

BENUTZERHANDBUCH

des Physikalischen Zufallsgenerators PRG320

Version 1.1

Autor: Frank Bergmann
Letzte Änderung: 07.08.2017 11:12

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	2
2	Copyright	3
3	Bedeutung von Zufallszahlen	4
4	Starke Entropie für kryptografische Applikationen	4
5	PRG320 in verschiedenen Bauformen.....	5
6	Technische Daten	7
7	Prinzip der Rauscherzeugung	7
8	Generierung des Zufallssignals	8
9	Entropie	9
10	Generierung von Zufallszahlen.....	9
11	Sicherheitsfunktionen	10
11.1	Tot-Test	10
11.2	Permanenter Online-Test	10
12	Statistische Qualität.....	11
13	Sicherheitshinweise	11
14	Anwendungen	11
15	Einsatzumgebung	12
16	Funktionen der Leuchtdioden	13
17	Schnittstelle und Treiber	13
18	Literatur.....	14
[10]	Dokumentation und Analyse des Linux ... - BSI	14

2 Copyright

Copyright (C) 2016

IBB Ingenieurbüro Bergmann
Sonnenweg 3
D-15537 Grünheide

Alle Rechte vorbehalten. Kein Teil dieser Dokumentation darf in irgendeiner Form (Fotokopie, Druck oder andere Verfahren) ohne ausdrückliche Genehmigung des Herstellers reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Der rechtmäßige Erwerb des Physikalischen Zufallsgenerators PRG320 erlaubt eine Nutzung ausschließlich entsprechend Lizenzvertrag.

Ausgabe vom 07.10.2016

Schutzrechte

Für wesentliche Schaltungsdetails des Physikalischen Zufallsgenerators sind Schutzrechte eingetragen.

Patente:

- Deutsches Patent DE 102 23 252 vom 18.06 2003
- Europäisches Patent EP 150 98 38 vom 15.03.2006

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 0172 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 Kryptoanalyse.

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 Kryptoanalyse.

4 Starke Entropie für kryptografische Applikationen

Mit dem PRG320 steht ein professioneller Zufallsgenerator der Klasse PTG.3 (hybrider Zufallsgenerator) für die permanente Generierung von kryptografisch sicherem Zufall zur Verfügung. Dieser Zufallsgenerator hat ein USB-Interface (virtuelle COM) 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.

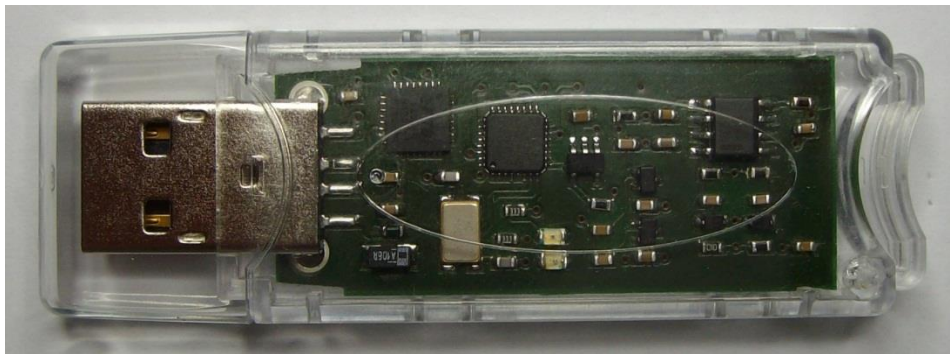
Die Qualität der vom PRG320 erzeugten Zufallszahlen ist qualitativ unvergleichlich höher und sicherer, als jede andere Art der Zufallserzeugung. Zum Verifizieren dieser Aussage stehen diverse Analysen des Zufallsgenerators, auch unter erhöhter thermischer Belastung, zur Verfügung. Die hohe Ausgabegeschwindigkeit des

PRG320 ermöglicht eine Füllung des Entropie-Pools (4096 Bit) in max. 30ms. Eine kryptografische Nachbearbeitung der ausgegebenen Zufallsdaten ist prinzipiell nicht erforderlich.

