DEVELOP.INI
- Beschreibung der Sektion [DeviceX], zuständig für die Detektoren

Detektoren werden im ini-File als Devices geführt, ebenso im weiteren Verlauf. Die im .ini-File für ein Device festzulegenden Eigenschaften und Parameter werden mit dem Sektionsnamen (in eckige Klammern gefasst) DeviceX eingeleitet. Dabei steht X für eine ganze Zahl. Die Anzahl der maximal definierbaren Devices ist mit const int nMaxDeviceAllowed in C_Layer.cpp gegenwärtig auf 3 begrenzt, d.h. X darf maximal 2 sein. Alles was größer ist als nMaxDeviceAllowed wird stillschweigend ignoriert.
Da die einlesende Funktion GetPrivateString nicht case-sensitiv ist, ist die Groß/Kleinschreibung von Sektionsnamen und Keys irrelevant. Folgende Keys (Schlüsselworte unter einem Sektionsnamen) sind möglich (und teilweise auch erforderlich):

Type

Beschreibung	Mit diesem Key wird der Typ des Device spezifiziert, entsprechend des Typs werden unterschiedliche Parameter (Keys) benötigt und entsprechende Konstruktoren aufgerufen.
mögliche Werte	Generic, Stoe-PSD, Radicon, Braun-PSD, PSD, Encoder, Matrox, Test
Defaultwert	Generic
Detektortyp	alle
Hinweis	Der Typ Matrox ist momentan durch die Anweisung #ifdef CCD_Extension deaktiviert. Der Typ Test ist in dem Sinne nicht als Stringangabe erforderlich, vielmehr führt jeder String der nicht zu den möglichen Werten gehört (momentan also auch Matrox) zu einem Device vom Typ Test.

Name

Beschreibung	Hiermit wird dem Device ein Name zu geordnet. Dieser Name wird fortan bei der Arbeit mit dem Device angezeigt
mögliche Werte	Es können die üblichen alphanumerischen Zeichen verwendet werden, insgesamt maximal 80. Von der Verwendung von Semikola ist abzuraten, weil diese als Kommentareinleitung von GetPrivateString interpretiert werden, und nachfolgende Zeichen somit ignoriert würden.
Defaultwert	Counter
Detektortyp	alle

Sound

Beschreibung Damit wird die akustische Ausgabe der Messung ein/ausgeschaltet. Aufgrund der Quelltextanalyse gehen wir davon aus, daß ein Setzen dieses Flags einen Hardwarezugriff zur Folge hat, ein Speaker wird (de)aktiviert.

mögliche Werte 0- aus; 1- ein

Defaultwert 1 (außer bei TRadicon : 0)

Detektortyp alle

Debug

Beschreibung Mit diesem Flag wird die Ausgabe von Debugginginformationen ein- bzw. ausgeschaltet. Es wird auf die Variable *bDebug* geschrieben, diese wird im weiteren Programmverlauf an bestimmten Stellen abgefragt. Wenn sie gesetzt ist, wird die Funktion SetInfo() mit einem String ausgeführt, welcher in die Statuszeile des Steuerprogramms geschrieben wird. Dieses Flag wird nur in Verbindung mit den Detektoren verwendet.

mögliche Werte 0- aus; 1- ein

Defaultwert 0

Detektortyp alle

Hinweis Auch dieses Flag ist unserer Meinung nach für alle Detektoren (abgesehen von Testdetektoren) gedacht worden. Es wird in der Methode Initialize() eingelesen und gesetzt. Jedoch hat der Entwickler im Konstruktor TBraun_PSD dieses Vorgehen wohl vergessen und eine neue Variable samt Key eingeführt: *DebugInfo*. Auf diese wird im weiteren Verlauf jedoch nie wieder zugegriffen, statt dessen wird wie sonst üblich auch in Braun_PSD Methoden auf *bDebug* zugegriffen. (siehe Key *DebugInfo*)

ExposureTime

Beschreibung Hier wird die Länge eines Meßintervalls festgelegt. Der Wert wird in der Variablen *fExposureTime* gespeichert.

mögliche Werte 0.1- 500.0

Defaultwert 4.0

Detektortyp alle

Hinweis Für *fExposureTime* existieren die Methoden *TDevice::GetExposureValues* und *TDevice::SetExposureValues* zum Lesen und Setzen, jedoch wird mindestens dreimal aus der Klasse *TCommonDevParam* direkt auf das Attribut zugegriffen. Der Typ PSD dient unserer Meinung nach dazu einen PSD Testzähler zu erzeugen.

