IBB Ingenieur-Büro Bergmann, Sonnenweg 3, 15537 Grünheide, info@ibbergmann.org

BENUTZERHANDBUCH

des Physikalischen Zufallsgenerators PRG620

Version 1.0

Autor: **Frank Bergmann**

Letzte Änderung: **12.11.2017 11:04**

1 Inhaltsverzeichnis

2 Copyright

IBB Ingenieurbüro Bergmann

Sonnenweg 3

D-15537 Grünheide

Haftung

Bei der Erarbeitung dieser Dokumentation wurde größter Wert auf die Vollständigkeit und Richtigkeit des Inhalts gelegt. Es kann dennoch keine Garantie für die Vollständigkeit und Richtigkeit übernommen werden.

Für Hinweise zu dieser Dokumentation sind wir dankbar.

Hotline

Die Hotline des Herstellers erreichen Sie unter +49(0)172 308 6554.

Warenzeichen

MS Windows ist eingetragenes Warenzeichen der Microsoft Corp.

3 Bedeutung von Zufallszahlen

Gute Zufallszahlen sind das Fundament vieler kryptographischer Verfahren und Protokolle. Es ist wichtig, dass die verwendeten Zufallszahlen nicht vorhersagbar sind. Solche Zufallszahlen zu erzeugen, fällt Computern naturgemäß schwer. Zahlreiche Meldungen kritisieren Lücken, Schwächen und Manipulationen bei der Erzeugung von Zufallszahlen für kryptografische Verfahren.

Bei allen Meldungen geht es nicht um spezielle, bedeutungslose Applikationen, sondern um millionenfach installierte Standardprogramme in professionellen Anwendungen. Vor allem dort, wo kontinuierlich viele Zufallszahlen benötigt werden (Netzwerke, Kommunikationssysteme), sind statistische Angriffe auf schwache Zufallsgeneratoren am erfolgreichsten.

Nach dem Kerckhoff-Prinzip (die Sicherheit soll nur auf der Geheimhaltung des Schlüssels beruhen, nicht auf der Geheimhaltung des kryptographischen Algorithmus) benötigt jede Art von Verschlüsselung eine geheime Komponente, die unter keinen Umständen vorhersagbar oder rekonstruierbar sein darf: der aus Zufallszahlen gebildete Schlüssel. Diese Zufallszahlen werden in den bekannten IT-Sicherheitsapplikationen aus Pseudozufallszahlen gebildet. Quelle der Generierung von Pseudozufall ist ein so genannter Seed (ein Startwert, bestehend aus Passwort, Timer-Register, Tastaturanschlägen, Mausbewegungen usw.), mit dem ein mathematisch-kryptografischer Algorithmus eine statistisch gut verteilte Zufallsfolge erzeugt. Aber die gesamte Sicherheit der per Pseudozufall erzeugten geheimen Schlüssel hängt *ausschließlich* von dieser Anfangsinitialisierung ab. Die Anfangsinitialisierung ist bei richtiger Wahl der Quelle der einzige wirklich zufällige Parameter, alles Weitere ist *deterministisch* und somit berechenbar. Eine schwache Anfangsinitialisierung (trivialer Seed) ist im statistischen Ergebnis nicht erkennbar, aber ein effizienter Angriffspunkt der Kryptoanalysis.

Auch professionelle Entwickler nutzen als Seed für Pseudozufall oftmals das Timerregister in der Annahme: wer will denn schon wissen, in welcher Sekunde das Register ausgelesen wurde. Für einen Angreifer kein Problem, denn ein Jahr hat ca. 32 Millionen Sekunden. Und um diese mit der totalen Probiermethode (brute force) durchzutesten, benötigt man nur eine durchschnittliche Rechenleistung. Wird der gleiche Seed mehrfach verwendet, so entstehen schlüsselgleiche Geheimtexte. Ein sicherer Erfolg für die Kryptoanalysis.

Der Kryptoanalysis stehen heute wesentlich leistungsfähigere Werkzeuge zur Verfügung, so dass immer häufiger Meldungen zu kompromittierten schwachen Zufallsgeneratoren veröffentlicht werden. Dagegen haben die in IT-Sicherheitslösungen implementierten Pseudozufallsgeneratoren einen Stand erreicht, der eine neue Qualität der Zufallserzeugung erfordert.