5 PRG320 in verschiedenen Bauformen

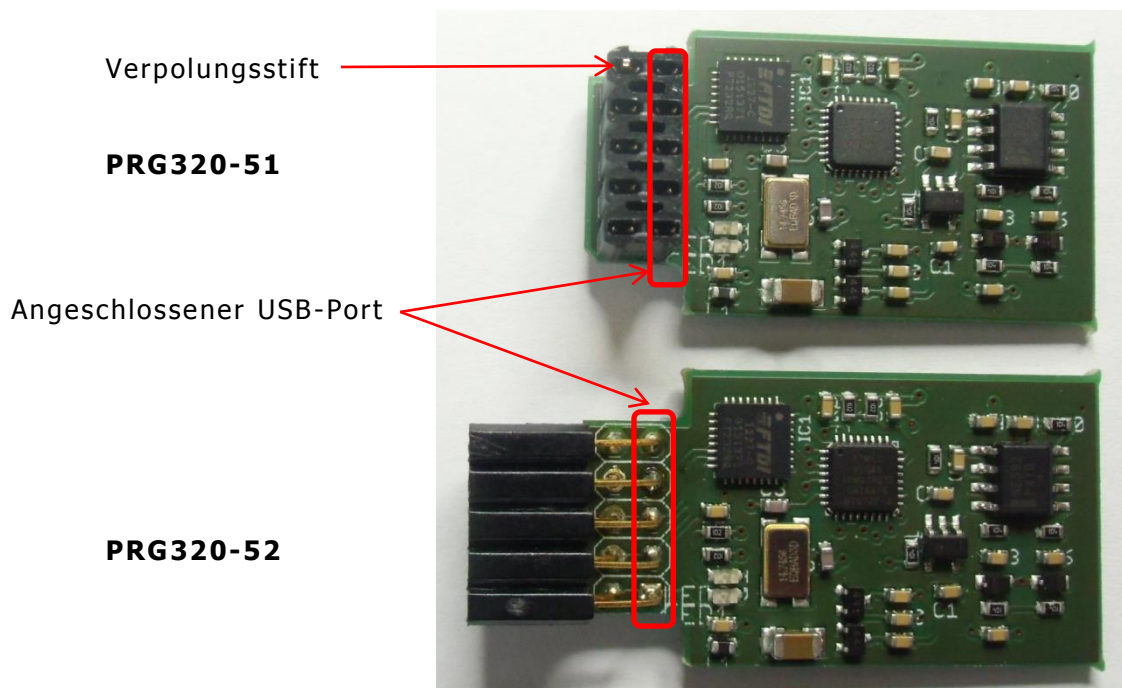
Bei dem PRG320 handelt es sich um einen physikalischen Zufallsgenerator mit einer USB-Schnittstelle (virtuelle COM). Der PRG320 ist voll hardware-kompatibel zum universellen PRG310. Der PRG320 kommt in zwei Ausführungen zum Einsatz:

- PRG320-4 ist für den externen Anschluss an beliebige PC-Systeme mit einem Standard-USB-Anschluss vorgesehen.



PRG320-4

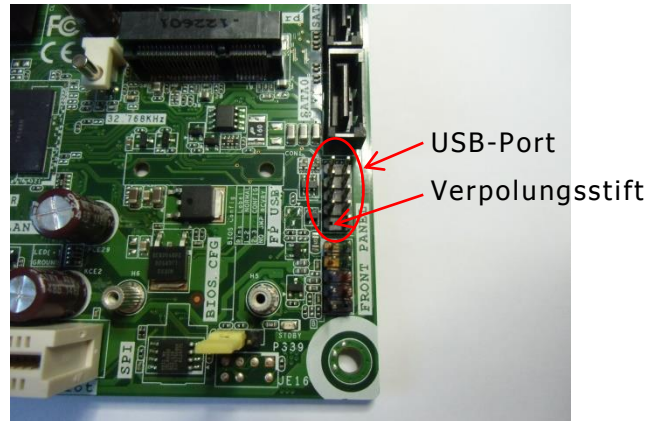
- Der bis auf den Steckverbinder vollkommen schaltungs- und funktionsgleiche PRG320-5 ist für den Einsatz innerhalb beliebiger PC-Systeme konzipiert. Interne USB-Anschlüsse sind auf Mainbord als Pfostensteckverbinder ausgeführt. Die Pfostensteckverbinder auf dem PRG320-5-Modul sind als horizontale und vertikale Ausführung verfügbar (PRG320-51: horizontal, PRG320-52: vertikal).



