend wird nun an der Universität Würzburg an einem weiteren Sternsensor geforscht, welcher mit einer noch geringeren Ressourcenauslastung auch Cubesats die Möglichkeit zur Lagebestimmung anhand der Sterne ermöglicht. Dieses Instrument, Nanostar, soll in etwa die Hälfte der Genauigkeit von STELLA erreichen, dafür aber auch nur die halbe Größe und eine weitaus höhere Updaterate haben. Der Abschluss der Entwicklung und Qualifizierung des Sensors wird für 2016 angestrebt.

BILD 1. Der Sternsensor STELLA

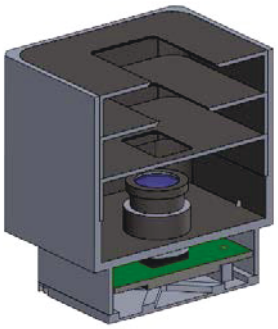

BILD 2. CAD-Model des Nanostar Sternsensors

[1] Das Projekt STELLA wurde von der Raumfahrt-Agentur des Deutschen Zentrums für Luft und Raumfahrt e. V. mit Mitteln des Bundesministeriums für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 50RM0901 gefördert.

Parameter	Wert
Genauigkeit	Pitch, Yaw: 0,01° Roll: 0.04°
Versorgungsspannung	3,3 V
Leistungsaufnahme	250 mW (Durchschnitt)
Abmessungen	98 x 46 x 58 mm³
Masse	170 g
Betriebstemperatur	-25° bis +55°C
Lebensdauer im LEO (Referenzorbit 500 km)	2 Jahre
Datenausgabeformat	Quaternionen
Ausgaberate	4 Hz
Schnittstelle	2 x CAN 2.0B 2 x UART

TAB 1. STELLA Parameter

Parameter	Wert
Genauigkeit	Pitch, Yaw: 0,02° Roll: 0.08°
Versorgungsspannung	5 V
Leistungsaufnahme	< 200 mW (Durchschnitt)
Abmessungen	55 x 50 x 50 mm³
Masse	100 g
Betriebstemperatur	-25° bis +55°C
Lebensdauer im LEO (Referenzorbit 500 km)	2 Jahre
Datenausgabeformat	Quaternionen
Ausgaberate	8 Hz
Schnittstelle	CAN 2.0B, SPI, UART, I²C (konfigurierbar)

TAB 2. Parameterzielsetzung für Nanostar

2. DATENVERARBEITUNGSKONZEPTE FÜR STERNSENSOREN

Die Lagebestimmung anhand von Sternbildern erfolgt bei allen gängigen Sternsensoren in vier Schritten:

1) Im ersten Schritt wird eine Aufnahme der Sterne mit einem Bildsensor gemacht. Dieser muss in seiner Belichtungszeit entsprechend kalibriert sein, damit sich die Sterne bis zu einem bestimmten Magnitudenwert vom Hintergrund in ihrer Helligkeit abheben.
2) Die vom Bildsensor ausgegebenen Daten werden analysiert und somit die aufgenommenen Sterne erkannt. Gleichzeitig wird anhand der zum jeweiligen Stern gehörenden Pixel ein nach der Helligkeit gewichteter Schwerpunkt des Sterns ermittelt und ausgegeben. Auch Störobjekte wie der Mond oder der Erdhorizont werden durch ihre Dimensionen erkannt und vom Algorithmus ausgeschlossen.
3) Anhand der Abstände, Winkel und ggf. auch der Helligkeitswerte der erkannten Sterne wird die Sternkonstellation beschrieben und die einzelnen Sterne durch einen Datenbankabgleich identifiziert.
4) Im letzten Schritt wird die Lage des Sternsensors anhand der im Bild identifizierten Sterne und deren Koordinaten berechnet und ausgegeben.

Da jeder dieser Schritte von den Ausgabewerten des vorangegangenen Schritts abhängig ist, müssen sie für eine Aufnahme stets sequentiell ablaufen.

