

# Reviewprotokoll

Projektname: **086350-S1NN-ADLATUS-TMS470R-MucSuc**

Interne Auftragsnummer: **086350**

Bearbeiter: **Werner**

Datum: **21.11.2008**

Kunde: **S1nn**

**Projektname**

Kunde Auftragsnummer:

Kunde Auftragsdatum:

**Teilnehmer:**

**Thema: Review ADLATUS VW/AUDI mit UDS Protokoll mit Erweiterung für das Flashen des Parrot**

**Review-Punkte:**

**Ergebnis: Bearbeiter:**

1 Das Projekt setzt nicht auf die aktuell bei VW/AUDI vorliegende Version 1.08 sondern auf eine unbekannte Vorgängerversion auf. B

2 **ADLATUS\_ParrotHal.c**

2.1 Die Elemente der Struktur „t\_PARhal\_HalCtrl\_ST“ werden nicht initialisiert. Dieser Fehler wurde gefixt.

A

2.2 Die Klammer um \*t\_StartTime\_PUL ist nicht notwendig

void FUN\_PARhal\_TimeoutStart\_V(ULONG\* t\_StartTime\_PUL)

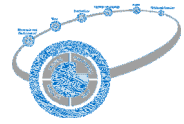
D

```
{
    /*-----
    -*/
    /* get current timer value
    */
    /*-----
    -*/
    (*t_StartTime_PUL) = FUN_PARhal_GetPerformanceCounter_UL();

    /*-----
    -*/
    /* return
    */
    /*-----
    -*/
    return;

} /*-- END OF FUNCTION 'FUN_PARhal_TimeoutStart_V' --+--+--+--+--+
+--*/
```

2.3. Es sind entsprechend der SMART Codierrichtlinien Kommetare und Doxigen- Kommentare gemischt zu finden. C



### 3 FlashLib\_TMS470\_PARROT.c

1. Es wird einer Variablen ausserhalb einer Funktion ein Wert zugewiesen. Da es keine Kopierfunktion (cinit) gibt, erfolgt diese Zuweisung in Wirklichkeit nicht. Dieser Fehler wurde gefixt. B

```
UBYTE t_ParrotState_UB=d_ParrotInitState;
```

2. Einem ulong Register wird ein uword Wert zugewiesen. Laut MISRA sollte ein anderer Postfix verwendet werden. C

```
Anstatt: e_GIO_ST.Port_ST[3].Dset_UL = 0x00002u; /* BOOTS high */  
Lieber: e_GIO_ST.Port_ST[3].Dset_UL = 0x00002u1; /* BOOTS high */
```

4. /\* setup loop counter to allow 128 tries \*/  
t\_endlessLoopProtection\_UB = 0xFEu; D  
0xFE sind 254.

5. Die Abfrage "if (d\_ParrotUpdaterInDevice != t\_ParrotState\_UB)" ist unnötig, da „t\_ParrotState\_UB“ niemals in den State gesetzt wird. Folglich wird der else Zweig nie durchlaufen. D

6. Die Codesequenz:

```
if (0xFF != t_rxbuffer_UB)  
{  
    t_index_UB = 0;  
}
```

Ist kein Abbruchkriterium, sondern bewirkt bei einem Fehler, dass die Schleife nochmals vollständig durchlaufen wird. B

7. Das Casting wird nicht in allen Fällen eingesetzt. Manchmal gibt es ein (UBYTE)0x...u manchmal nicht. C

8. Die gebildete Checksumme ist keine CRC, sondern eine Aufaddition. D

```
/* calc crc */  
t_BootloadersCrc_UW += (UWORD)*((flashParam->data) +  
t_index_UW);
```

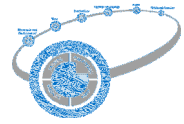
9. Anstatt if (TRUE == flashParam->Command\_UB) sollte if (FALSE != flashParam->Command\_UB) verwendet werden D

### 5 ADLATUS\_Eep.c

1. In der Funktion „FUN\_BioEepWrite“ sollte anstatt der Funktion „FUN\_CCwtdog\_TriggerWatchdogEraseMem\_V“ die Funktion „FUN\_CCtim\_SystemTimerCheck\_V“ aufgerufen werden. B

### 6 ADLATUS\_ExternalWatchdog.c

1. Der Define für „TOGGLEPIN“ ist nicht auskommentiert. D.h der Pin wird angesteuert. B



## 7 tms470regs.h

1. Es gibt eine Zeile :

```
Extern SCC_ST e_SCC_ST, e_SCC3_ST, e_SCC3_ST;
```

C

Es muss aber

```
extern SCC_ST e_SCC_ST, e_SCC2_ST, e_SCC3_ST;
```

heissen (dies ist der Name der Referenz im lkf).

## 8 ADLATUS\_KWP2000Audi.c

1. Vor dem Aufruf der Funktion FlashInit wird die Funktion „FUN\_CCwdog\_TriggerWatchdogEraseMem\_V“ aufgerufen, besser wäre es wenn an Stelle dessen die Funktion „FUN\_CCwdog\_TriggerWatchdogProgMem\_V“ benützt würde. D

## 9 FlashLib\_TMS470.c

1. In der Funktion „FlashWrite“ gibt es die Befehlsfolge

```
if (flashParam->address > d_Bank0End)
{
    t_Core_UL = 0x01u;
}
else
{
    if (flashParam->address > d_Bank1End)
    {
        t_Core_UL = 0x02u;
    }
    else
    {
        t_Core_UL = 0x00;
    }
} /* -5- END OF if(..) -----*/
```

B

Die Abfragereihenfolge ist falsch, sie müsste

```
if (flashParam->address > d_Bank1End)
{
    t_Core_UL = 0x02u;
}
else
{
    if (flashParam->address > d_Bank0End)
    {
        t_Core_UL = 0x01u;
    }
    else
    {
        t_Core_UL = 0x00;
    }
} /* -5- END OF if(..) -----*/
```

lauten. Dieser Fehler wurde gefixt.

- 10 Im His-Treiber des TMS470 wird der interne Watchdog getriggert im His-Treiber für den Parrot nicht. Der interne Watchdog sollte in beiden Modulen gleich behandelt werden. D

11

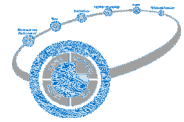
12

KUNDE Anschrift:	SMART Anschrift:
	SMART Electronic Development GmbH
	Rötestraße 17
	70197 Stuttgart
	Tel: 0711/25521-0
	Fax: 0711/25521-10

Datum: 21.11.08	Name: werner	O:\Projekte\Bootblock\S1NN\086350-S1NN-ADLATUS-TMS470R-MucSuc\2SW\6EXT\Auslieferung_2008_11_17_Offiziell\Vorab\2	Seite: 3 von 4
-----------------	--------------	--	----------------



TED (Technische Entwicklungsdokumentation)  
Bereich: GP3 - Auftragsbezogene Entwicklung  
Reviewprotokoll



web: [www.smart-gmbh.de](http://www.smart-gmbh.de)

**Review durch KUNDE:**

**Review durch SMART:**

Datum:	_____	Datum:	21.11.2008
Ort:	_____	Ort:	Stuttgart
Unterschrift:	_____	Unterschrift:	Jürgen Werner