

Device	Command	ExpResponse	Delay	Version From	Version To
RadiCentre	*IDN?	D\A\RL\EU\ RadiCentre version,? \d+\.\d+\.\d+			
EMCenter	*IDN?	ETS Lindgren EMCenter version \d+\.\d+\.\d+			
RadiCentre:EMCenter	LOCAL	OK			
RadiCentre:EMCenter	RESET	OK			
RadiCentre:EMCenter	ID_NUMBER?	\d+(\.\d+)?(7)			
RadiCentre:EMCenter	VERSION_HW?	\d+(\.\d+){0,2}			
RadiCentre:EMCenter	VERSION_SW?	\d+(\.\d+){0,2}			
	<b>Alle commando's voor de RSW1002A kaart</b>				
RSW1002A	*IDN?	RSW1002A_Version \d+\.\d+\.\d+		10	
RSW1002A	LOCAL	OK		100	
RSW1002A	RESET	OK		100	
RSW1002A	INT_RELAY_A_NO	NO		200	
RSW1002A	INT_RELAY_B_NC	NC		200	
RSW1002A	INT_RELAY_A?	NO		10	
RSW1002A	INT_RELAY_B?	NC		10	
RSW1002A	INT_RELAY_B_NO	NO		200	
RSW1002A	INT_RELAY_A?	NO		10	
RSW1002A	INT_RELAY_B?	NO		10	
RSW1002A	INT_RELAY_A_NC	NC		200	
RSW1002A	INT_RELAY_A?	NC		10	
RSW1002A	INT_RELAY_B?	NO		10	
RSW1002A	INT_RELAY_B_NC	NC		200	
RSW1002A	INT_RELAY_A?	NC		10	
RSW1002A	INT_RELAY_B?	NC		10	
RSW1002A	INT_RELAY_A_NOC	ERROR 1."		10	
RSW1002A	INT_RELAY_C_NO	ERROR 1."		10	
RSW1002A	EXT_RELAY_1_ACTIVE	ACTIVE		200	
RSW1002A	EXT_RELAY_2_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_3_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_4_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_1?	ACTIVE		10	
RSW1002A	EXT_RELAY_2?	IDLE		10	
RSW1002A	EXT_RELAY_3?	IDLE		10	
RSW1002A	EXT_RELAY_4?	IDLE		10	
RSW1002A	EXT_RELAY_2_ACTIVE	ACTIVE		200	
RSW1002A	EXT_RELAY_1?	ACTIVE		10	
RSW1002A	EXT_RELAY_2?	ACTIVE		10	
RSW1002A	EXT_RELAY_3?	IDLE		10	
RSW1002A	EXT_RELAY_4?	IDLE		10	
RSW1002A	EXT_RELAY_3_ACTIVE	ACTIVE		200	
RSW1002A	EXT_RELAY_1?	ACTIVE		10	
RSW1002A	EXT_RELAY_2?	ACTIVE		10	
RSW1002A	EXT_RELAY_3?	ACTIVE		10	
RSW1002A	EXT_RELAY_4?	IDLE		10	
RSW1002A	EXT_RELAY_4_ACTIVE	ACTIVE		200	
RSW1002A	EXT_RELAY_1?	ACTIVE		10	
RSW1002A	EXT_RELAY_2?	ACTIVE		10	
RSW1002A	EXT_RELAY_3?	ACTIVE		10	
RSW1002A	EXT_RELAY_4?	ACTIVE		10	
RSW1002A	EXT_RELAY_1_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_1?	IDLE		10	
RSW1002A	EXT_RELAY_2?	ACTIVE		10	
RSW1002A	EXT_RELAY_3?	ACTIVE		10	
RSW1002A	EXT_RELAY_4?	ACTIVE		10	
RSW1002A	EXT_RELAY_2_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_1?	IDLE		10	
RSW1002A	EXT_RELAY_2?	IDLE		10	
RSW1002A	EXT_RELAY_3?	ACTIVE		10	
RSW1002A	EXT_RELAY_4?	ACTIVE		10	
RSW1002A	EXT_RELAY_3_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_1?	IDLE		10	
RSW1002A	EXT_RELAY_2?	IDLE		10	
RSW1002A	EXT_RELAY_3?	IDLE		10	
RSW1002A	EXT_RELAY_4?	ACTIVE		10	
RSW1002A	EXT_RELAY_4_IDLE	IDLE		200	
RSW1002A	EXT_RELAY_1?	IDLE		10	
RSW1002A	EXT_RELAY_2?	IDLE		10	

Bijlage I - Commando Database

RSW1002A	EXT_RELAY_3?	IDLE	10		
RSW1002A	EXT_RELAY_4?	IDLE	10		
RSW1002A	EXT_RELAY_1_INACTIVE	ERROR 1.*	10		
RSW1002A	EXT_RELAY_5_ACTIVE	ERROR 1.*	10		
RSW1002A	RANDOMINVAIIDCOMMAND	ERROR 1.*	10		
RSW1002A	INT_RELAY_A_NO	NO	100		
RSW1002A	INT_RELAY_B_NO	NO	100		
RSW1002A	EXT_RELAY_1_ACTIVE	ACTIVE	100		
RSW1002A	EXT_RELAY_2_ACTIVE	ACTIVE	100		
RSW1002A	EXT_RELAY_3_ACTIVE	ACTIVE	100		
RSW1002A	EXT_RELAY_4_ACTIVE	ACTIVE	200		
RSW1002A	REBOOT SYSTEM	OK	3000		
RSW1002A	INT_RELAY_A?	NC	10		
RSW1002A	INT_RELAY_B?	NC	10		
RSW1002A	EXT_RELAY_1?	IDLE	10		
RSW1002A	EXT_RELAY_2?	IDLE	10		
RSW1002A	EXT_RELAY_3?	IDLE	10		
RSW1002A	EXT_RELAY_4?	IDLE	10		
USB1004A	*IDN?	D\A\R\E\I\, RadiPower USB1004A, \d+\. \d+\. \d+	10		
7002-001	*IDN?	ETS-Lindgren, EMPower 7002-001, \d+\. \d+\. \d+			
USB1004A;7002-001	RESET	OK	500		
USB1004A;7002-001	REBOOT SYSTEM	OK	3000		
	<b>RadiPower RPR2006P</b>				
RPR2006C	*IDN?	DARE\I\, RPR2006C, (2\.[3-9]\d){3-9}\. \d\d	10		
7002-002	*IDN?	ETS-Lindgren, 7002-002, (2\.[3-9]\d){3-9}\. \d\d	10		
RPR2006P	*IDN?	DARE\I\, RPR2006P, (2\.[3-9]\d){3-9}\. \d\d	10		
7002-003	*IDN?	ETS-Lindgren, 7002-003, (2\.[3-9]\d){3-9}\. \d\d	10		
RPR2006W	*IDN?	DARE\I\, RPR2006W, \d+\. \d+	10		
RPR2018C	*IDN?	DARE\I\, RPR2018C, (2\.[3-9]\d){3-9}\. \d\d	10		
7002-004	*IDN?	ETS-Lindgren, 7002-004, (2\.[3-9]\d){3-9}\. \d\d	10		
RPR2018P	*IDN?	DARE\I\, RPR2018P, (2\.[3-9]\d){3-9}\. \d\d	10		
7002-005	*IDN?	ETS-Lindgren, 7002-005, (2\.[3-9]\d){3-9}\. \d\d	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ID_NUMBER?	\d+\. \d+\. (7)	10		
RPR2006; RPR2018; 7002-002; 7002-003; 7002-004; 7002-005	VERSION_SW?	(2\.[3-9]\d){3-9}\. \d\d	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	VERSION_HW?	\d+\. \d+	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	REBOOT SYSTEM	OK	500		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	RESET	OK	500		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	TEMPERATURE?	([0-3]\d? \d){400}	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	BAUD?	1	10		
RPR2006P;RPR2018P;7002-003;7002-005	MODE 1	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	MODE?	1	10		
RPR2006P;RPR2018P;7002-003;7002-005	MODE 5	ERROR_50	100		
RPR2006P;RPR2018P;7002-003;7002-005	MODE?	1	10		
RPR2006P;RPR2018P;7002-003;7002-005	MODE 0	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	MODE?	0	10		
RPR2006C;RPR2018C;7002-002;7002-004	MODE?	ERROR 1			
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	STORE	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE 1	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE?	1	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE 5	ERROR_50	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE?	1	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE 0	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE?	0	10		
RPR2006;7002-002;7002-003	FREQUENCY? MIN	9 kHz	10		
RPR2006;7002-002;7002-003	FREQUENCY? MAX	6000000 kHz	10		
RPR2006;7002-002;7002-003	FREQUENCY 9	OK	100		
RPR2006;7002-002;7002-003	FREQUENCY?	9 kHz	10		
RPR2006;7002-002;7002-003	FREQUENCY 6000000 KHZ	OK	100		
RPR2006;7002-002;7002-003	FREQUENCY?	6000000 kHz	10		
RPR2006;7002-002;7002-003	FREQUENCY 8	ERROR_51	100		
RPR2006;7002-002;7002-003	FREQUENCY 6100000 KHZ	ERROR_52	100		
RPR2006;7002-002;7002-003	FREQUENCY?	6000000 kHz	10		
RPR2018;7002-004;7002-005	FREQUENCY? MIN	10000 kHz	10		
RPR2018;7002-004;7002-005	FREQUENCY? MAX	18000000 kHz	10		
RPR2018;7002-004;7002-005	FREQUENCY 10000	OK	100		
RPR2018;7002-004;7002-005	FREQUENCY?	10000 kHz	10		
RPR2018;7002-004;7002-005	FREQUENCY 18000000 KHZ	OK	100		