Ist die Lage des Sternsensors einmal bekannt und ist keine große Lageänderung im Zeitraum zwischen zwei Aufnahmen zu erwarten besteht die Möglichkeit nur noch die Teile des aufgenommenen Bildes zu untersuchen, von denen bekannt ist, dass sich dort Sterne befinden. In diesem meist als „Tracking“ bezeichneten Modus kann der Rechenaufwand, insbesondere in den Schritten zwei und drei, stark reduziert werden, da die ungefähren Koordination und die Identität der sich im Bild befindlichen Sterne bereits bekannt ist. [3]

Während die Ablaufplanung des Lagebestimmungsalgorithmus also auf unterschiedlichen Sensormodellen recht ähnlich aussieht können bei der Allokation des Algorithmus (der Zuweisung der einzelnen Schritte auf bestimmte Ressourcentypen) und bei dessen Bindung (der Zuweisung der einzelnen Schritte auf bestimmte Ressourceninstanzen) große Unterschiede auftreten. Im Folgenden sollen zwei verschiedene Ansätze vorgestellt und bewertet werden: zum einen der des STELLA Sensors, zum anderen der des sich in der Entwicklung befindlichen Nanostar Systems.

2.1. Das STELLA Datenverarbeitungskonzept

Das STELLA Datenverarbeitungskonzept wird bei Wojtkowiak et al. (2011) ausführlich beschrieben. Im Folgenden sollen die Kernpunkte nochmals im Hinblick auf die Umsetzung des Lagebestimmungsalgorithmus und die Unterschiede zum Nanostar Sensor vorgestellt werden. Das Hardwarekonzept von STELLA ist in BILD 3 dargestellt.

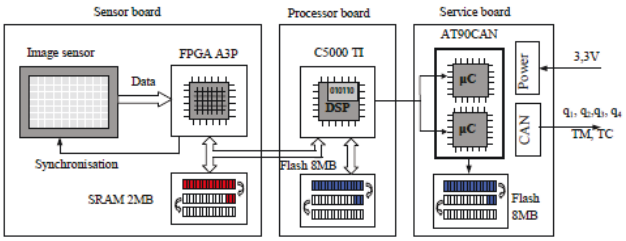

BILD 3. STELLA Hardwarekonzept

Die Datenverarbeitung ist bei STELLA auf drei unterschiedliche Platinen aufgeteilt: Auf der Sensorplatine befindet sich der Bildsensor, ein FPGA sowie ein SRAM-Baustein. Der FPGA ist für das Auslösen des Bildsensors sowie den Datenempfang von diesem und das direkte Speichern der gesamten Bilddaten auf dem SRAM zuständig. Die zweite Platine beherbergt einen DSP sowie einen Flashspeicherbaustein. Der DSP liest die im SRAM der Sensorplatine gespeicherten Bilddaten, erkennt die aufgenommen Sterne, berechnet deren Mittelpunkte und identifiziert sie durch einen Abgleich mit dem im Flashspeicher abgelegten Sternenkatalog. Anhand der identifizierten Sterne berechnet der DSP daraufhin die Lage von STELLA. Diese Lage wird über einen internen I²C-Bus an die Serviceplatine weitergegeben, auf der sich ein redundantes Mikrocontrollerpaar befindet. Dieses verpackt die berechnete Lage in für den

Satellitenbus passende Frames und verschickt sie über die ebenfalls auf der Serviceplatine befindlichen CAN-Transceiver. Werden diese einzelnen Schritte nun in einem Ablaufplan dargestellt, den genannten Ressourcen zugewiesen und mit den im Betrieb ermittelten Laufzeiten versehen ergibt sich folgendes Bild:

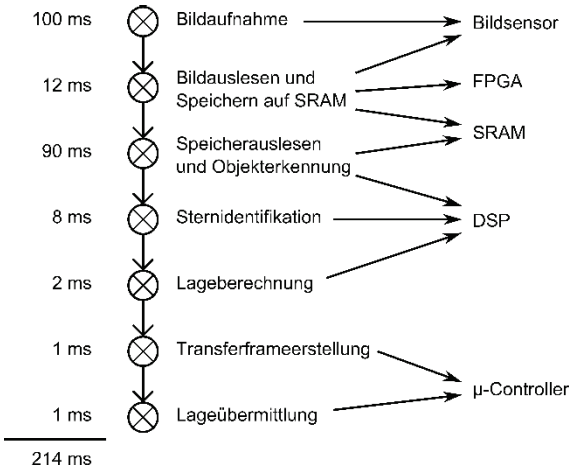

BILD 4. STELLA Lagebestimmungsablaufplan

Hierbei ist anzumerken, dass es sich bei den Zeitangaben zur Objekterkennung, Sternidentifikation, Lageberechnung, Transferframeerstellung und Lageübermittlung um gemessene Durchschnittswerte handelt, da diese Prozesse von verschiedenen Einflussfaktoren, wie der Anzahl der auf dem Bild vorhandenen Sterne, abhängen. Eine Überlappung mehrerer Prozesse ist deshalb nicht möglich, da es sonst zu Konflikten beim Zugriff auf den SRAM kommen kann.

Aus dem Ablaufplan ist erkennbar, dass durchschnittlich etwa 90 Prozent der gesamten Ablaufzeit für die Bildaufnahme und Objekterkennung aufgebracht werden müssen. Das meiste zeitliche Einsparpotential liegt somit in diesen beiden Prozessen. Das Verringern der Belichtungsdauer und damit der Bildaufnahmezeit bringt in der Regel eine geringere Empfindlichkeit bzw. ein erhöhtes Rauschen, besonders in der durch eine erhöhte Strahlung belasteten Weltraumumgebung, mit sich. Die Aufnahmezeit soll in diesem Kontext deshalb nicht weiter betrachtet werden. Die weitaus einfacheren Möglichkeiten zur zeitlichen Optimierung des Ablaufs liegen in der Objekterkennung. Da dieser Schritt mehr als vierzig Prozent der Gesamtzeit in Anspruch nimmt, ist auch das Einsparpotential im Vergleich zu allen anderen Datenverarbeitungsschritten am größten.

2.2. Das Nanostar Datenverarbeitungskonzept

Um den Ablauf zu beschleunigen, die mögliche Updaterate zu erhöhen und die Anzahl der für die Datenverarbeitung nötigen Komponenten zu minimieren wurde für Nanostar ein neues Hardwarekonzept entwickelt. Bei diesem läuft der Schritt der Objekt- und Sternerkennung nicht erst in der Prozessoreinheit sondern direkt beim Auslesen des Bildsensors in einer freiprogrammierbaren Logikeinheit, wie beispielsweise einem FPGA, ab. Dadurch entsteht zwischen dem Sensorauslesen und der Schwerpunktbestimmung keine weitere Verzögerung, die Prozessoreinheit erhält noch während der Bildsensor die Bilddaten ausgibt bereits die Schwerpunkte der im Bild vorhandenen Sterne übermittelt. Für sie bleiben somit nur noch die Aufgaben der Sternidentifikation, Lageberechnung, Frameerstellung und -übermittlung. Da in dieser Konfiguration das Auslesen von Bildern aus der Sternenkamera nicht möglich ist, weil diese nirgends zwischengespeichert werden können, kann noch ein zusätzlicher Speicher vorgesehen werden. In diesem Fall überspielt die freiprogrammierbare Logik, neben der Sternerkennung, das Bild noch in den Speicher und ermöglicht dem Mikrocontroller das Auslesen der Daten. Diese Konfiguration ist in BILD 5 dargestellt.

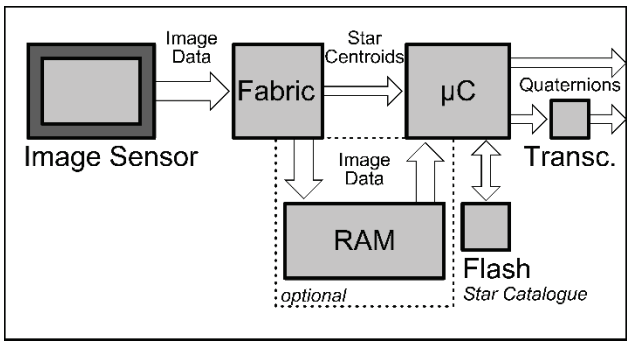

BILD 5. Nanostar Hardwarekonzept

3. ALGORITHMEN ZUR STERNERKENNUNG

Zur Realisierung des Nanostarhardwarekonzepts ist ein effizienter und auf Logikebene programmierbarer Algorithmus notwendig, der aus den vom Bildsensor empfangenen Daten Sterne extrahiert und in Form von Metadatensätzen an die Recheneinheit weitergibt. Die Weitergabe von Bilddaten ist nicht notwendig, auch nicht in Ausschnitten, da diese für die weitere Verarbeitung nicht gebraucht werden.