Pseudozufall basierende Zufallsgeneratoren sammeln in diversen Entropiequellen in der Hoffnung, davon ausreichende Mengen zu sammeln. Zwar werden in diversen Chip-Sätzen (VIA, Transmeta, Renesas) und Security-Chipkarten derartige Zufallsgeneratoren angeboten, aber über Entropie und Qualität der Zufallsrohdaten werden keine oder nur unzureichende Angaben gemacht. Aus deren Chips konnten Sicherheitsforscher um Bernstein über 80 eindeutige RSA-Schlüssel auslesen, die gemeinsame Primfaktoren haben. Grund dafür war ein fehlerhafter Random Number Generator im AE45C1-Chip von Renesas, der nicht genügend Entropie erzeugt. Die veröffentlichten Daten zur Entropie des VIA C3 PadLock (7,64 Bit/Byte) zeigen für hohe Sicherheitsansprüche nicht ausreichende Werte. Zur Bit-Unabhängigkeit werden keine Informationen aufgeführt. Aber: nichts sagt mehr über die Eigenschaften eines physikalischen Zufallsgenerators aus, wie seine Rohdaten. Jede weitere Verarbeitung der Rohdaten verschleiert nur die wirklich statistischen Basiseigenschaften und kann vor allem die Entropie nicht weiter erhöhen.

4 Starke Entropie für kryptografische Applikationen

Mit dem PRG620 steht ein professioneller Zufallsgenerator der Klasse PTG.3 (hybrider Zufallsgenerator) für die permanente Generierung von kryptografisch sicherem Zufall zur Verfügung. Dieser Zufallsgenerator hat ein UART-Interface und **arbeitet ohne Kommando-Interface**. Nach PON werden kontinuierlich Zufallszahlen mit hoher Geschwindigkeit ausgegeben. Ein permanent im Hintergrund laufendes Sicherheitssystem garantiert, dass bei Ausfall oder Manipulation der Rauschquellen die Zufallsausgabe sofort eingestellt und solange weiter getestet wird, bis alle Qualitätskriterien wieder eingehalten werden.

Die generierten Zufallszahlen sind garantiert kryptografisch sicher und werden vorzugsweise zur Entropieerhöhung und -bereitstellung in Sicherheitsapplikationen verwendet. Unter Linux gibt es bekanntlich immer wieder Probleme mit der Bereitstellung von Zufall im Entropie-Pool. Besonders kritisch wird die Situation, wenn keine Eingabegeräte an einem Server zur Verfügung stehen.

Die Qualität der vom PRG620 erzeugten Zufallszahlen ist qualitativ unvergleichlich höher und sicherer, als jede andere Art der Zufallserzeugung. Zum Verifizieren dieser Aussage stehen diverse Analysen des Zufallsgenerators, auch unter erhöhter thermischer Belastung, zur Verfügung. Die hohe Ausgabegeschwindigkeit des PRG620 ermöglicht eine Füllung des Entropie-Pools (4096 Bit) in 100ms. Eine kryptografische Nachbearbeitung der ausgegebenen Zufallsdaten ist prinzipiell nicht erforderlich.

5 PRG620

Bei dem PRG620 handelt es sich um einen physikalischen Zufallsgenerator mit einer UART-Schnittstelle mit 3,3V-Pegel. Der PRG620 unterstützt die professionelle Generierung von kryptografisch sicheren Zufallszahlen der Klasse PTG.3. Vorzugsweise ist der PRG620 für den Einsatz auf den Raspberry-Pi-Boards vorgesehen, kann aber auch auf jeder anderen Pin-kompatiblen Plattform eingesetzt werden.