RPR2018;7002-004;7002-005	FREQUENCY?	18000000 kHz	10		
RPR2018;7002-004;7002-005	FREQUENCY 9999	ERROR_51	100		
RPR2018;7002-004;7002-005	FREQUENCY 18000001 KHZ	ERROR_52	100		
RPR2018;7002-004;7002-005	FREQUENCY?	18000000 kHz	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 1	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	1	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 2	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	2	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 3	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	3	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 4	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	4	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 5	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	5	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 6	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	6	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 7	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	7	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER 8	ERROR_52	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	7	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER AUTO	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	AUTO	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET -100	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	-100l.00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET 100 dBm	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	100l.00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET 0 dBm	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	0l.00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET -101 dBm	ERROR_603	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	0l.00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET 101 dBm	ERROR_602	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	0l.00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT 1	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT?	1	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER?	\d\.\d\d (n u m)?W	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT 5	ERROR_50	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT?	1	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT 0	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT?	0	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER?	-?d?d\.\d\d dBm	10		
RPR2006;7002-002;7002-003	VBW 0	OK	100		
RPR2006;7002-002;7002-003	VBW?	0	10		
RPR2006;7002-002;7002-003	VBW 1	OK	100		
RPR2006;7002-002;7002-003	VBW?	1	10		
RPR2006;7002-002;7002-003	VBW 2	OK	100		
RPR2006;7002-002;7002-003	VBW?	2	10		
RPR2006;7002-002;7002-003	VBW 3	OK	100		
RPR2006;7002-002;7002-003	VBW?	3	10		
RPR2006;7002-002;7002-003	VBW 5	ERROR_50	100		
RPR2006;7002-002;7002-003	VBW?	3	10		
RPR2006;7002-002;7002-003	VBW AUTO	OK	100		
RPR2006;7002-002;7002-003	VBW?	AUTO	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED 20	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED?	20	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED 100	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED?	100	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED 1000	OK	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED?	1000	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED 1000000	ERROR_50	100		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED?	1000	10		
	<b>Special commands for mode 2</b>				
RPR2006P;RPR2018P;7002-003;7002-005	MODE 2	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	MODE?	2	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_RESET	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_STATUS?	0l	10		
RPR2006C;RPR2018C;7002-002;7002-004	ACQ_LOG_STATUS?	ERROR_1	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_DATA?	((-?d?d\.\d\d dBm);+(-?d?d\.\d\d dBm)) (NO DATA)	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_THRESHOLD -60.0 dBm	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_THRESHOLD?	-60l.00	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_THRESHOLD 12.0 dBm	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_THRESHOLD?	12l.00	10		

Bijlage I - Commando Database

RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG THRESHOLD -80.0 dBm	ERROR_602	100		
_RPR2006P:;_RPR2018P	ACQ LOG THRESHOLD?	12\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG THRESHOLD 15.0 dBm	ERROR_603	100		
RPR2006P:;_RPR2018P	ACQ LOG THRESHOLD?	12\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER 0,0,2	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER?	0;0;2	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER 1,0,3	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER?	1;0;3	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER 0,1,10	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER?	0;1;10	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER 9,9,9	ERROR_50	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIGGER?	0;1;10	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG MAX?	:-?;d?;d\,;d\,d dBm	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_DIST 2	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_DIST?	2	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_TRIG_DIST 11	ERROR_50	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_TRIG_DIST?	2	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_DIST 10	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_DIST?	10	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ AUTO_TRIGGER 1	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ AUTO_TRIGGER?	1	10		
_RPR2006P:;_RPR2018P	ACQ AUTO_TRIGGER 5	ERROR_50	100		
RPR2006P:;_RPR2018P	ACQ AUTO_TRIGGER?	1	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ AUTO_TRIGGER 0	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ AUTO_TRIGGER?	0	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY 0	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY?	0	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY 2000000	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY?	2000000	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY 2000001	ERROR_50	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_DELAY?	2000000	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_HOLDOff 0	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_HOLDOff?	0	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG TRIG_HOLDOff 1000000	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_TRIG_HOLDOff?	1000000	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_TRIG_HOLDOff 1000001	ERROR_50	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG_TRIG_HOLDOff?	1000000	10		
	<b>Special commands for mode 3</b>		100		
RPR2006P:RPR2018P:7002-003;7002-005	MODE 3	OK	10		
RPR2006P:RPR2018P:7002-003;7002-005	MODE?	3	100		
RPR2006C:RPR2018C:7002-002;7002-004	BM MEASURE_PERIOD 1	ERROR_1	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM MEASURE_PERIOD 1	OK	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM MEASURE_PERIOD?	1	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_MEASURE_PERIOD 1000 ms	OK	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_MEASURE_PERIOD?	1000	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_MEASURE_PERIOD 1001	ERROR_50	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_MEASURE_PERIOD?	1000	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER 0	OK	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER?	0	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER 5000	OK	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER?	5000	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER 5001	ERROR_50	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_NOISE_TIMER?	5000	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL -60	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL?	-60\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL 12 DBM	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL?	12\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL -80	ERROR_602	100		
_RPR2006P:;_RPR2018P	BM_TRIG_LEVEL?	12\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_TRIG_LEVEL 15 DBM	ERROR_603	100		
RPR2006P:;_RPR2018P	BM_TRIG_LEVEL?	12\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_GO	OK	500		
RPR2006P:RPR2018P:7002-003;7002-005	BM_STAT?	1	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_BURST_COUNT?	\,d+	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_BURST_DATA 1	(\,d+;\,d+;-;?;d+;\,d(2)) (NO DATA)	10		
RPR2006P:RPR2018P:7002-003;7002-005	BM_BURST_DATA_DUMP	(\,d+;\,d+;-;?;d+;\,d(2)) (NO DATA)	10		
	<b>Test threshold and trigger when offset is configured</b>				
RPR2006P:RPR2018P:7002-003;7002-005	POWER_OFFSET 100 dBm	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG THRESHOLD 40.0 dBm	OK	100		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG THRESHOLD?	40\,00	10		
RPR2006P:RPR2018P:7002-003;7002-005	ACQ LOG THRESHOLD 112.0 dBm	OK	100		

RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_THRESHOLD?	112\,00	10		
RPR2006P;RPR2018P;7002-003;7002-005	BM_TRIG_LEVEL_40	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	BM_TRIG_LEVEL?	40\,00	10		
RPR2006P;RPR2018P;7002-003;7002-005	BM_TRIG_LEVEL_112_DBM	OK	100		
RPR2006P;RPR2018P;7002-003;7002-005	BM_TRIG_LEVEL?	112\,00	10		
	test done, set back to default and check				
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	RESET	OK	500		
RPR2006P;RPR2018P;7002-003;7002-005	MODE?	0	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	AUTO_STORE?	0	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FREQUENCY?	13000000 kHz	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	FILTER?	AUTO	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_OFFSET?	0\,00	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	POWER_UNIT?	0	10		
RPR2006;7002-002;7002-003	VBW?	3	10		
RPR2006;RPR2018;7002-002;7002-003;7002-004;7002-005	ACQ_SPEED?	1000	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_TRIGGER?	0;1;2	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_AUTO_TRIGGER?	0	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_DELAY?	0	10		
RPR2006P;RPR2018P;7002-003;7002-005	ACQ_LOG_TRIG_HOLDOFF?	0	10		
RPR2006P;RPR2018P;7002-003;7002-005	BM_MEASURE_PERIOD?	1000	10		
RPR2006P;RPR2018P;7002-003;7002-005	BM_NOISE_TIMER?	10	10		
RPR2006P;RPR2018P;7002-003;7002-005	BM_TRIG_LEVEL?	-40\,00	10		
Alle commando's voor de RSW2006A kaart					
RSW2006A	*IDN?	RSW2006A, Version \d+\. \d+\. \d+			
RSW2006A	S6:LOCAL	OK			
RSW2006A	S6:RESET	OK			
RSW2006A	S6:INT_RELAY_A_NO	NO			
RSW2006A	S6:INT_RELAY_B_NO	NO			
RSW2006A	S6:INT_RELAY_A_NC	NC			
RSW2006A	S6:INT_RELAY_B_NC	NC			
RSW2006A	S6:EXT_RELAY_1_ACTIVE	ACTIVE			
RSW2006A	S6:EXT_RELAY_2_ACTIVE	ACTIVE			
RSW2006A	S6:EXT_RELAY_3_ACTIVE	ACTIVE			
RSW2006A	S6:EXT_RELAY_4_ACTIVE	ACTIVE			
RSW2006A	S6:EXT_RELAY_1_IDLE	IDLE			
RSW2006A	S6:EXT_RELAY_2_IDLE	IDLE			
RSW2006A	S6:EXT_RELAY_3_IDLE	IDLE			
RSW2006A	S6:EXT_RELAY_4_IDLE	IDLE			
Alle commando's voor de RadiSense LPS1001A kaart					
LPS1001A	*IDN?	RadiSense LPS1001A, Version \d+\. \d+\. \d+	10		
RSS1004;RSS1006;RSS1018	ZERO	:Z	3000		
RSS1004;RSS1006;RSS1018	U1	:U1	50		
RSS1004;RSS1006;RSS1018	R1	:R1	50		
RSS1004;RSS1006;RSS1018	R2	:R2	50		
RSS1004;RSS1006;RSS1018	R3	:R3	50		
RSS1004;RSS1006;RSS1018	R4	:R4	50		
RSS1004;RSS1006;RSS1018	R5	ERROR 4	50		
RSS1004;RSS1006;RSS1018	ADDD	:A	50		
RSS1004;RSS1006;RSS1018	D1	:D00\,00 V	50		
RSS1004;RSS1006;RSS1018	D2	:D00\,00 V 000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	:D0\,00;0\,00;0\,00 V	50		
RSS1004;RSS1006;RSS1018	AEDD	:A	50		
RSS1004;RSS1006;RSS1018	D1	:D\d\d\,\d\d V	50		
RSS1004;RSS1006;RSS1018	D2	:D\d\d\,\d\d V ((([0-1]\d\d))([2[0-5][2]]))NNEDD	50		
RSS1004;RSS1006;RSS1018	D3	:D\d\,\d\d;0\,00;0\,00 V	50		
RSS1004;RSS1006;RSS1018	ADED	:A	50		
RSS1004;RSS1006;RSS1018	D1	:D\d\d\,\d\d V	50		
RSS1004;RSS1006;RSS1018	D2	:D\d\d\,\d\d V ((([0-1]\d\d))([2[0-5][2]]))NNDED	50		
RSS1004;RSS1006;RSS1018	D3	:D0\,00;\d\,\d\d;0\,00 V	50		
RSS1004;RSS1006;RSS1018	ADDE	:A	50		
RSS1004;RSS1006;RSS1018	D1	:D\d\d\,\d\d V	50		
RSS1004;RSS1006;RSS1018	D2	:D\d\d\,\d\d V ((([0-1]\d\d))([2[0-5][2]]))NNDEE	50		