3.1. Prozessorbasierte Sternerkennung

Für die Konzipierung des logikbasierten Algorithmus ist es wichtig zu verstehen, wie für gewöhnlich die Sternerkennung von im Speicher abgelegten Bilddaten durch eine Prozessoreinheit abläuft. Dies soll im folgenden Abschnitt noch einmal kurz erläutert werden.

Sind die Bilddaten einer Aufnahme im Speicher abgelegt werden sie Zeile für Zeile und Pixel für Pixel betrachtet. Liegt der Helligkeitswert eines Pixels über einem vom Designer festgelegtem Schwellenwert wird diese Pixel maskiert. Der Maskierungswert ist abhängig von Helligkeitswerten der umliegenden Pixel. Wurde hier bereits ein anderes Pixel maskiert dann erhält das neue Pixel den gleichen Maskierungswert wie das andere helle Pixel. Ist dies nicht der Fall wird ein neuer Maskierungswert vergeben. Die so erzeugte Bildmaske kann zum Abschluss wie in BILD 6 dargestellt aussehen.

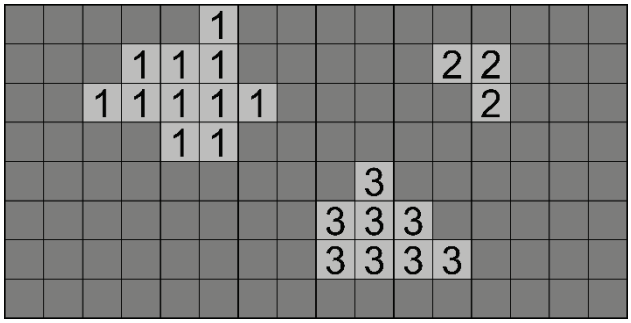

BILD 6. Maskierte Sternenaufnahme

Nun werden nur noch die jeweils zu einem Maskierungswert gehörenden Pixel betrachtet. Zur Ermittlung des Schwerpunkts werden mit den x- bzw. y-Koordinaten multiplizierten Helligkeitswerte einer Maske aufaddiert und

durch die Gesamthelligkeit der Maske wie in den Gleichung 1 und 2 angegeben dividiert. Der so ermittelte Schwerpunkt hat eine Genauigkeit im Subpixelbereich.

$$x = \frac{\sum_{x=0}^{image_width} \sum_{y=0}^{image_hight} x \cdot image(x,y)}{\sum_{x=0}^{image_width} \sum_{y=0}^{image_hight} image(x,y)} \quad (1)$$

$$y = \frac{\sum_{x=0}^{image_width} \sum_{y=0}^{image_hight} y \cdot image(x,y)}{\sum_{x=0}^{image_width} \sum_{y=0}^{image_hight} image(x,y)} \quad (2)$$

Eine detaillierte Erklärung dieses Algorithmus findet sich beispielsweise bei Liebe (2002).

3.2. Logikbasierte Sternerkennung

Ein erster auf Logikebene implementierbarer Algorithmus zur Sternerkennung wurde 2009 von Jiang et al. an der Beijing University of Aeronautics & Astronautics entwickelt und wird in [6] vorgestellt. Da dieser einen sehr hohen Bedarf an für die Logik verfügbarem Speicher hat ist er für die Implementierung auf Nanostar ungeeignet. Hier bedarf es eines ressourcenschonenderen Algorithmus.