Abbildung: PRG620 auf einem Raspberry-Pi-Board (rot umrandet)

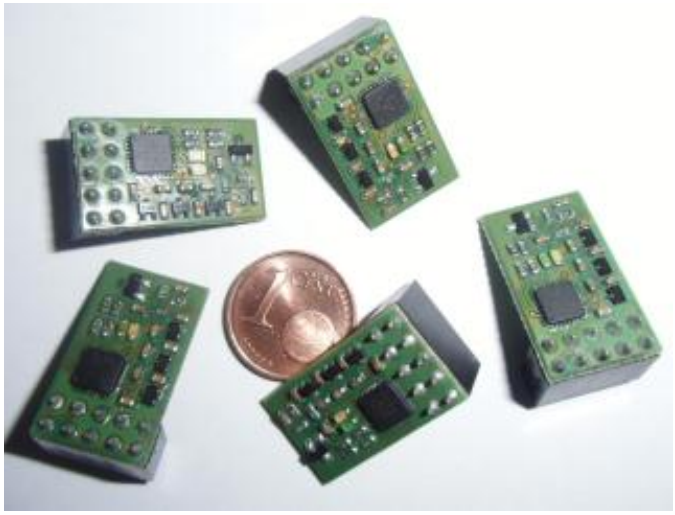


Abbildung: Exemplare des PRG620

6 Technische Daten

Abmessungen:	22*13*12,0 mm (mit Pfostensteckverbinder)
Versorgungsspannung:	3,3V (+/- 10%)
Stromaufnahme:	max. 7mA
Temperaturbereich:	funktionell und statistisch stabil von 0°C..+70°C
Schnittstellen:	UART-Interface, 115,2 Kbit/s
Qualitätssicherung:	Tot-Test zur Überwachung der Rauschquellen
	Online-Test zur statistischen Überwachung des Zufallssignals
	Abschaltung der Zufallsausgabe bei Ausfall der Rauschquelle oder unzureichender statistischer Qualität des digitalisierten Rauschsignals
Entropie:	>7,997 Bits/Byte (ermittelt aus Zufalls-Rohdaten nach Shannon)
0/1-Verhältnis:	garantiert im Bereich 0,499..0,501 (> 100 KByte)
Datengenerierung:	> 40 Kbit/s
Entropie-Pool füllen:	4096 Bit in 100ms

7 Verwendete GPIO des Raspberry-Pi

Der PRG620 nutzt folgende Pins des Raspberry-Boards:

- Pin 1: 3,3V
- Pin 6: GND
- Pin 8: TXD
- Pin 10: RXD

PRG620-Modul

(Draufsicht)

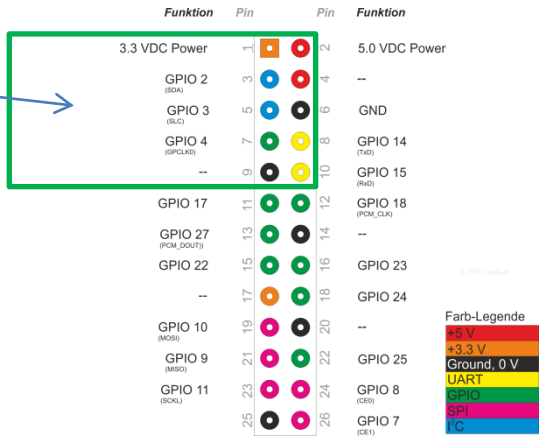


Abbildung: Anschlussbelegung des PRG620

8 Stochastische Modell

Ein stochastisches Modell unterstützt Gutachten über die Qualität und Zuverlässigkeit eines Zufallsgenerators. Es beschreibt die Entropiequelle, die Verarbeitung des digitalisierten Rauschsignals, die kryptografische Nachbereitung und die Sicherheitsfunktionen zur Überwachung der Signalverarbeitung.

Das stochastische Modell des PRG620 wird in folgendem Schema verdeutlicht:

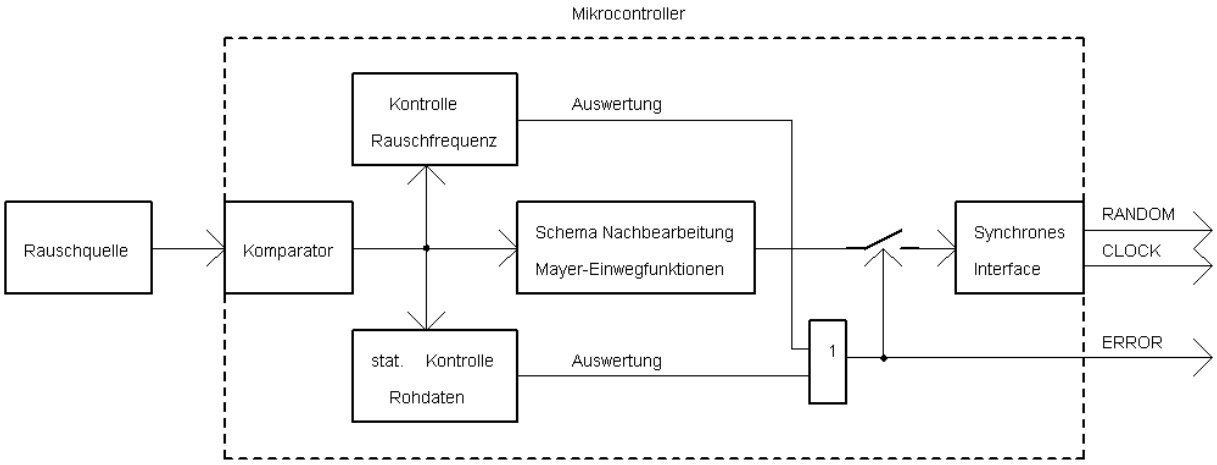


Abbildung: Schema des stochastischen Modells

9 Prinzip der Rauscherzeugung des PRG620

Rauschen ist ein physikalisches Phänomen und stellt eine Störgröße mit breitem unspezifischem Frequenzspektrum dar. Dieses Frequenzspektrum besteht aus der Überlagerung mehrerer Schwingungen oder Wellen mit unterschiedlicher Amplitude und Frequenz beziehungsweise Wellenlänge. Diese Eigenschaften wurden erstmalig 1918 durch Walter Schottky beschrieben. Später wurde das thermische Rauschen experimentell durch John Bertrand Johnson verifiziert. Eine Modellvorstellung der spektralen Leistungsdichte des thermischen Rauschens erfolgte durch Harry Nyquist

Das in diesem Zufallsgenerator verwendete 1/f-Rauschen bezeichnet ein Rauschen, dass mit steigender Frequenz abnimmt, die Amplitudenverteilung ist umgekehrt proportional zur Frequenz (~ 1/f). Die verwendete Rauschquelle ist ein Transistor mit ausgeprägtem Avalanche-Effekt. Rauschen entsteht hier durch den Lawineneffekt (Avalancheeffekt) in der pn-Sperrschicht des Halbleiterbauelements.

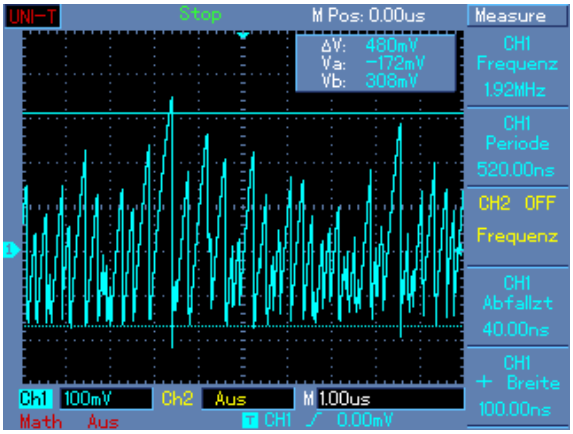


Abbildung: Rauschsignal des eingesetzten Transistors ohne Verstärkung

10 Generierung des Zufallssignals

Der eingesetzte Transistor kann reproduzierbar sehr hohe Rauschspannungen (>300mVss) bei einem breiten Rauschspektrum erzeugen, so dass keine Verstärkung erforderlich ist und der Signalpegel deutlich über dem Störpegel elektronischer Schaltungen liegt. Diese Rauschquelle muss nicht ausgemessen werden, da sie, technologisch bedingt, immer gleiche Rauschamplituden und ein gleiches Frequenzspektrum erzeugt. Ursache des Rauschens ist ein ausgeprägter Avalanche-Effekt.

Zur Digitalisierung des generierten Rauschsignals wird ein integrierter Analog-Komparator des eingesetzten Mikrocontrollers verwendet. Die Referenzspannung wird aus dem Gleichspannungsanteil des Rauschsignals erzeugt.