Bijlage I - Commando Database

RSS1004;RSS1006;RSS1018	D3	.D0.00;0.00;\d.\d\d V	50		
RSS1004;RSS1006;RSS1018	ADDD	.A	50		
RSS1004;RSS1006;RSS1018	U2	.U2	50		
RSS1004;RSS1006;RSS1018	D1	.D0\,000mW2	50		
RSS1004;RSS1006;RSS1018	D2	.D0\,000mW2000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	.D0\,000;0\,000;0\,000mW2	50		
RSS1004;RSS1006;RSS1018	ADDD	.A	50		
RSS1004;RSS1006;RSS1018	U3	.U3	50		
RSS1004;RSS1006;RSS1018	D1	.D00\,00 V2	50		
RSS1004;RSS1006;RSS1018	D2	.D00\,00 V2000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	.D0\,00;0\,00;0\,00 V2	50		
RSS1004;RSS1006;RSS1018	ADDD	.A	50		
RSS1004;RSS1006;RSS1018	UN	.U1	50		
RSS1004;RSS1006;RSS1018	D1	.D00\,00 V	50		
RSS1004;RSS1006;RSS1018	D2	.D00\,00 V 000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	.D0\,00;0\,00;0\,00 V	50		
RSS1004;RSS1006;RSS1018	UN	.U2	50		
RSS1004;RSS1006;RSS1018	D1	.D0\,000mW2	50		
RSS1004;RSS1006;RSS1018	D2	.D0\,000mW2000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	.D0\,000;0\,000;0\,000mW2	50		
RSS1004;RSS1006;RSS1018	UN	.U3	50		
RSS1004;RSS1006;RSS1018	D1	.D00\,00 V2	50		
RSS1004;RSS1006;RSS1018	D2	.D00\,00 V2000NNDDD	50		
RSS1004;RSS1006;RSS1018	D3	.D0\,00;0\,00;0\,00 V2	50		
RSS1004;RSS1006;RSS1018	D4	ERROR 4	50		
RSS1004;RSS1006;RSS1018	FILTER 1	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 1	50		
RSS1004;RSS1006;RSS1018	FILTER 2	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 2	50		
RSS1004;RSS1006;RSS1018	FILTER 3	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 3	50		
RSS1004;RSS1006;RSS1018	FILTER 4	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 4	50		
RSS1004;RSS1006;RSS1018	FILTER 5	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 5	50		
RSS1004;RSS1006;RSS1018	FILTER 6	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER 6	50		
RSS1004;RSS1006;RSS1018	FILTER DYN	OK	10		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER DYN	50		
RSS1004;RSS1006;RSS1018	FILTER 7	ERROR 5	50		
RSS1004;RSS1006;RSS1018	FILTER?	FILTER DYN	50		
RSS1004;RSS1006;RSS1018	B	.B\d\d\d.\d\d	50		
RSS1004;RSS1006;RSS1018	C1	.C	50		
RSS1004;RSS1006;RSS1018	C2	.C	50		
RSS1004;RSS1006;RSS1018	C3	.C	50		
RSS1004;RSS1006;RSS1018	C4	.C	50		
RSS1004;RSS1006;RSS1018	C5	.C	50		
RSS1004;RSS1006;RSS1018	C6	.C	50		
RSS1004;RSS1006;RSS1018	C7	ERROR 4	50		
RSS1004;RSS1006;RSS1018	TC	.T\d\d\d\d\d\.	50		
RSS1004;RSS1006;RSS1018	TF	.T\d\d\d\d\d\.	50		
RSS1004;RSS1006;RSS1018	TQ	ERROR 4	50		
RSS1004;RSS1006;RSS1018	S5	.S	50		
RSS1004;RSS1006;RSS1018	S	ERROR 4	50		
RSS1004;RSS1006;RSS1018	AE	ERROR 4	50		
	EPH1001A				
EPI1001A	*IDN?	D\,A\,R\,E\ V, RadiSense EPI1001A, \d+\. \d+\. \d+	10		
7007-001	*IDN?	ETS-Lindgren, EMSense 7007-001, \d+\. \d+\. \d+	10		
EPI1001A;7007-001	LOCAL	OK	100		
EPI1001A;7007-001	RESET	OK	100		
EPI1001A;7007-001	ID_NUMBER?	\d+\. \d+\. (7)	10		
HI-6005	B	.B\d\d\d.\d\d	10		
HI-6005	D3	.D(\.\d)(5))(3)(NIF)	10		
HI-6005	D5	.D(\.\d)(5))(4)(NIF)	10		
HI-6005	I	.i6005.(26)(NIF)	10		
HI-6005	TC	.T00[0-4]\d\.	10		
HI-6005	TF	.T0[0-1]\d\d\.	10		

## Bijlage I - Commando Database

Bijlage I - Commando Database

RGN6000B:7003-001	TEMPERATURE:MAXIMUM?	:TEMPERATURE:MAXIMUM 60\,0 / :TEMPERATURE:MAXIMUM 0\,0	10		
RGN6000B:7003-001	TEMP:MAX?	:TEMP:MAX 60\,0 / :TEMP:MAX 0\,0	10		
RGN6000B:7003-001	STATUS:QUESTIONABLE:INTEGRITY:HARDWARE?	:STATUS:QUESTIONABLE:INTEGRITY:HARDWARE [0-1]	10		
RGN6000B:7003-001	STAT:QUES:INT:HARD?	:STAT:QUES:INT:HARD [0-1]	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
			10		
Frequency testing OUTPUT 1					
RGN6000B:7003-001	OUTPUT1:STATE ON				
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	FREQ:MIN?	:FREQ:MIN 800000000	10		
RGN6000B:7003-001	FREQ:MAX?	:FREQ:MAX 60000000000	10		
RGN6000B:7003-001	FREQUENCY:MINIMUM?	:FREQUENCY:MINIMUM 800000000	10		
RGN6000B:7003-001	FREQUENCY:MAXIMUM?	:FREQUENCY:MAXIMUM 60000000000	10		
RGN6000B:7003-001	FREQ 800000000		100		
RGN6000B:7003-001	FREQ?	:FREQ 800000000	10		
RGN6000B:7003-001	FREQ 127000 KHZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 1270000000	10		
RGN6000B:7003-001	FREQ 1240000000 HZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 1240000000	10		
RGN6000B:7003-001	FREQ 6000 MHZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 60000000000	10		
RGN6000B:7003-001	FREQ 0.129 GHZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 1290000000	10		
RGN6000B:7003-001	FREQUENCY 80000000		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 800000000	10		
RGN6000B:7003-001	FREQUENCY 1240000000 HZ		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 1240000000	10		
RGN6000B:7003-001	FREQUENCY 127000 KHZ		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 1270000000	10		
RGN6000B:7003-001	FREQUENCY 6000 MHZ		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 60000000000	10		
RGN6000B:7003-001	FREQUENCY 0.129 GHZ		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 1290000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	FREQUENCY 6.1 GHZ		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 1290000000	10		
Frequency testing OUTPUT 1					
RGN6000B:7003-001	OUTPUT2:STATE ON				
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 2	10		
RGN6000B:7003-001	FREQ:MIN?	:FREQ:MIN 9000	10		
RGN6000B:7003-001	FREQ:MAX?	:FREQ:MAX 2300000000	10		
RGN6000B:7003-001	FREQUENCY:MINIMUM?	:FREQUENCY:MINIMUM 9000	10		
RGN6000B:7003-001	FREQUENCY:MAXIMUM?	:FREQUENCY:MAXIMUM 2300000000	10		
RGN6000B:7003-001	FREQ 9000		100		
RGN6000B:7003-001	FREQ?	:FREQ 9000	10		
RGN6000B:7003-001	FREQ 2300000000		100		
RGN6000B:7003-001	FREQ?	:FREQ 2300000000	10		
RGN6000B:7003-001	FREQ 8999		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	FREQ?	:FREQ 2300000000	10		
RGN6000B:7003-001	FREQ 2300000001		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	FREQ?	:FREQ 2300000000	10		
Step testing					
	UP TESTING				
RGN6000B:7003-001	FREQ 1290000000		100		
RGN6000B:7003-001	FREQ:STEP 1000000		100		
RGN6000B:7003-001	FREQ:STEP?	:FREQ:STEP 1000000	10		
RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ:STEP 2000 KHZ		100		
RGN6000B:7003-001	FREQ:STEP?	:FREQ:STEP 2000000	10		
RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ:STEP 3 MHZ		100		



RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 3000000	10		
RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ.STEP 0.004 GHZ		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 4000000	10		
RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ?	.FREQ 139000000	10		
RGN6000B:7003-001	SYSTEM.ERROR?	.SYSTEM.ERROR 0, "No error"	10		
RGN6000B:7003-001	FREQ.STEP 0.004 GHZ		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 4000000	10		
			10		
	<b>DOWN TESTING</b>				
RGN6000B:7003-001	FREQ.STEP 1000000		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 1000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ.STEP 2000 KHZ		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 2000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ.STEP 3 MHZ		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 3000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ.STEP 0.004 GHZ		100		
RGN6000B:7003-001	FREQ.STEP?	.FREQ.STEP 4000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ?	.FREQ 129000000	10		
RGN6000B:7003-001	SYSTEM.ERROR?	.SYSTEM.ERROR 0, "No error"	10		
			10		
	<b>UP TESTING</b>				
RGN6000B:7003-001	FREQUENCY:STEP 1000000		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 1000000	10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY:STEP 2000 KHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 2000000	10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY:STEP 3 MHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 3000000	10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY:STEP 0.004 GHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 4000000	10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY?	.FREQUENCY 139000000	10		
RGN6000B:7003-001	SYSTEM.ERROR?	.SYSTEM.ERROR 0, "No error"	10		
			10		
	<b>DOWN TESTING</b>				
RGN6000B:7003-001	FREQUENCY:STEP 1000000		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 1000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY:STEP 2000 KHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 2000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY:STEP 3 MHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 3000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY:STEP 0.004 GHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP?	.FREQUENCY:STEP 4000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY?	.FREQUENCY 129000000	10		
RGN6000B:7003-001	SYSTEM.ERROR?	.SYSTEM.ERROR 0, "No error"	10		
			10		
	<b>START STOP TESTING Instellen</b>				
RGN6000B:7003-001	FREQ 90 MHZ		100		
RGN6000B:7003-001	FREQ.STEP 2 MHZ		100		
RGN6000B:7003-001	FREQ.STAR 88 MHZ		100		
RGN6000B:7003-001	FREQ.STAR?	.FREQ.STAR 88000000	10		
RGN6000B:7003-001	FREQ.STOP 92 MHZ		100		
RGN6000B:7003-001	FREQ.STOP?	.FREQ.STOP 92000000	10		
RGN6000B:7003-001	SYSTEM.ERROR?	.SYSTEM.ERROR 0, "No error"	10		
	<b>Functionaliteit testen</b>		10		
RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ?	.FREQ 92000000	10		

## Bijlage I - Commando Database

RGN6000B:7003-001	FREQ UP		100		
RGN6000B:7003-001	FREQ?	:FREQ 88000000	10		
RGN6000B:7003-001	FREQ 90 MHZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 90000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ?	:FREQ 88000000	10		
RGN6000B:7003-001	FREQ DOWN		100		
RGN6000B:7003-001	FREQ?	:FREQ 92000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	FREQ 95 MHZ		100		
RGN6000B:7003-001	FREQ?	:FREQ 92000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
	<b>Instellen</b>		10		
RGN6000B:7003-001	FREQUENCY 90 MHZ		100		
RGN6000B:7003-001	FREQUENCY:STEP 2 MHZ		100		
RGN6000B:7003-001	FREQUENCY:START 88 MHZ		100		
RGN6000B:7003-001	FREQUENCY:START?	:FREQUENCY:START 88000000	10		
RGN6000B:7003-001	FREQUENCY:STOP 92 MHZ		100		
RGN6000B:7003-001	FREQUENCY:STOP?	:FREQUENCY:STOP 92000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
	<b>Functionaliteit testen</b>		10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 92000000	10		
RGN6000B:7003-001	FREQUENCY UP		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 88000000	10		
RGN6000B:7003-001	FREQUENCY 90 MHZ		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 90000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 88000000	10		
RGN6000B:7003-001	FREQUENCY DOWN		100		
RGN6000B:7003-001	FREQUENCY?	:FREQUENCY 92000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
	<b>POWER TEST</b>				
RGN6000B:7003-001	POW -50		100		
RGN6000B:7003-001	POW?	:POW -50\,0	10		
RGN6000B:7003-001	POW -70 DBM		100		
RGN6000B:7003-001	POW?	:POW -70\,0	10		
RGN6000B:7003-001	POW 0.5		100		
RGN6000B:7003-001	POW?	:POW 0\,5	10		
RGN6000B:7003-001	POW 10		100		
RGN6000B:7003-001	POW?	:POW 10\,0	10		
RGN6000B:7003-001	POW 10 DBM		100		
RGN6000B:7003-001	POW?	:POW 10\,0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	POWER -50		100		
RGN6000B:7003-001	POWER?	:POWER -50\,0	10		
RGN6000B:7003-001	POWER -70 DBM		100		
RGN6000B:7003-001	POWER?	:POWER -70\,0	10		
RGN6000B:7003-001	POWER 0.5		100		
RGN6000B:7003-001	POWER?	:POWER 0\,5	10		
RGN6000B:7003-001	POWER 10		100		
RGN6000B:7003-001	POWER?	:POWER 10\,0	10		
RGN6000B:7003-001	POWER 10 DBM		100		
RGN6000B:7003-001	POWER?	:POWER 10\,0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	POWER -71 DBM		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	POWER?	:POWER 10\,0	10		
RGN6000B:7003-001	POWER 11 DBM		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	POWER?	:POWER 10\,0	10		
RGN6000B:7003-001	OUTP2:STAT OFF		100		
RGN6000B:7003-001	OUTP1:STAT ON		100		
RGN6000B:7003-001	OUTP1:STAT?	:OUTP1:STAT ON	10		
RGN6000B:7003-001	OUTP2:STAT?	:OUTP2:STAT OFF	10		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	OUTP2:STAT ON		100		

RGN6000B:7003-001	OUTP1:STAT OFF		100		
RGN6000B:7003-001	OUTP1:STAT?	:OUTP1:STAT OFF	10		
RGN6000B:7003-001	OUTP2:STAT?	:OUTP2:STAT ON	10		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	OUTP2:STAT OFF		100		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 2	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	OUTPUT2:STATE OFF		100		
RGN6000B:7003-001	OUTPUT1:STATE ON		100		
RGN6000B:7003-001	OUTPUT1:STATE?	:OUTPUT1:STATE ON	10		
RGN6000B:7003-001	OUTPUT2:STATE?	:OUTPUT2:STATE OFF	10		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	OUTPUT2:STATE ON		100		
RGN6000B:7003-001	OUTPUT1:STATE OFF		100		
RGN6000B:7003-001	OUTPUT1:STATE?	:OUTPUT1:STATE OFF	10		
RGN6000B:7003-001	OUTPUT2:STATE?	:OUTPUT2:STATE ON	10		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	OUTPUT2:STATE OFF		100		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 2	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	OUTPUT1:STATE ab		100		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -104, "Data type error"	10		
	<b>AM FREQ</b>				
RGN6000B:7003-001	AM:INT:FREQ 10		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 10	10		
RGN6000B:7003-001	AM:INT:FREQ 100 HZ		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 100	10		
RGN6000B:7003-001	AM:INT:FREQ 1 KHZ		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 1000	10		
RGN6000B:7003-001	AM:INT:FREQ 0.1 MHZ		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 100000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY 10		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 10	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY 100 HZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 100	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY 1 KHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 1000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY 0.1 MHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 100000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001					
RGN6000B:7003-001	AM:INT:FREQ 9		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 100000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	AM:INT:FREQ 1001 KHZ		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 100000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
	<b>AM STEP</b>				
RGN6000B:7003-001	AM:INT:FREQ 10		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP 10		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 10	10		
RGN6000B:7003-001	AM:INT:FREQ UP		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP 1 KHZ		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 1000	10		
RGN6000B:7003-001	AM:INT:FREQ UP		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP 0.01MHZ		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 10000	10		
RGN6000B:7003-001	AM:INT:FREQ UP		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 11020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001					
RGN6000B:7003-001	AM:INT:FREQ:STEP 10		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 10	10		
RGN6000B:7003-001	AM:INT:FREQ DOWN		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP 1 KHZ		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 1000	10		

## Bijlage I - Commando Database

RGN6000B:7003-001	AM:INT:FREQ DOWN		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP 0.01MHZ		100		
RGN6000B:7003-001	AM:INT:FREQ:STEP?	:AM:INT:FREQ:STEP 10000	10		
RGN6000B:7003-001	AM:INT:FREQ DOWN		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 10	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY 10		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 10		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 10	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY UP		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 1 KHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 1000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY UP		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 0.01MHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 10000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY UP		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 11020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 10		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 10	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY DOWN		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 1 KHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 1000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY DOWN		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP 0.01MHZ		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY:STEP?	:AM:INTERNAL:FREQUENCY:STEP 10000	10		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY DOWN		100		
RGN6000B:7003-001	AM:INTERNAL:FREQUENCY?	:AM:INTERNAL:FREQUENCY 10	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM 0		100		
RGN6000B:7003-001	AM?	:AM 0\0	10		
RGN6000B:7003-001	AM 54.3		100		
RGN6000B:7003-001	AM?	:AM 54\3	10		
RGN6000B:7003-001	AM 54.5 %		100		
RGN6000B:7003-001	AM?	:AM 54\5	10		
RGN6000B:7003-001	AM 100		100		
RGN6000B:7003-001	AM?	:AM 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM 101		100		
RGN6000B:7003-001	AM?	:AM 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	AM -1		100		
RGN6000B:7003-001	AM?	:AM 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -120, "Numeric data error"	10		
RGN6000B:7003-001	AM:DEPTH 0		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 0\0	10		
RGN6000B:7003-001	AM:DEPTH 54.3		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 54\3	10		
RGN6000B:7003-001	AM:DEPTH 54.5 %		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 54\5	10		
RGN6000B:7003-001	AM:DEPTH 100		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM:DEPTH 101		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	AM:DEPTH -1		100		
RGN6000B:7003-001	AM:DEPTH?	:AM:DEPTH 100\0	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -120, "Numeric data error"	10		
RGN6000B:7003-001	AM:STAT OFF		100		
RGN6000B:7003-001	AM:STAT?	:AM:STAT OFF	10		
RGN6000B:7003-001	AM:STAT ON		100		
RGN6000B:7003-001	AM:STAT?	:AM:STAT ON	10		
RGN6000B:7003-001	AM:STATE OFF		100		