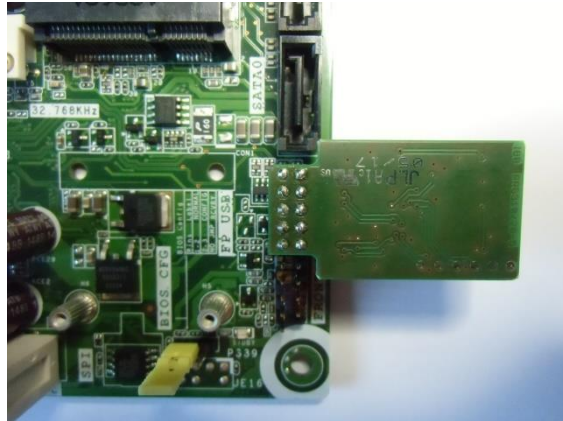
Modifiziertes PRG320-51-Modul

Mainboards werden in zwei Varianten mit USB-Steckverbindern bestückt, so dass sich unterschiedliche Steckkarten der horizontalen Variante ergeben. Hier ein Beispiel:

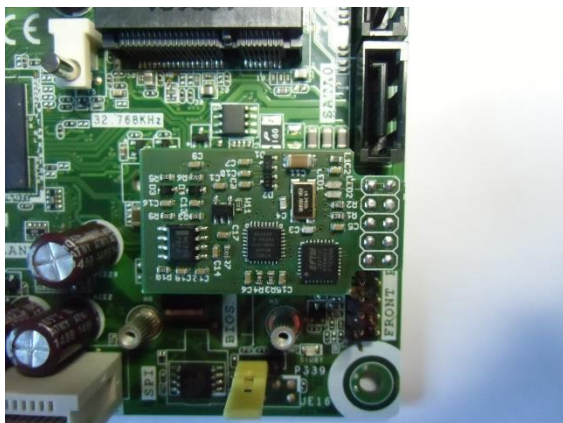
USB-Port auf einem Mainboard



Variante PRG320-51



Variante PRG320-511



Kern des PRG320 ist ein patentierter physikalischer Zufallsgenerator des IBB:

- Deutsches Patent DE 102 23 252 vom 18.06 2003
- Europäisches Patent EP 150 98 38 vom 15.03.2006

6 Technische Daten

Abmessungen:	PRG320-4: 45*19*2,8 mm (ohne USB-Steckverbinder) PRG320-5: 35*20*2,8 mm (ohne Pfostensteckverbinder)
Versorgungsspannung:	5V (+/- 10%)
Stromaufnahme:	max. 35mA
Temperaturbereich:	funktionell und statistisch stabil von -20°C..+85°C
Schnittstellen:	USB1.1-Interface über virtuellen COM, 921.600 bps, Protokoll 8,N,1
Qualitätssicherung:	automatischer Selbstabgleich von Verstärkung und Digitalisierung Tot-Test zur Überwachung der Rauschquellen Abschaltung der Zufallsausgabe bei Ausfall einer Rauschquelle Permanenter Online-Test pro Sekunde zur statistischen Überwachung des Zufallssignals
Entropie:	>7,997 Bits/Byte (ermittelt aus Zufalls-Rohdaten nach Shannon)
0/1-Verhältnis:	garantiert im Bereich 0,49..0,51 (Rohdaten > 8.000 Bit)
Ausgabegeschwindigkeit:	ca. 142 Kbit/s