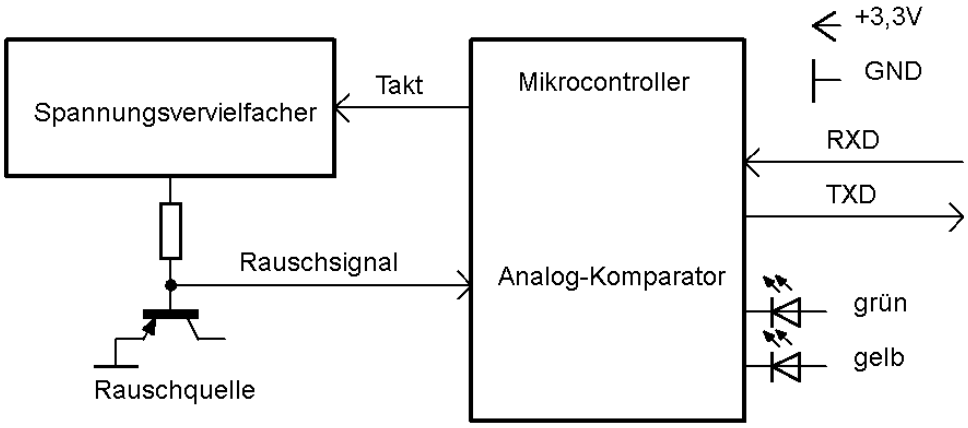


Abbildung: Blockschaltbild des PRG620

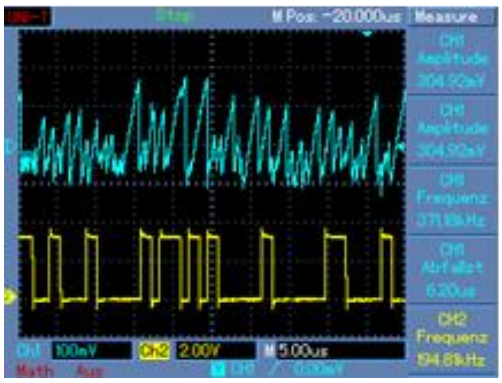


Abbildung: analoges und digitalisiertes Rauschsignal

11 Entropie

Die Entropie der Zufallsrohdaten ist die entscheidende Eigenschaft eines Zufallsgenerators und sollte so hoch als möglich sein. Die Generierung von Zufallsrohdaten ist per Kommando möglich. Zur Ermittlung der Entropie wurden jeweils 10Mbyte-Dateien generiert.

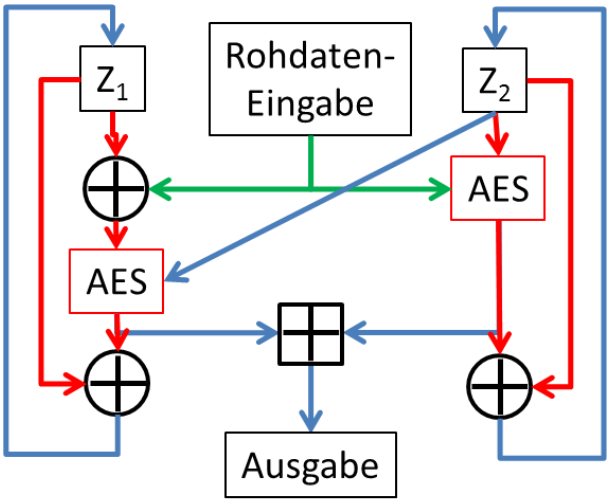
Folgende Entropiewerte (nach Shannon) der Rohdaten wurden für verschiedene Applikationen ermittelt:

PRG620	Mittelwert 0/1	Entropie der Rohdaten
Modul 1	0.50010359	7.99999975
Modul 2	0.49723013	7.99982290
Modul 3	0.49873226	7.99996290
Modul 4	0.49797257	7.99990512
Modul 5	0.49929176	7.99998842

12 Schema der kryptografischen Nachbearbeitung

Auf Grund der sehr hohen und robusten Entropie des physikalischen Zufallsgenerators wurden die Funktionen für die Zufallsgenerierung für die Klasse PTG.3 (hybrider Zufallsgenerator) ausgelegt. Für diese Klasse können beliebig lange Zufallsfolgen generiert werden.