RGN6000B:7003-001	AM:STATE?	:AM:STATE OFF	10		
RGN6000B:7003-001	AM:STATE ON		100		
RGN6000B:7003-001	AM:STATE?	:AM:STATE ON	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	AM:STATE ab		100		
RGN6000B:7003-001	AM:STATE?	:AM:STATE ON	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -104, "Data type error"	10		
RGN6000B:7003-001	PULS:WIDT 1		100		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 1\,00000000	10		
RGN6000B:7003-001	PULS:WIDT 0.1 S		100		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 0\,10000000	10		
RGN6000B:7003-001	PULS:WIDT 1 MS		100		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 0\,00100000	10		
RGN6000B:7003-001	PULS:WIDT 3 US		100		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 0\,00000300	10		
RGN6000B:7003-001	PULS:WIDT 200 NS		10		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 0\,00000020	10		
RGN6000B:7003-001	PULS:WIDT 1		100		
RGN6000B:7003-001	PULS:WIDT 150 NS		10		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 0\,00000020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULSE:WIDTH 1		100		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 1\,00000000	10		
RGN6000B:7003-001	PULSE:WIDTH 0.1 S		100		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0\,10000000	10		
RGN6000B:7003-001	PULSE:WIDTH 1 MS		100		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0\,00100000	10		
RGN6000B:7003-001	PULSE:WIDTH 3 US		100		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0\,00000300	10		
RGN6000B:7003-001	PULSE:WIDTH 200 NS		10		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0\,00000020	10		
RGN6000B:7003-001	PULSE:WIDTH 1		100		
RGN6000B:7003-001	PULSE:WIDTH 150 NS		10		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0\,00000020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULS:WIDT 1		100		
RGN6000B:7003-001	PULS:WIDT 101		10		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 1\,00000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULS:WIDT 149 NS		10		
RGN6000B:7003-001	PULS:WIDT?	:PULS:WIDT 1\,00000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULS:DEL 1		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 1\,00000000	10		
RGN6000B:7003-001	PULS:DEL 0.1 S		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 0\,10000000	10		
RGN6000B:7003-001	PULS:DEL 1 MS		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 0\,00100000	10		
RGN6000B:7003-001	PULS:DEL 3 US		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 0\,00000300	10		
RGN6000B:7003-001	PULS:DEL 200 NS		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 0\,00000020	10		
RGN6000B:7003-001	PULS:DEL 1		100		
RGN6000B:7003-001	PULS:DEL 150 NS		100		
RGN6000B:7003-001	PULS:DEL?	:PULS:DEL 0\,00000020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULSE:DELAY 1		100		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 1\,00000000	10		
RGN6000B:7003-001	PULSE:DELAY 0.1 S		100		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0\,10000000	10		
RGN6000B:7003-001	PULSE:DELAY 1 MS		100		

Bijlage I - Commando Database

RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0.00100000	10		
RGN6000B:7003-001	PULSE:DELAY 3 US		100		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0.00000300	10		
RGN6000B:7003-001	PULSE:DELAY 200 NS		10		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0.00000020	10		
RGN6000B:7003-001	PULSE:DELAY 1		100		
RGN6000B:7003-001	PULSE:DELAY 150 NS		10		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0.00000020	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULSE:DELAY 1		100		
RGN6000B:7003-001	PULSE:DELAY 101		10		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 1.00000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULSE:DELAY 149 NS		10		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 1.00000000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
	<b>GATED PM</b>				
RGN6000B:7003-001	PULM:STAT OFF		100		
RGN6000B:7003-001	PULM:STAT?	:PULM:STAT OFF	10		
RGN6000B:7003-001	PULM:STAT ON		100		
RGN6000B:7003-001	PULM:STAT?	:PULM:STAT ON	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:STATE OFF		100		
RGN6000B:7003-001	PULM:STATE?	:PULM:STATE OFF	10		
RGN6000B:7003-001	PULM:STATE ON		100		
RGN6000B:7003-001	PULM:STATE?	:PULM:STATE ON	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:STAT AB		100		
RGN6000B:7003-001	PULM:STAT?	:PULM:STAT ON	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -104, "Data type error"	10		
RGN6000B:7003-001	AM:INT:FREQ 300		100		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 300	10		
RGN6000B:7003-001	PULSE:WIDTH 3 US		100		
RGN6000B:7003-001	PULSE:WIDTH?	:PULSE:WIDTH 0.00000300	100		
RGN6000B:7003-001	PULSE:DELAY 3.330 MS		100		
RGN6000B:7003-001	PULSE:DELAY?	:PULSE:DELAY 0.00333000	10		
	<b>Number of pulses</b>				
RGN6000B:7003-001	PULM:BURST:NUM 1		100		
RGN6000B:7003-001	PULM:BURST:NUM?	:PULM:BURST:NUM 1	10		
RGN6000B:7003-001	PULM:BURST:NUM 10		100		
RGN6000B:7003-001	SOUR:PULM:BURST:NUM?	:SOUR:PULM:BURST:NUM 10	10		
RGN6000B:7003-001	SOUR:PULM:BURST:NUM 298		100		
RGN6000B:7003-001	PULM:BURST:NUM?	:PULM:BURST:NUM 298	10		
RGN6000B:7003-001	SOUR:PULM:BURST:NUM 250		100		
RGN6000B:7003-001	SOUR:PULM:BURST:NUM?	:SOUR:PULM:BURST:NUM 250	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:BURST:NUMBER 1		100		
RGN6000B:7003-001	PULM:BURST:NUMBER?	:PULM:BURST:NUMBER 1	10		
RGN6000B:7003-001	PULM:BURST:NUMBER 10		100		
RGN6000B:7003-001	SOURCE:PULM:BURST:NUMBER?	:SOURCE:PULM:BURST:NUMBER 10	10		
RGN6000B:7003-001	SOURCE:PULM:BURST:NUMBER 298		100		
RGN6000B:7003-001	PULM:BURST:NUMBER?	:PULM:BURST:NUMBER 298	10		
RGN6000B:7003-001	SOURCE:PULM:BURST:NUMBER 250		100		
RGN6000B:7003-001	SOURCE:PULM:BURST:NUMBER?	:SOURCE:PULM:BURST:NUMBER 250	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:BURST:NUMBER 300		100		
RGN6000B:7003-001	PULM:BURST:NUMBER?	:PULM:BURST:NUMBER 250	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULM:BURST:NUMBER 0		100		
RGN6000B:7003-001	PULM:BURST:NUMBER?	:PULM:BURST:NUMBER 250	10		

RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
	PERIOD				
RGN6000B:7003-001	PULM:BURST:PER 1		100		
RGN6000B:7003-001	SOUR:PULM:BURST:PER?	:SOUR:PULM:BURST:PER 1\,000	10		
RGN6000B:7003-001	PULM:BURST:PER 0.5 S		100		
RGN6000B:7003-001	PULM:BURST:PER?	:PULM:BURST:PER 0\,500	10		
RGN6000B:7003-001	SOUR:PULM:BURST:PER 100 MS		100		
RGN6000B:7003-001	PULM:BURST:PER?	:PULM:BURST:PER 0\,100	10		
RGN6000B:7003-001	SOUR:PULM:BURST:PER 2000 US		100		
RGN6000B:7003-001	SOUR:PULM:BURST:PER?	:SOUR:PULM:BURST:PER 0\,002	10		
RGN6000B:7003-001	PULM:BURST:PERIOD 1		100		
RGN6000B:7003-001	SOURCE:PULM:BURST:PERIOD?	:SOURCE:PULM:BURST:PERIOD 1\,000	10		
RGN6000B:7003-001	PULM:BURST:PERIOD 0.5 S		100		
RGN6000B:7003-001	PULM:BURST:PERIOD?	:PULM:BURST:PERIOD 0\,500	10		
RGN6000B:7003-001	SOURCE:PULM:BURST:PERIOD 100 MS		100		
RGN6000B:7003-001	PULM:BURST:PERIOD?	:PULM:BURST:PERIOD 0\,100	10		
RGN6000B:7003-001	SOURCE:PULM:BURST:PERIOD 2000 US		100		
RGN6000B:7003-001	SOURCE:PULM:BURST:PERIOD?	:SOURCE:PULM:BURST:PERIOD 0\,002	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:BURST:PER 1		100		
RGN6000B:7003-001	PULM:BURST:PER 2		100		
RGN6000B:7003-001	PULM:BURST:PER?	:PULM:BURST:PER 1\,000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULM:BURST:PER 1999 US		100		
RGN6000B:7003-001	PULM:BURST:PER?	:PULM:BURST:PER 1\,000	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -222, "Data out of range"	10		
RGN6000B:7003-001	PULM:BURST:PER 1 S				
RGN6000B:7003-001	PULS:DEL 3 MS				
RGN6000B:7003-001	PULS:WIDT 1 MS				
RGN6000B:7003-001	PULM:BURST:NUM 248				
RGN6000B:7003-001	PULM:BURST:STAT ON		100		
RGN6000B:7003-001	PULM:BURST:STAT?	:PULM:BURST:STAT ON	10		
RGN6000B:7003-001	PULM:BURST:STAT OFF		100		
RGN6000B:7003-001	SOUR:PULM:BURST:STAT?	:SOUR:PULM:BURST:STAT OFF	10		
RGN6000B:7003-001	SOUR:PULM:BURST:STAT ON		100		
RGN6000B:7003-001	PULM:BURST:STAT?	:PULM:BURST:STAT ON	10		
RGN6000B:7003-001	SOUR:PULM:BURST:STAT OFF		100		
RGN6000B:7003-001	SOUR:PULM:BURST:STAT?	:SOUR:PULM:BURST:STAT OFF	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	PULM:BURST:NUM 250		100		
RGN6000B:7003-001	PULM:BURST:STAT?	:PULM:BURST:STAT OFF	10		
RGN6000B:7003-001	PULM:BURST:STAT ON		100		
RGN6000B:7003-001	PULM:BURST:STAT?	:PULM:BURST:STAT OFF	10		
RGN6000B:7003-001	PULM:BURST:STAT AB		100		
RGN6000B:7003-001	PULM:BURST:STAT?	:PULM:BURST:STAT OFF	10		
RGN6000B:7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR -104, "Data type error"	10		
RGN6000B:7003-001	TEMP?	:TEMP \d\d\d\d / :TEMP \d\d\d\d	100		
RGN6000B:7003-001	TEMPERATURE?	:TEMPERATURE \d\d\d\d / :TEMPERATURE \d\d\d\d	100		
RGN6000B: 7003-001	FREQ 1				
RGN6000B:7003-001	*RST		200		
RGN6000B: 7003-001	SYSTEM:ERROR?	:SYSTEM:ERROR 0, "No error"	10		
RGN6000B:7003-001	FREQ?	:FREQ 125000000	10		
RGN6000B:7003-001	FREQ:START?	:FREQ:START 800000000	10		
RGN6000B:7003-001	FREQ:STOP?	:FREQ:STOP 80000000000	10		
RGN6000B:7003-001	FREQ:STEP?	:FREQ:STEP 800000000	10		
RGN6000B:7003-001	POW?	:POW -30.0	10		
RGN6000B:7003-001	OUTP1:STATE?	:OUTP1:STATE OFF	10		
RGN6000B:7003-001	OUTP2:STATE?	:OUTP2:STATE OFF	10		
RGN6000B:7003-001	OUTPUT:SELECTED?	:OUTPUT:SELECTED 1	10		
RGN6000B:7003-001	AM:INT:FREQ?	:AM:INT:FREQ 1000	10		