Das für Nanostar entwickelte Sternerkennungsverfahren empfängt als Eingangsdaten vom Bildsensor die einzelnen Pixeldaten Zeile für Zeile von oben nach unten und Pixel für Pixel von links nach rechts. Da für Sternenkameras in der Regel monochrome Bildsensoren benutzt werden, muss nicht zwischen verschiedenfarbigen Pixeln unterschieden werden. Ein Pixel kommt standardmäßig als acht oder zwölf Bit breiter Datensatz auf parallelen Datensignalen und mit einem vom Bildsensor generierten Clock-Signal synchronisiert an. Der Sternerkennungsalgorithmus wertet diese Bilddaten nun so aus, dass keine einzelnen Pixelwerte zwischengespeichert werden müssen und der Schwerpunkt eines Sterns ausgegeben werden kann unmittelbar nachdem alle zum Objekt gehörenden Pixel empfangen wurden. Jeder Schwerpunkt wird hierbei als ein Parametersatz von vier Werte ausgegeben: der akkumulierte Helligkeitswert des Objekts, die Anzahl der zum Objekt gehörenden Pixel, sowie aufsummiert alle Helligkeitswerte multipliziert mit ihren jeweiligen x-Indizes der zum Stern gehörenden Pixel und aufsummiert alle Helligkeitswerte multipliziert mit ihren jeweiligen y-Indizes der zum Stern gehörenden Pixel.

Ob es sich bei dem erkannten Objekt um einen Stern handelt wird anhand der Anzahl der zum Objekt gehörenden Pixel festgestellt. Liegt diese bei unter drei handelt es sich mit großer Wahrscheinlichkeit um ein oder zwei Pixel die durch Rauschen den Schwellenwert überschritten haben. Das Objekt wird deshalb verworfen. Besteht ein Objekt aus zu vielen Pixeln ist dies ein Störobjekt wie etwa die Erde oder der Mond. Auch in diesem Fall wird das Objekt verworfen.

Die Schwerpunkte der ausgegebenen Sterne lassen sich anhand seiner drei akkumulierten Helligkeitswerte, nach den Gleichungen 1 und 2, wie bei der prozessorbasierten Variante bestimmen. Werden die mit den x-Indizes multiplizierten Helligkeiten durch die Gesamthelligkeit des Sterns geteilt ergibt sich die x-Koordinate des Schwerpunkts, werden die mit den y-Indizes multiplizierten Helligkeiten durch die Gesamthelligkeit des Sterns geteilt ergibt sich die y-Koordinate des Schwerpunkts.

Da das für Nanostar verwendete Verfahren momentan auf seine Patentierbarkeit geprüft wird ist eine detaillierte Erklärung an dieser Stelle nicht möglich.

Beim Vergleich dieses Algorithmus mit der prozessorbasierten Sternerkennung werden die Performanceunterschiede dennoch klar: Die logikbasierte Variante benötigt eine weit komplexere Auslese- und Objekterkennungslogik zur Pixelverarbeitung. Diese kann, wie jede andere auf eine Anwendung zugeschnittene Logik, üblicher Weise auch nur für die Sternerkennung genutzt werden, und ist für die restlichen Schritte des Lagebestimmungsalgorithmus unbrauchbar. Die Anzahl der gleichzeitig erkennbaren Sterne ist hier außerdem durch den Umfang der Objekterkennungslogik beschränkt, welche im Vorfeld definiert werden muss und abhängig von den zur Verfügung stehenden Ressourcen ist. Der Vorteil liegt aber auch auf der Hand: Während bei der prozessorbasierten Variante ein zusätzlicher, sehr zeitintensiver Arbeitsschritt für die Objekterkennung im Ablaufplan benötigt wird, erfolgt die Objekterkennung bei der logikbasierten Variante gleichzeitig mit dem Auslesen des Bildsensors. Für diesen Schritt wird keine zusätzliche Zeit benötigt.