In der Klasse PTG.3 erfolgt die Nachbearbeitung der generierten Rohdaten mit Mayer-Einwegfunktionen mit folgendem Schema (rechtes Bild):



Diese Nachbereitung nutzt 2 Prinzipien des Entropiesammelns: das Aufxorieren der Rohdaten für z1 und die Glättung durch Aufxorieren des mit zufälligem Schlüssel verschlüsselten Zustands für z2. Beide Komponenten nutzen die Mayer-Einwegfunktion AES(k,s) xor k, um den vorangegangenen inneren Zustand zu schützen. Durch die XOR-Summe der Zwischenwerte kann nicht auf den inneren Zustand geschlossen werden.

Dadurch werden auch bei einem kurzzeitigen Ausfall oder Manipulation der Rauschquellen statistisch gleichverteilte Zufallsdaten ausgegeben. Aus 16 Byte Rohdaten werden 16 Byte Zufallszahlen der Klasse PTG.3 generiert. Da Eingabe und Ausgabe jeweils 16 Byte (=128 Bit) betragen, wird explizit kein Pseudozufall generiert.

Da alle Zufallszahlen gleichverteilt sind, ist keine Synchronisation mit dem Empfänger erforderlich.

13 Sicherheitsfunktionen

13.1 Tot-Test

Es ist ein Kontrollsystem installiert, welches das digitalisierte Rauschsignal am Anschluss des Mikrocontrollers überwacht. Unmittelbar vor jeder Ausgabe der nach Schema nachbearbeiteten Zufallsrohdaten wird eine Funktion aufgerufen, die folgende Aufgabe hat:

- Es wird die Zeit ermittelt die erforderlich ist, um vier wechselnde Flanken des digitalisierten Rauschsignals zu erfassen
- Wird nach 50µs (entspricht 40 KHz) diese Bedingung nicht erfüllt, wird eine Fehlermeldung generiert, die zur sofortigen Einstellung aller Zufallsausgaben führt. Beide Leuchtdioden leuchten permanent.
- Dieser Zustand ist nur ist nur durch einen automatisch aktivierten „Intensiven Selbsttest" oder PON aufzulösen
- Typische Zeiten, um die Bedingung zu erfüllen, sind 5..8µs

13.2 Permanenter Online-Test

Im permanenten Halbbytetest (zyklisch im Abstand von 1 Sekunde) zur Kontrolle der statistischen Qualität der Zufallsrohdaten werden folgende drei Kriterien differenziert:

- Sind die Werte innerhalb der statistischen Vorgaben, wird kein Fehler generiert
- Sind die Werte außerhalb der statistischen Vorgaben, aber noch innerhalb von weitestgehend gleichverteilten Zufallsdaten, wird ein Fehlerzähler inkrementiert und beide Leuchtdioden blinken im Sekundentakt
- Ist einer der 16 Werte im Halbbytetest gleich Null, wird von einem Totalausfall ausgegangen und jede Zufallsausgabe blockiert. Angezeigt wird dieser Zustand durch das Leuchten der beiden LEDs. Dieser Zustand wird solange durchlaufen, bis alle Parameter wieder eingehalten werden.

14 Bedeutung der Leuchtdioden

Die auf der Platine befindlichen Leuchtdioden reflektieren die jeweils ablaufenden Funktionen und Zustände des PRG620. Das Blinken der Leuchtdioden erfolgt immer im Sekundentakt. Synchron zur Änderung des Blinkens (ein→aus und aus→ein) wird der permanente Online-Test gestartet.