## Bijlage I - Commando Database

[illegible]



RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_A?	NC	10		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B?	NO	10		
RSW1024;7001-002;7001-012	INT_RELAY_C?	NO	10		
RSW1024;7001-002;7001-012	INT_RELAY_D?	NO	10		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B_NC	NC	100		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_A?	NC	10		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_C?	NO	10		
RSW1024;7001-002;7001-012	INT_RELAY_D?	NO	10		
RSW1024;7001-002;7001-012	INT_RELAY_C_NC	NC	100		
RSW1024;7001-002;7001-012	INT_RELAY_A?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_B?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_C?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_D?	NO	10		
RSW1024;7001-002;7001-012	INT_RELAY_D_NC	NC	100		
RSW1024;7001-002;7001-012	INT_RELAY_A?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_B?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_C?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_D?	NC	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_A_AA	ERROR 1(+)?	10		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B_AA	ERROR 1(+)?	10		
RSW1024;7001-002;7001-012	INT_RELAY_C_AA	ERROR 1(+)?	10		
RSW1024;7001-002;7001-012	INT_RELAY_D_AA	ERROR 1(+)?	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_E_NC	ERROR 208	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_F?	ERROR 208	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	EXT_RELAY_A?	ERROR 1(+)?	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	EXT_RELAY_1?	ERROR 1(+)?	10		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_A_NO	NO	100		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B_NO	NO	100		
RSW1024;7001-002;7001-012	INT_RELAY_C_NO	NO	100		
RSW1024;7001-002;7001-012	INT_RELAY_D_NO	NO	100		
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	REBOOT SYSTEM	OK	3000		4.0.0
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	REBOOT SYSTEM	OK	3000	4.0.0	
RSW1024;RSW1022;RSW1021;7001-002;7001-012;7001-001;7001-011;7001-021	INT_RELAY_A?	NC	10		
RSW1024;RSW1022;7001-002;7001-012;7001-001;7001-011	INT_RELAY_B?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_C?	NC	10		
RSW1024;7001-002;7001-012	INT_RELAY_D?	NC	10		
	Alle commando's voor de RSW1062S				
RSW1061S	*IDN?	DLA\RL\E\I\, RadiSwitch RSW1061S, \d+\\.\\d+\\.\\d+	10		
7001-005	*IDN?	ETS-Lindgren, EMSwitch 7001-005, \d+\\.\\d+\\.\\d+	10		
RSW1062S	*IDN?	DLA\RL\E\I\, RadiSwitch RSW1062S, \d+\\.\\d+\\.\\d+	10		
7001-003	*IDN?	ETS-Lindgren, EMSwitch 7001-003, \d+\\.\\d+\\.\\d+	10		
RSW1061K	*IDN?	DLA\RL\E\I\, RadiSwitch RSW1061K, \d+\\.\\d+\\.\\d+	10		
7001-015	*IDN?	ETS-Lindgren, EMSwitch 7001-015, \d+\\.\\d+\\.\\d+	10		
RSW1062K	*IDN?	DLA\RL\E\I\, RadiSwitch RSW1062K, \d+\\.\\d+\\.\\d+	10		
7001-013	*IDN?	ETS-Lindgren, EMSwitch 7001-013, \d+\\.\\d+\\.\\d+	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	RESET	OK	100		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	REBOOT SYSTEM	OK	3000		4.0.0
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	REBOOT SYSTEM	OK	3000	4.0.0	
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	ID_NUMBER?	\d+\\.\\d+\\(7)	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	VERSION_SW?	\d+\\.\\d+\\(2)	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	VERSION_HW?	\d+\\.\\d+\\(0,2)	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_0	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_6	OK	100		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	0	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	6	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_1	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_5	OK	100		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	1	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	5	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_2	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_4	OK	100		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	2	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	4	10		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_3	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_3	OK	100		
RSW1062;RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	3	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	3	10		

## Bijlage I - Commando Database

RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_4	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_2	OK	100		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	4	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	2	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_5	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_1	OK	100		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	5	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	1	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_6	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B_0	OK	100		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	6	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	0	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A_9	ERROR_208	10		
RSW1062;7001-003;7001-013	INT_RELAY_B_7	ERROR_208	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_C_2	ERROR 1.*	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	EXT_RELAY_A_2	ERROR 1.*	10		
RSW1062;7001-003;7001-013	INT_RELAY_B_3	OK	100		
RSW1062;7001-003;7001-013	INT_RELAY_B?	3	10		
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	REBOOT SYSTEM	OK	3000		4.0.0
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	IREBOOT SYSTEM	OK	3000	4.0.0	
RSW1062:RSW1061;7001-003;7001-005;7001-013;7001-015	INT_RELAY_A?	0	10		
RSW1062;7001-003;7001-013	INT_RELAY_B?	0	10		
RLK1003A	RadiLink 1003A				
RLK1003A	*IDN?	D:\A\R\E\W, RadiLink RLK1003A, \d\.\d\.	10		
RLK1003A	RESET	OK	100		
RLK1003A	ID_NUMBER?	\d+(\.\d+)?(Z)	10		
RLK1003A	TEMPERATURE?	(\d\d)(\d\d)(\d\d)	10		
RadiLinkLaser	STARTLASER	OK	100		
RadiLinkLaser	STATUSLASER?	2	10		
RadiLinkLaser	STOPLASER	OK	100		
RadiLinkLaser	STATUSLASER?	1	10		
RadiLinkLaser	STARTLASER	OK	100		
RadiLinkLaser	CLEAR	OK	100		
RadiLinkLaser	STATUSLASER?	1	10		
RadiLinkLaser	STARTLASER	OK	100		
RadiLinkLaser	N1*IDN?	RadiLink Laser version 1\2\4	10		
RadiLinkLaser	N1ID_NUMBER?	\d+(\.\d+)?(Z)	10		
RadiLinkLaser	N1TEMPERATURE?	(\d\d)(\d\d)(\d\d)	10		
RadiLinkLaser	N1VBATRC?	\d(4)mV	10		
RadiLinkLaser	N1LASER_LEVEL?	\d(4)	10		
RadiLinkLaser	N1LASER_BIAS?	\d(3)	10		
RadiLinkLaser	N1LASER_INTENSITY?	\d\d%	10		
RadiLinkLaser	N1LASER_TEMP?	\d(3)	10		
RadiLinkLaser	CLEAR	OK	100		

## Bijlage II – Volledige broncode RadiCentre test

```
option explicit 'variables must always be declared with DIM before they can be used

Const SLOT_COUNT = 8 'maximum number of slots in RadiCentre

'Numbers used for raising customer errors (using the Err.number as error level)
Const ERRlvl_CRITICAL_GLOBAL = 1
Const ERRlvl_CRITICAL = 2
Const ERRlvl_LOG = 3
Const ERRlvl_WARNING = 4

Const ERR_VISA_EXP = -2146232828 'general error number when there is an exception in visa
Const TIME_OUT_DELAY = 5000 'time to wait after time out has occurred (in ms)

class ParentDevice
'----- private class members -----
private ViInterface
private m_device()

'-- public properties to acces members -----

Public Property Get device(ByVal index)
If index < 0 OR index > UBound(m_device) Then
Err.Raise vbObjectError + ERRlvl_CRITICAL, "get ParentDevice.Device", index & " is an invalid device index"
Exit Property
End If

'return member object device, even when it's nothing
set device = m_device(index)
End Property

Public Property Set device(ByVal index, devObj)
'parameter devObj is not const, the original object WILL change by using this function

'check parameter: Index >= 0 and <= m_device array size
If index < 0 OR index > UBound(m_device) Then
Err.Raise vbObjectError + ERRlvl_CRITICAL, "set ParentDevice.Device", index & " is an invalid device index"
Exit Property
End If
'Check parameter: devObj Type should be RadiDev object or Nothing object
If TypeName(devObj) <> "RadiDev" AND not devObj is Nothing Then
Err.Raise vbObjectError + ERRlvl_CRITICAL, "set ParentDevice.Device", TypeName(devObj) & " is no valid member device"
Exit Property
End If

Set m_device(index) = devObj 'create child device
If TypeName(devObj) = "RadiDev" Then 'If it's a device object (so not nothing), set device parameters
m_device(index).slotNr = index
Set m_device(index).ParentDevice = Me 'set reference to THIS object as parent
End If
End Property

'----- Member functions
'constructor
Private Sub Class_Initialize
set ViInterface = dotNET.DARE.ScriptUtils.CreateVISAObject 'create object for VISA connection
ReDim m_device(SLOT_COUNT) 'set the array size for member devices

'set each member device to the special object Nothing (so it can be handled as an object)
dim i
For i = 0 to UBound(m_device)
set m_device(i) = nothing 'set all device members to Nothing object
Next
End Sub

'destructor
Private Sub Class_Terminate
'terminateChilds 'destruct child members of this instance, (so their reference to THIS is gone as well)
'doesn't add anything, because when destructor is only called when child are already terminated!
Disconnect 'close the VISA session which is a member object of THIS object
End Sub

'terminate member object, so lose all child references
Public Sub terminateChilds
'the child objects of THIS object has a reference to THIS object
'therefore, when removing the instance of THIS object (by set it to nothing), it's still alive, because
'there are (unreachable) references to the object left (at the child objects)
'Therefore, first kill the child's so the local instance is the only reference to the object.

Erase m_device 'removes all elements of the member array, their destructor's are called
End Sub

'open connection with visa
Public Sub Connect(ByVal resource)
'opening a VISA session, using the (custom) .NET library DARE.VisaInterface

'check for valid parameters
If varType(resource) <> vbString Then 'check if it's a string (so not empty or something else)
```