ExposureCounts

Beschreibung Mit diesem Key wird die maximale Anzahl der Impulse pro Meßintervall festgelegt. Der Wert wird in der Variablen *dwExposureCounts* gespeichert.

mögliche Werte 1 - 300000

Defaultwert 10000

Detektortyp alle

Hinweis Für *dwExposureCounts* existieren die Methoden *TDevice::GetExposureValues* und *TDevice::SetExposureValues* zum Lesen und Setzen, jedoch wird mindestens dreimal aus der Klasse *TCommonDevParam* direkt auf das Attribut zugegriffen

OverflowIntensity

Beschreibung Der hier angegebene Wert gibt an, welches der größte mögliche Meßwert ist.

mögliche Werte 0 - ?

Defaultwert 50000.0

Detektortyp PSD

BaseAddr

Beschreibung Hier wird die I/O-Adresse (hexadezimal) für die Kommunikation mit der Hardware angegeben. Der Wert wird im Attribut *nBaseAddr* abgelegt.

mögliche Werte 0x100 – 0x3F8 in 4h Schritten (Quelle: BraunPSD Betriebsanleitung)

Defaultwert 0x300

Detektortyp PSD

Hinweis Das Programm führt an keiner Stelle eine Überprüfung des Inhaltes von *nBaseAddr* durch.

SignalGrowUp

Beschreibung Dieses Flag gibt an, ob auch vor Messungsende Zwischendaten übernommen werden sollen.

mögliche Werte 0- nein; 1- ja

Defaultwert 1

Detektortyp PSD

HVRegelung

Beschreibung Mit diesem Flag gibt man an, ob die Hochspannung am Gerät gesteuert werden soll.

mögliche Werte 0- nein; 1- ja

Defaultwert 0

Detektortyp PSD

ReadLeftFirst

Beschreibung Mit diesem Flag legt man fest, daß beim Auslesen der Geräte von links begonnen werden soll.

mögliche Werte 0- nein; 1- ja

Defaultwert 1

Detektortyp PSD

AngleStep

Beschreibung Dieser Key gibt den Winkel an, der von einem Kanal abgedeckt werden soll.

mögliche Werte 0-?

Defaultwert 1.0

Detektortyp PSD

Unit

Beschreibung Mit diesem Key wird die Einheit der Winkelangabe festgelegt.

mögliche Werte Grad, Sekunden, Minuten, Minuts

Defaultwert Sekunden

Detektortyp PSD

AddedChannels

Beschreibung Hiermit wird die Anzahl der zu einem Meßwert zusammenzufassenden Kanäle angegeben.

mögliche Werte 1 - ?

Defaultwert 4

Detektortyp PSD

FirstChannel

Beschreibung Dieser Key gibt den ersten zu benutzenden Kanal an.

mögliche Werte 0- 4095

Defaultwert 0

Detektortyp PSD

Hinweis Wenn der Wert von *FirstChannel* größer oder gleich *LastChannel* ist, wird für *FirstChannel* =0 und für *LastChannel*=4095 gesetzt.

LastChannel

Beschreibung Hier wird der letzte zu benutzende Kanal angegeben.

mögliche Werte 0- 4095

Defaultwert 4095

Detektortyp PSD

Hinweis Wenn der Wert von *FirstChannel* größer oder gleich *LastChannel* ist wird für *FirstChannel* =0 und für *LastChannel* =4095 gesetzt.

IOAddr

Beschreibung Hier wird die I/O-Adresse für die Hardwarekommunikation mit dem Radicon-Device angegeben, sie wird in *Rdd* gespeichert.

mögliche Werte 0x100 – 0x160 in 0x10 Schritten, 0x180 – 0x1E0 in 0x10 Schritten, 0x300, 0x310, 0x330 – 0x360 in 0x10 Schritten, 0x390, 0x3E0 (Quelle: SCSCS Operator‘s Manual)

Defaultwert 0x100

Detektortyp Radicon

Hinweis Das Programm führt an keiner Stelle eine Überprüfung des Inhaltes von *Rdd* durch. Des weiteren wäre anzumerken, daß es unnötig ist, zwei Bezeichnungen für den gleichen Sachverhalt zu vergeben. Es handelt sich zwar bei IOAddr und BaseAddr nicht um den gleichen Inhalt (IOAddr ist für Radicon, BaseAddr für PSD), da sie aber nie in der gleichen Sektion zugleich vorkommen und sowieso getrennte Methoden zum Einlesen dieser Werte existieren, wäre es sinnvoll und ein weiterer Schritt in Richtung Benutzerfreundlichkeit, sich auf eine Bezeichnung zu einigen.