7 Prinzip der Rauscherzeugung

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, das mit steigender Frequenz abnimmt, die Amplitudenverteilung ist umgekehrt proportional zur Frequenz ($\sim 1/f$). Die verwendeten Rauschquellen sind Z-Dioden, die in Sperrrichtung betrieben werden. Rauschen entsteht hier durch den Lawineneffekt (Avalancheeffekt) in der pn-Sperrschicht des Halbleiterbauelements (Dioden und Transistoren). Dioden und Transistoren lassen sich durch ein kontrolliertes Avalanche-Verhalten vor Zerstörung durch Überspannungen schützen.

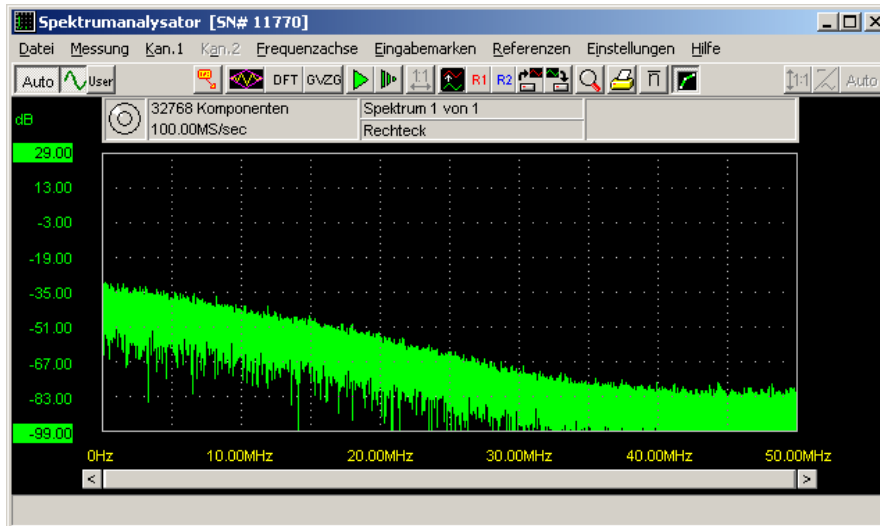


Abbildung: Typisches 1/f-Rauschen nach der Verstärkung

8 Generierung des Zufallssignals

Die Rauschsignale zweier Z-Dioden werden einem Differenzverstärker zugeführt, der eine ca. 300-fache Verstärkung realisiert und durch seine hohe Gleichtaktunterdrückung Störsignale sehr wirksam eliminiert. Ein schneller Schmitt-Trigger mit einer Hysterese von ca. 0,3V digitalisiert das verstärkte Rauschsignal und führt es einem Porteingang des Mikrocontrollers zu.

Das Blockschaltbild zeigt wesentliche Komponente des PRG320:

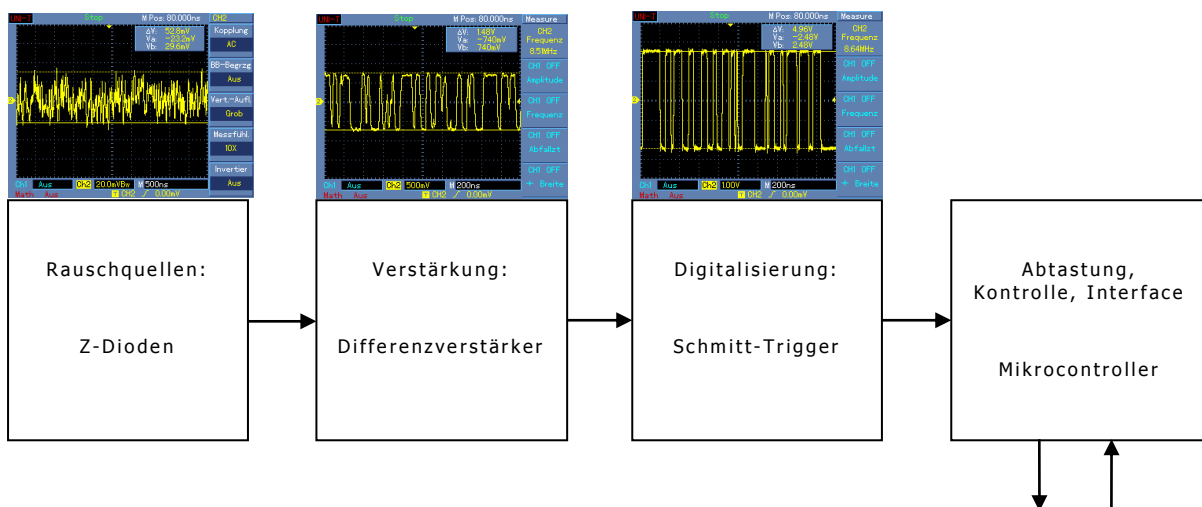


Abbildung: Blockschaltbild des PRG320

9 Entropie

Die Entropie der Zufallsrohdaten ist die entscheidende Eigenschaft eines echten Zufallsgenerators und sollte so hoch als möglich sein. Die reproduzierbare Generierung von Zufallsdaten mit hoher Entropie der PRG320-Applikationen zeigen beispiellose Werte.

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

PRG320-Typ	Mittelwert 0/1	Entropie der Rohdaten
PRG320-4	0.50342629	7.99972901
PRG320-4	0.50048198	7.99999464
PRG320-4	0.50165468	7.99993680
PRG320-51	0.49894816	7.99997446
PRG320-51	0.49868832	7.99996029
PRG320-52	0.50007339	7.99999988

Abbildung: Entropie der PRG-Applikationen

10 Generierung von Zufallszahlen

Grundlage der Entwicklung des PRG320 ist ein stochastisches Modell, mit dem die Leistungsfähigkeit zur Generierung kryptografisch sicherer Zufallszahlen begründet wird. Dieses Modell erklärt:

- die robuste und hohe Entropie der Rauschquelle
- das Prinzip der Abtastung des analogen Rauschsignals,
- die permanente Überwachung der Rauschquelle durch Frequenzmessung
- die kryptografische Nachbearbeitung durch Mayer-Einwegfunktionen
- den permanenten statistischen Online-Test