## Bijlage II – Volledige broncode RadiCentre test

```
Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.Connect", "no valid resource name"
Exit Sub
End If

'try opening a session
dim opened 'flag to indicate if opening a session was successfully or not
opened = ViInterface.Open(resource) 'open session (result is true or false)
If not opened Then 'if opening the session has failed
    'throw critical error. When caught, script execution will stop
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.Connect", "Opening Visa Session failed"
End If

end sub

'if necessary, configure serial parameters
Public Sub ConfSerial(ByVal Baud, ByVal DataBits, ByVal StopBits)
'interface for VISA function to configure the parameters for serial connections

'check Baud parameter
If not IsNumeric(Baud) Then 'check if parameter is numeric
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "Baud-rate parameter is invalid"
Exit Sub
ElseIf Baud <= 0 OR Baud > 128000 Then 'if parameter is numeric, check the range
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "Baud-rate parameter is invalid"
Exit Sub
End If
'check DataBits parameter
If not IsNumeric(DataBits) Then 'check if parameter is numeric
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "DataBits parameter is invalid"
Exit Sub
ElseIf DataBits < 4 OR DataBits > 8 Then 'if parameter is numeric, check the range
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "DataBits parameter is invalid"
Exit Sub
End If
'check StopBits parameter
If not IsNumeric(StopBits) Then 'check if parameter is numeric
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "StopBits parameter is invalid"
Exit Sub
ElseIf StopBits < 1 OR StopBits > 2 Then 'if parameter is numeric, check the range
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.ConfSerial", "StopBits parameter is invalid"
Exit Sub
End If

'parameters are valid, configure serial parameters
ViInterface.ConfSerial Baud, DataBits, StopBits
End Sub

'close active visa connection
Public Sub Disconnect

'check if object is a valid session object (else closing is not possible)
If TypeName(ViInterface) = "DARE.VisaInterface" Then
    ViInterface.Close 'close the active session
End If
End Sub

'perform write and read in one function
Public Function Query(ByVal command)
'this function sends a command to the active VISA session, and immediately tries to read a reply

'check the command parameter
If VarType(command) <> vbString Then
    Err.Raise vbObjectError + ERR_LVL_LOG, "ParentDevice.Query", _
        "parameter 'Command' is not a valid string value, query not executed" 'exit function with error
End If
'check if there is a valid session object
If TypeName(ViInterface) <> "DARE.VisaInterface" Then
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "ParentDevice.Query", _
        "Visa connection is not opened (any more), query not possible" 'exit function with error
End If

Dim tryAgain : tryAgain = false
do
    dim answer
    On Error Resume Next
    answer = ViInterface.Query(Command&vbCrLf) 'Try query
    If Err.Number = 0 Then
        tryAgain = false
    ElseIf Err.Number = ERR_VISA_EXP AND tryAgain = false Then
        Err.Clear 'clear the current error to catch new time out
    End If
    Delay TIME_OUT_DELAY, "Connection failed, waiting for second attempt"
    answer = ViInterface.Read 'check if there is
    something to read
    If Err.Number = ERR_VISA_EXP Then 'time out, nothing to read
        tryAgain = true
    End If
    'try query again once
    End If
Else
    dim eDescription, eSource, eNumber
    eDescription = Err.Description 'copy actual error details
    eSource = Err.Source
End If
```

## Bijlage II – Volledige broncode RadiCentre test

```
eNumber = Err.Number
On Error Goto 0 'stop local error catching, so that err.raise will exit the function
Err.Raise ERRlvl_CRITICAL, "ParentDevice.Query", "Query failed, even after retry. Original error: " & vbCrLf _
    & "Number: " & eNumber & vbCrLf _
    & "Source: " & eSource & vbCrLf _
    & "Description: " & eDescription
'raise custom error and copy original details in the description
End If
On Error Goto 0
Loop while tryAgain

answer = aqString.Replace(answer,vbLf, "") 'remove LineFeed from answer, by replacing with "" (empty string)
Query = answer 'return answer
End Function

'perform write action without reading a answer
Public Sub Write(ByVal command)
'This function sends the command to the active Visa Session, without reading a reply

'check the command parameter
If VarType(command) <> vbString Then
    Err.Raise vbObjectError + ERRlvl_LOG, "ParentDevice.Query", "parameter 'Command' is not a valid string value, query
not executed"
Exit Sub
End If
'check if there is a valid session object
If TypeName(ViInterface) <> "DARE.VisaInterface" Then
    Err.Raise vbObjectError + ERRlvl_CRITICAL, "ParentDevice.Query", "Visa connection is not opened (any more), query not
possible"
End If

ViInterface.Write Command&vbCrLf 'write the command to the actual visa session
End Sub

End Class

Function GetParentDeviceInst
'it's not possible to create an object of a class which is defined in other file.
'this function is a workaround to get an object of the class in another scriptfile
Set GetParentDeviceInst = new ParentDevice
End Function

class RadiDev
'-----private class members-----
Private m_ParentDevice
Private m_Name
Private m_hwPrefix
Private m_slotNr
Private m_Version
Private m_SubDev
'-----Public properties to access members-----

Public Property Set ParentDevice(ParentObj)
'property to give object a reference to the parent device.
'inheritance and overriding is not possible in VBScript. This reference is a workaround to make
'it possible to call member functions of the parent (e.g. used for Query)
'This reference is only for internal use, so that's why it has only a SET property

'check parameter
If TypeName(ParentObj) <> "ParentDevice" Then
    Err.Raise vbObjectError + ERRlvl_CRITICAL, "Set RadiDev.ParentDevice", _
        TypeName(parentObj) & " is no valid parent device object"
Exit Property
End If

'set reference to parentobj
set m_ParentDevice = ParentObj
End Property

Public Property Let Name(ByVal devName)
'check parameter
If varType(devName) <> vbString Then
    Err.Raise vbObjectError + ERRlvl_CRITICAL, "let RadiDev.Name", _
        "name parameter is not a valid string. device name not set"
Exit Property
End If

'copy the name to data member
m_Name = devName
End Property

Public Property Get Name
'returns the name, even if it's still empty
Name = m_Name
End Property

Public Property Let hwPrefix(ByVal prefix)
'check parameter
If varType(prefix) <> vbString Then
```

## Bijlage II – Volledige broncode RadiCentre test

```
Err.Raise vbObjectError + ERRlvl_CRITICAL , "let RadiDev.hwPrefix" , _
    "prefix parameter is not a valid string. prefix not set"
Exit Property
End If

'copy the prefix to the data member
m_hwPrefix = prefix
End Property

Public Property Get hwPrefix
'return the hwPrefix data member, even if it's empty
hwPrefix = m_hwPrefix
End Property

Public Property Let slotNr(ByVal number)
If not IsNumeric(number) OR number < 0 OR number > SLOT_COUNT Then
Err.Raise vbObjectError + ERRlvl_LOG , "set RadiDev.Slotnumber" , number & " is invalid slot number"
Exit Property
End If

m_slotNr = number 'set the given slot number in data member
If number > 0 Then 'when it's not 0 so the radicentre itself
    hwPrefix = number & ":" 'generate a hwPrefix as "#:", used for communication
End If

End Property

Public Property Get slotNr
'return the slot number from data member, even when it's empty
slotNr = m_slotNr
End Property

Public Property Let Version(ByVal swVersion)
'check the parameter
If varType(swVersion) <> vbString Then
Err.Raise vbObjectError + ERRlvl_CRITICAL , "let RadiDev.Version" , "Version parameter is not a valid string. Version
not set"
Exit Property
End If

'write version to data member
m_Version = swVersion
End Property

Public Property Get Version
Version = m_Version 'return data member, even when it's empty
End Property

Public Property Set subDevice(deviceObj)
'check parameter
If TypeName(deviceObj) <> "RadiDev" OR deviceObj Is Nothing Then 'check if object to set is RadiDev or Nothing
object
Err.Raise vbObjectError + ERRlvl_CRITICAL, "set RadiDev.SubDevice", _
    TypeName(deviceObj) & " is no valid subDevice object" 'throw error if not
Exit Property
End if

Set m_SubDev = deviceObj 'set the subDevice with object from parameter
Set m_SubDev.ParentDevice = m_ParentDevice 'subDevice will get same parent as the device (so NOT device itself as
parent)

If IsEmpty(m_SubDev.hwPrefix) Then 'when given prefix (set by user) is empty
m_SubDev.hwPrefix = m_hwPrefix 'copy own prefix to subDevice
End If
End Property

Public Property Get subDevice
dim result

'always return an object: The RadiDev object, or a "nothing" object
If TypeName(m_SubDev) = "RadiDev" Then 'if it's a valid subDevice
    set result = m_SubDev 'return subDevice
Else 'no subDevice
    set result = nothing 'return nothing object
End If

set subDevice = result 'return result

End Property

'-----Member functions-----

Public Function Query(ByVal command)
'Send the command to the RadiCentre, and read back the reply

'This function only calls the parent's query function, as alternative for inheritance.
```