UpperThresh

Beschreibung Dieser Wert repräsentiert die obere Schranke für die Hochspannung, mit der das Gerät betrieben werden soll.

mögliche Werte 1 – 1023

Defaultwert 950

Detektortyp Radicon

LowerThresh

Beschreibung Dieser Wert repräsentiert die untere Schranke für die Hochspannung, mit der das Gerät betrieben werden soll.

mögliche Werte 1 – 1023

Defaultwert 150

Detektortyp Radicon

HighVoltage

Beschreibung Dieser Key gibt den Wert der Hochspannung bei Initialisierung an.

mögliche Werte 1 - 900

Defaultwert 640

Detektortyp Radicon

EnergyScale

Beschreibung Dieser Key bestimmt die Energieskalierung.

mögliche Werte 0,1,2,3

Defaultwert 2

Detektortyp Braun

AbbruchMitShutter

Beschreibung Dieses Flag setzt die Option: Messung abbrechen mit Shutter.

mögliche Werte 0- nein; 1- ja

Defaultwert 0

Detektortyp Braun

PositionScale

Beschreibung Dieser Key bestimmt die Positionsskalierung.

mögliche Werte 0,1,2,3

Defaultwert 2

Detektortyp Braun

EnergyHigh

Beschreibung Dieser Key legt die obere Grenze für das Energiefenster fest.

mögliche Werte 0(?) bis (0xFFF / (EnergyScale+1))

Defaultwert 870

Detektortyp Braun

EnergyLow

Beschreibung Hiermit wird die untere Grenze für das Energiefenster festgelegt.

mögliche Werte 0(?) bis (0xFFF / (EnergyScale+1))

Defaultwert 526

Detektortyp Braun

MuxTimeDet1

Beschreibung Dieser Key dient der Einstellung der Multiplexerzeit für den Detektor 1. Unklar ist was genau damit gemeint ist. Wir konnten diese Information lediglich aus einem einem String ziehen.

mögliche Werte 0 – 61439(0xEFFF)

Defaultwert 40

Detektortyp Braun

Ratemeter

Beschreibung Über dieses Flag wird das externe Ratemeter selektiert.

mögliche Werte 0- aus; 1- ein

Defaultwert 0

Detektortyp Braun

RealLifeTime

Beschreibung Mit diesem Flag kann zwischen Real- und Lifetimeberechnung umgeschaltet werden.

mögliche Werte 0- Realtime, 1-Lifetime

Defaultwert 0

Detektortyp Braun

DeathTime

Beschreibung Dieser Wert gibt den notwendigen Mindestimpulsabstand an.

mögliche Werte 0- 99

Defaultwert 10

Detektortyp Braun

DelayFast

Beschreibung Die Bedeutung dieses Key konnten wir nicht endgültig klären, es handelt sich jedoch vermutlich um einen Verzögerungswert, welcher bei der Kommunikation mit der Hardware einen Timeout-Wert darstellt.

mögliche Werte Ebenso fanden wir keine Anhaltspunkte für die möglichen Werte.

Defaultwert 2

Detektortyp Braun

DelaySlow

Beschreibung Die Bedeutung dieses Key konnten wir ebenfalls nicht endgültig klären, es handelt sich jedoch vermutlich auch hier um einen Verzögerungswert, welcher bei der Kommunikation mit der Hardware einen Timeout-Wert darstellt.

mögliche Werte Ebenso fanden wir keine Anhaltspunkte über die möglichen Werte.

Defaultwert 4

Detektortyp Braun

DebugInfo

Beschreibung Dieser Key soll vermutlich die Ausgabe von Debugging-Informationen aktivieren, jedoch wird er nur einmal gesetzt und im weiteren Verlauf nie wieder verwendet, was vermutlich damit zu erklären ist, daß der bereits weiter oben erläuterte Key *Debug* diese Aufgabe übernimmt. Vermutlich stellt *DebugInfo* einen Moment der Unklarheit des Entwicklers über die bereits implementierten Attribute dar, mit anderen Worten: *DebugInfo* ist überflüssig und kann bedenkenlos weggelassen werden, da auch die Routinen der PSD-Klasse mit dem Attribut *bDebug* (entsprechende Variable für den Key *Debug*) arbeiten.

mögliche Werte 0- aus; 1- ja

Defaultwert 0

Detektortyp Braun

Hinweis siehe Information weiter oben über Key *Debug*

IOAddr

Beschreibung Dieser Key dient ebenfalls der Festlegung der I/O-Kommunikationsadresse. Er wird von *TAm9513a::LoockUp* eingelesen.

mögliche Werte ?

Defaultwert 0x230

Detektortyp AM9513 (Generic)

Hinweis Siehe Hinweise oben unter BaseAddr und IOAddr

TimeCorrection

Beschreibung Die Bedeutung dieses Key konnte von uns nicht eindeutig geklärt werden. Es bleibt uns lediglich vom Namen auf die Bedeutung zu schließen, was in diesem Fall auf die Funktion einer Zeitkorrekturkonstante deutet.

mögliche Werte Ebenfalls gab es keine Anhaltspunkte für die hier möglichen Werte.

Defaultwert 1.0

Detektortyp AM9513 (Generic)