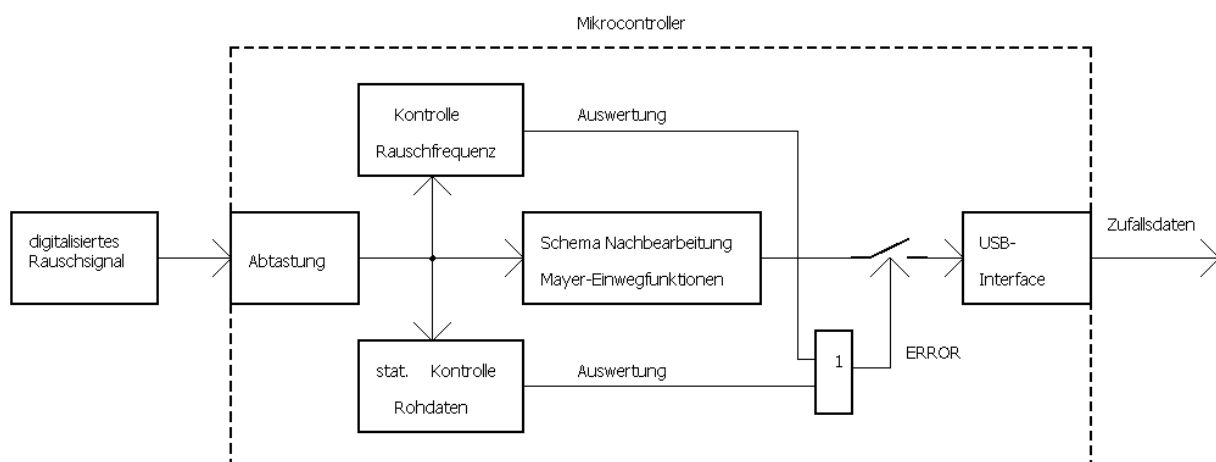


Abbildung: Schema zum stochastischen Modell

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

Die kryptografische Nachbereitung nach PTG.3 nutzt zwei Prinzipien des Entropiesammelns (siehe Schema): das Aufxorieren der Rohdaten für z_1 und die Glättung durch Aufxorieren des mit zufälligem Schlüssel verschlüsselten Zustands für z_2 . 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.

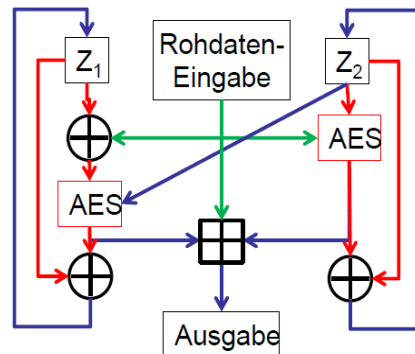


Abbildung: Schema der kryptografischen Nachbearbeitung

Die Rohdaten-Eingabe erfolgt mit 16 Byte des physikalischen Zufallsgenerators und ergeben nach der Verarbeitung 16 Byte Ausgabedaten. Es wird also explizit kein Pseudozufall generiert.

11 Sicherheitsfunktionen

11.1 Tot-Test

Es ist ein Kontrollsystem installiert, welches das digitalisierte Rauschsignal am Anschluss des Mikrocontrollers überwacht. Unmittelbar vor jeder Byte-Generierung des digitalisierten Rauschsignals 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 $16\mu\text{s}$ (entspricht 125 KHz) diese Bedingung nicht erfüllt, wird eine Fehlermeldung generiert, die zur sofortigen Einstellung aller Zufallsausgaben führt
- Dieser Zustand ist nur durch ein positives Ergebnis des weiter laufenden Tot-Test aufzulösen
- Typische Zeiten, um die Bedingung zu erfüllen, sind $2..8\mu\text{s}$

Sollte eine der Rauschquellen ausfallen entsteht am Eingang des Mikrocontrollers ein Mäander von ca. 20ms und wird durch den Tot-Test eindeutig als Fehlzustand erkannt

11.2 Permanenter Online-Test

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

- Sind die Werte innerhalb der statistischen Vorgaben, wird kein Fehler generiert
- Ist einer der 16 Werte im Halbbytetest gleich Null, wird von einem Totalausfall mindestens einer Rauschquelle ausgegangen und jede Zufallsausgabe blockiert.
- Dieser Zustand ist nur durch ein positives Ergebnis des weiter laufenden Online-Test aufzulösen

12 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 für Rohdaten (keine digitale Nachbearbeitung; 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.

13 Sicherheitshinweise

Um eine sichere Generierung von Zufallszahlen zu gewährleisten, sind folgende Sicherheitshinweise zu beachten:

- Zugriff und Nutzung in sicherheitsrelevanten Bereichen ist nur autorisierten Personen zu gestatten
- Der Einsatz des PRG320-5 (interne Variante) erschwert eine Manipulation der Zufallserzeugung und ist zu bevorzugen

14 Anwendungen

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.



Abbildung: PRG320-4 am PC

Insbesondere die Implementierung unter Linux zur permanenten und schnellen Entropie-Erhöhung, ist eine wesentliche Anwendung.