LED grün	LED gelb	Zustand
Blinkt	Ein	Selbsttest ok, Zufallsdaten werden ausgegeben
Ein	Ein	Hardwarefehler im Online-Test festgestellt

15 Statistische Qualität

Dieser physikalische Zufallsgenerator generiert kontinuierlich Zufallsbits in herausragender statistischer Qualität. Eine Qualitätsaussage zu einem solchen Produkt ist aber nicht durch einen einzelnen statistischen Test möglich. In Zusammenarbeit mit Mathematikern und Kryptologen hat sich die Summe aus folgenden statistischen Tests für eine sichere Qualitätsaussage eines physikalischen Zufallsgenerators bewährt:

- Statistische Forderungen der AIS31-Dokumente für Rohdaten (keine digitale Nachbearbeitung, denn nichts beschreibt besser die Eigenschaften eines physikalischen Zufallsgenerators, als seine Rohdaten!)
- NIST-Test-Suite (Summe verschiedener Test der USA-Sicherheitsbehörde)
- Diehard-Test-Suite nach George Marsaglia
- Proprietärer statistischer Basistest

Zur Evaluierung wurden umfangreiche statistische Tests durchgeführt. So wurden die ausgewählten Tests auf mehrere erzeugte Bitfolgen angewendet. Keine dieser Testfolgen konnte Unterschiede zu einem idealen Zufallszahlengenerator aufzeigen. Darüber hinaus wurden Testergebnisse mit Untersuchungen von kryptographisch starken Pseudo-Zufallszahlengeneratoren, wie dem DES-, AES- und SH1-Generator verglichen. Die Vergleiche zeigten keine signifikanten Unterschiede zu den Testergebnissen dieser deterministischen Generatoren.

16 Anwendungen

Auf Grund der sehr hohen Entropie des physikalischen Zufallsgenerators wurden die Funktion für die Klasse PTG.3 (hybrider Zufallsgenerator) ausgelegt. Bezugnehmend auf die AIS31-Dokumente (Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren: www.bsi.bund.de/zertifiz/zert/interpr/ais31e.pdf) ist ein permanenter Online-Test und ein Tot-Test (Ausfall der Rauschquelle) der Zufallsrohdaten integriert. Fällt die Rauschquelle aus oder werden statistische Grenzen überschritten, wird die Ausgabe von Zufallsdaten so lange blockiert, bis alle Parameter wieder im Toleranzbereich liegen.

Der stetig steigende Bedarf an Zufallszahlen in vielen Bereichen von Wissenschaft und Technik erfordert eine zuverlässige und stabile Generierung von Zufallszahlen mit physikalischem Zufall und sicherer Gewährleistung aller erforderlichen statistischen und funktionellen Normen. Damit sind vor allem Entwickler von Sicherheitsapplikationen in der Lage, Wirksamkeit und Widerstand gegen Angriffe besser zu kalkulieren. Besonders hohe Ansprüche an die Statistik von Zufallszahlen werden für kryptografisch sichere Zufallszahlen gestellt.

Exemplarische Beispiele für den Einsatz des PRG620:

- Implementierung in proprietäre Applikationen der IT-Sicherheit
- einfachere Administration und sichere Verschlüsselung bei drahtloser Datenübertragung: WLAN, Bluetooth, GSM, ZiggBee, Industriedatenfunk
- für die Generierung kryptografisch sicherer Parameter auf Bords wie:
    - Raspberry-Pi
    - Odroid-C1
    - Banana-Pi

17 Einsatzumgebung

Der PRG620 ist für den permanenten Einsatz in beliebigen Applikationen entwickelt und getestet worden. Auch bei erhöhtem Industriestandard (0°C bis +70°C) bleiben die Entropiewerte sehr hoch und unterschreiten die Vorgaben aus den AIS31-Dokumenten nicht. Statistische Analysen, auch für diese Temperaturbereiche, befinden sich auf der Internetseite des Autors.

17 Literatur

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012

[5] AIS20: Functionality classes and evaluation methodology for deterministic random number generators, Version 2.1, 02.12.2011, Bundesamt für Sicherheit in der Informationstechnik

[6] AIS31: Functionality classes and evaluation methodology for true (physical) random number generators, Version 2.1, 02.12.2011, Bundesamt für Sicherheit in der Informationstechnik

[7] Killmann, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 2.0, September 18, 2011

[8] Evaluation of random number generators, Version 0.8, Bundesamt für Sicherheit in der Informationstechnik

[9] https://de.wikipedia.org/wiki//dev/random

[10] **Dokumentation und Analyse des Linux ... - BSI**