4. NANOSTAR PLATTFORMIMPLEMENTIERUNG

Zum Aufbau des Nanostar Sternsensors wurde eine Hardwareplattform entworfen, welche die in BILD 5 dargestellten Elemente möglichst platz- und energiesparend zur Verfügung stellt und gleichzeitig die für die Lagebestimmung benötigte Rechenleistung mitbringt. Zur Kombination von freiprogrammierbarer Logik mit Mikroprozessoren werden momentan vermehrt System-on-Chip (SoC) eingesetzt. Ein solches kommt auch bei Nanostar zum Einsatz: Das SmartFusion2 des Herstellers Microsemi bringt neben einem leistungsstarken Cortex-M3 Mikroprozessor auch einen, je nach Modell in der Größe wählbaren, freiprogrammierbaren Logikanteil mit. Die beiden Teile des SoC werden über eine High-Speed AHB/APB-Schnittstelle verbunden, welche den Austausch großer Datenmengen in kürzester Zeit erlaubt. Ein entsprechendes Entwicklungsmodell mit Bildsensor, Optik und anderer Peripherie ist auf der Basis eines Development Boards bereits fertig aufgebaut und der Sternerkennungsalgorithmus auf der Plattform implementiert. Das Modell wird momentan am Sternensimulator der Universität Würzburg getestet, erste Ergebnisse versprechen bereits eine sehr gute Erkennungsrate.

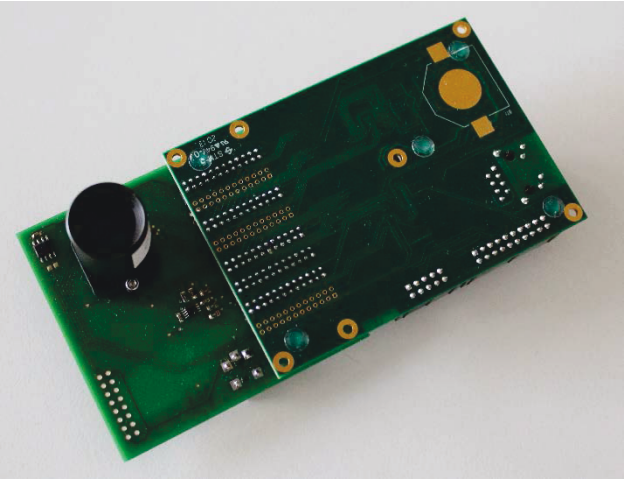

BILD 7. Nanostar Entwicklungsmodell

5. NÄCHSTE SCHRITTE

Nach Abschluss der Sternerkennungstests und der Portierung der übrigen Prozesse von der STELLA-Plattform auf das Nanostar-Entwicklungsmodell, ist der Entwurf eines funktionalen und geometrischen Prototyps geplant. Gleichzeitig soll die Plattform, insbesondere der Bildsensor und das SoC, auf ihre Beständigkeit gegenüber Weltraumstrahlung getestet werden. Der Abschluss der Entwicklung und Qualifizierung ist für 2016 geplant.

SCHRIFTTUM

[1] Balagurin, O.; Kayal, H.; Wojtkowiak, H.: Validation and Qualification of a CMOS Based Miniature Star Tracker for Small Satellites. Small Satellites Systems and Services Symposium, 2012

[2] Barschke, M.; Ballheimer, W.; Dornburg, L.; Noak, D.; Brieß, K.; Adirim, H.; Pilz, N.; Kayal, H.; Balagurin, O.; Wojtkowiak, H.; Nitzschke, C: TechnoSat - A Nanosatellite Mission for On-Orbit Technology Demonstration. 27th Annual AIAA/USU Conference on Small Satellites, 2013

[3] Alexander, J. W.; Chang, D. H.: Cassini Star Tracking and Identification Algorithms, Scene Simulation, and Testing. SPIE's 1996 International Symposium on Optical Science, Engineering, and Instrumentation. International Society for Optics and Photonics, 1996.

[4] Wojtkowiak, H; Balagurin, O; Kayal, H.: A Novel APS Star Tracker for Pico and Nano-Satellites. 62nd International Astronautical Congress, Cape Town, South Africa, 2011

[5] Liebe, C. C.: Accuracy performance of star trackers – a tutorial. IEEE Transactions on Aerospace and Electronic Systems (Volume 38, Issue 2), pages 587 – 599, April 2002.

[6] Jiang, J; Zhang, G.; Wei, X.; Li, X.: Rapid Star Tracking Algorithm for Star Sensors. IEEE Aerospace and Electronic Magazine (Volume 24, Issue 9), pages 23 – 33, September 2009