Exemplarische Beispiele für den Einsatz des PRG320:

- Netzwerksicherheit
- Transaktionen beim Homebanking (SSL-Verschlüsselung)
- Dateiverschlüsselung mit Security-Applikationen (Speicher-Sticks)
- einfachere Administration und sichere Verschlüsselung bei drahtloser Datenübertragung: WLAN, Bluetooth, GSM, ZiggBee, Industriedatenfunk
- Internet-Verschlüsselung
- elektronischer Zahlungsverkehr
- Erstellung von PKI-Zertifikaten
- OneTimePad-Verfahren

Weitere Informationen im Kontext finden Sie im Dokument:

„Professionelle_Zufallsgeneratoren.pdf“

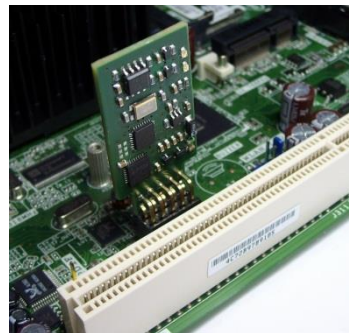


Abbildung: PRG320-51 (links) und PRG320-52 auf einem Mainboard

15 Einsatzumgebung

Der PRG320 ist für den permanenten Einsatz in beliebigen PC-Systemen entwickelt und getestet worden. Auch bei erhöhtem Industriestandard (-20°C bis +85°C) bleiben die Entropiewerte sehr hoch und unterschreiten die Vorgaben aus den AIS31-Forderungen nicht. Getestet wurden Applikationen in einem Temperaturbereich von -60 bis +110°C. Alle Grenzwerte der Zufallsrohdaten (Entropie, Halbbytetest) wurden nicht überschritten. Statistische Analysen, auch für diese Temperaturbereiche, befinden sich auf der Homepage des Autors.

16 Funktionen der Leuchtdioden

Die auf der Platine befindlichen Leuchtdioden reflektieren die jeweils ablaufenden Funktionen und Zustände des PRG320. 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

17 Schnittstelle und Treiber

Die Schnittstelle des PRG320 basiert auf einem Chip der Firma FTDI (<http://www.ftdichip.com/FTDrivers.htm>). Auf der Internetseite werden alle aktuellen Betriebssysteme unterstützt, wobei WINDOWS ab Vista bereits den Chip automatisch erkennt und unterstützt.

Ist der PRG320 erkannt, wird ein unikater virtueller Treiber mit einer nummerierten COM eingerichtet. In der Systemsteuerung ist die eingerichtete COM-Schnittstelle eindeutig spezifiziert.

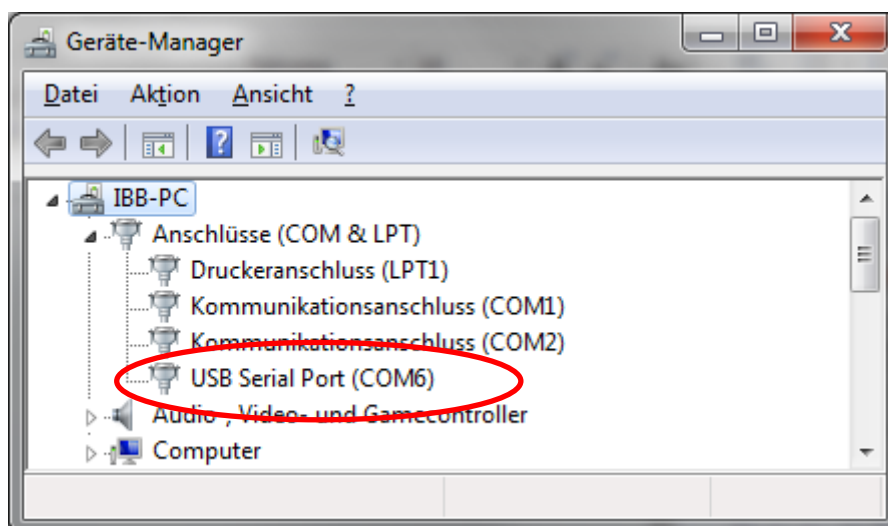


Abbildung: nach dem Einstecken des PRG320 eingerichteter virtueller COM-Port

18 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](#)