## Bijlage II – Volledige broncode RadiCentre test

```
'Therefore, parameter is not checked, this is already implemented at parent's query function.

dim result
result = m_ParentDevice.Query(m_hwPrefix & command) 'call Query of parent, with hwPrefix of this device in front of
command

Query = result 'return the result
End Function

Public Sub Write(ByVal command)
'check the command parameter
'This function calls the Write function of the parent object, as alternative to inheritance.
'Therefor, parameter is not checked, this is implemented in the parent's write fucntion

m_parentDevice.Write m_hwPrefix & command 'send command to device, with hardware prefix of THIS object in front of it
End Sub

Public Function detectHardware
'This do-function have to scan for hardware
dim hwLetter : hwLetter = Array("P", "S", "M", "L", "G", "T", "X", "W") 'possible hw prefixes
dim i : i=0 'index for hwLetter array
Dim reply : reply = ""
Dim deviceFound: deviceFound = false

do
On Error Resume Next 'start LOCAL error catching
reply = Query("IDN?") 'Try query: first with default prefix (only digit e.g. 3:)
If Err.Number <> 0 Then : Exit Function : End If
On Error GoTo 0 'Stop LOCAL error catching (has NO effect on caller's error catching)

If Left(reply, 5) <> "ERROR" Then 'device replied with no ERROR in reply
deviceFound = true 'set flag
ElseIf not IsEmpty(hwPrefix) Then 'If device DOES reply error, and has a prefix (so not for RadiCentre)
hwPrefix = hwLetter(i) & slotNr & ":" 'ADD a hardware letter to the prefix
i = i+1 'increase index, to get the next letter next time
End If
Loop While deviceFound = false AND i <= UBound(hwLetter) AND not IsEmpty(hwPrefix)
'keep trying while NO device found AND not all letters are tried AND it's a device WITH prefix

If deviceFound Then
dim regex, matches
set regex = new RegExp 'create regular expression object

regex.pattern = "[A-Z]{3}\d{4}[A-Z]|(RadiCentre)|(EMCenter)|(\d{4}-\d{3})" 'search for valid device type name
set matches = regex.execute(reply)

If matches.Count >= 1 Then 'device name is found
Name = matches(0) 'get the found match and save as device name
Else
'throw error
Err.Raise vbObjectError + ERRRLVL_LOG, "Hardware detection", _
"Can't find device name in the next device reply:" & reply
End If 'end if device name found

regex.pattern = "\d+(\.\d+)+" 'search for valid version number 1[1234.15345.456315.4583158....]
set matches = regex.execute(reply)
If matches.count >= 1 Then 'version number found
Version = matches(0) 'save the found match as version number
Else
Err.Raise vbObjectError + ERRRLVL_LOG, "Hardware detection", _
"Can't find software version in the next device reply:" & reply
End If 'End if version number found

End If 'End If device Found

detectHardware = deviceFound
End Function

Private Sub Class_Initialize
End Sub

Private Sub Class_Terminate
'set m_ParentDevice = nothing
'de-referencing (clearing) memberobject will already be done automatically after the destructor is called
End Sub
End Class

Function GetRadiDevInst()
set GetRadiDevInst = new RadiDev
End Function

'---- Main class, used for testing the RadiCentre and connected hardware --
class RadiTest
'----private Class Members-----
private xlsDbase
private m_RadiCentre
private m_ErrObjects 'will be a dynamic array as soon as it's used
'---public properties to access class members-----

Public Property Get RadiCentre()
Set RadiCentre = m_RadiCentre
End Property
```

## Bijlage II – Volledige broncode RadiCentre test

```
'-----member functions-----

'Constructor
Private Sub Class_Initialize
    set m_RadiCentre = new ParentDevice          'Set RadiCentre member object
    set xlsDbase = Project.Variables.cmdDatabase  'set reference to command database

    'Initialize member array for device list
    dim i
    for i=0 To SLOT_COUNT
        set m_RadiCentre.device(i) = new RadiDev
    Next
End Sub

'Destructor
Private Sub Class_Terminate
    abortTest 'private member routine, to make sure all members are killed and destructor's are called
End Sub

Private Sub abortTest
    'This private routine should be called always when finishing log, to make sure all objects are deleted
    m_RadiCentre.terminateChilds 'first terminate all child's, so their ref's to the parent are gone
    set m_RadiCentre = nothing    'destroy RadiCentre, this should be the last ref to RadiCentre
    Log.Message "Test Stopped"
End Sub

Public Sub AddErrRef(ByRef ErrObj)
    If IsArray(m_ErrObjects) Then 'the variable is set as array before, so should have at least 1 element
        ReDim Preserve m_ErrObjects(UBound(m_ErrObjects) + 1) 'increase the size of the list
        set m_ErrObjects(UBound(m_ErrObjects)) = ErrObj        'add (reference to) object in last element
    Else 'm_ErrObjects is no array, so has no elements
        m_ErrObjects = Array(ErrObj) 'create an array with ErrObj as first element
    End If
End Sub

Public Sub ErrorChecker
    'this function will check if an error has occurred, and handle them if necessary

    dim localErr
    For Each localErr In m_ErrObjects 'check all err object on error (until error found)
        If localErr.Number <> 0 Then
            On Error Resume Next 'catch it local, because we're gone handle it ourselves
            Err.Raise localErr.Number, localErr.Source, localErr.Description 'raise the found error local
            localErr.Clear 'clear the error in the source err object
            Exit For 'stop searching for error
        End If
    Next

    If Err.Number <> 0 Then 'When there is an error

        Select Case Err.Number - vbObjectError 'subtract vbObjectError to get custom number back again
            Case ERRRLVL_CRITICAL_GLOBAL:
                'show error box, add error to log and stop execution of complete project
                MsgBox "Critical error occurred. Test will stop without running next test items. See test log for details", _
                    vbCritical, "RadiCentreTest - TestComplete"
                Log.Error "critical error in : " & Err.Source, _
                    "Error source: " & Err.Source & VbLf & "Error message: " & Err.Description
                abortTest 'finish the testclass (close connection, destruct members)
                Err.Clear

                Runner.Stop False 'Stop Current test, but continue next test item (of same level, so not the childs of
This)

            Case ERRRLVL_CRITICAL:
                'show Error box and add error to log
                dim userChoice
                userChoice = MsgBox("Critical error occurred in : " & Err.Source & VbLf & Err.Description & VbLf & VbLf & "Continue
with next item?" , _
                    vbCritical+vbYesNo, "RadiCentreTest - TestComplete")
                Log.Error "critical error in : " & Err.Source, _
                    "Error source: " & Err.Source & VbLf & "Error message: " & Err.Description
                abortTest 'finish the testclass (close connection, destruct members)
                Err.Clear

                If userChoice = vbYes Then
                    Runner.Stop True 'Stop Current test, but continue next test item (of same level, so not the
childs of This)
                Else
                    Runner.Stop False 'Stop testing
without continueing with any next test item
                End If

            Case ERRRLVL_LOG:
                'Just add error to the log
                Log.Error "Error while testing in : " & Err.Source, _
                    "Error source: " & Err.Source & VbLf & "Error message: " & Err.Description
                'continue testing, error is not critical

            Case ERRRLVL_WARNING:
                'Just add warning to log
                Log.Warning "Warning in : " & Err.Source, _
                    "Warning source: " & Err.Source & VbLf & "Error message: " & Err.Description
```



## Bijlage II – Volledige broncode RadiCentre test

```
'continue testing, warning is not critical

Case Else:
'unknown error, show error box and log the unknown exception
MsgBox "Unexpected error occurred while testing. See test log for details", vbCritical, "RadiCentreTest -
Testcomplete"
Log.Error "Unknown error occurred in: " & Err.Source, _
        "Error source: " & Err.Source & vbCrLf & "Original Error description:" & Err.Description
Me.AbortTest          'finish the testclass (close connection, destruct members)
Err.Clear
Runner.Stop False    'stop execution of current test, don't continue with next item
End Select
End If
Err.Clear
End Sub

Public Sub testSlot(slotNr)
    dim hwFound :hwFound = false

    On Error Resume Next
    hwFound = me.RadiCentre.device(slotNr).detectHardware    'Function to detect hardware, returns true or false
    ErrorChecker
    On Error Goto 0

    If hwFound Then
        testCommands m_RadiCentre.device(slotNr)            'start testing current device
        If not m_RadiCentre.device(slotNr).subDevice Is Nothing Then 'if device has ext. connected HW (entered by user)
            testCommands m_RadiCentre.device(slotNr).subDevice 'test this ext. HW also
        End If
    Else
        Log.Message "no hardware found in slot " & slotNr    'log message that no hardware is detected in this slot
    End If
End Sub

Private Sub testCommands(device)
    'start testing

    Indicator.PushText "Initializing command database"        'change indicator text

    On Error Resume Next
    xlsDbase.Reset          'Try: Load excel file and start reading at first row
    ErrorChecker
    On Error Goto 0
    Indicator.PopText        'return previous indicator text

    'add device info as folder in the log, next log items (commands) will be child's of this folder
    Log.AppendFolder "Device " & device.hwPrefix & device.Name & ", Version:" & device.Version
    Indicator.PushText Indicator.Text & "Testing " & device.hwPrefix & device.Name    'change indicator text
    Dim commandCount: commandCount = 0    'command counter, used for indicator text

    'initialize variables, used for reading from excel file
    dim dbDevType, dbCommand, dbExpAnswer, dbDelay, dbVerFrom,dbVerTo
    dbDevType = xlsDbase.ColumnNames(0)    'Read 1st column name (cell A0)
    dbCommand = xlsDbase.ColumnNames(1)    'Read 2nd column name (cell B0)
    dbExpAnswer = xlsDbase.ColumnNames(2)    'Read 3rd column name (cell C0)
    dbDelay = xlsDbase.ColumnNames(3)    'Read 4th column name (cell D0)
    dbVerFrom = xlsDbase.ColumnNames(4)    'Read 6th column name (cell E0)
    dbVerTo = xlsDbase.ColumnNames(5)    'Read 7th column name (cell F0)

    dim commandFound: commandFound = False    'flag used for searching commands
    While Not xlsDbase.IsEOF    'read complete excel file

        If not IsNull(xlsDbase.Value(dbDevType)) Then    'skip if first cell (device type) is empty

            dim regEx: set regEx = new RegExp    'regular expression object
            regEx.pattern = xlsDbase.Value(dbDevType)    'read devtype from excel
            regEx.pattern = replace(regEx.pattern, ";", "|")    'replace ; for | (or)
            regEx.pattern = Trim(regEx.pattern)    'remove spaces

            'look for: device name from excel matches device name of current device
            'actual sw version of device is HIGHER then minimal sw version of excel command
            'actual sw version of device is LOWER then maximal sw version of excel command
            If regEx.Test(device.Name) _
                And (IsNull(xlsDbase.Value(dbVerFrom)) OR device.Version >= xlsDbase.Value(dbVerFrom)) _
                And (IsNull(xlsDbase.Value(dbVerTo)) OR device.Version <= xlsDbase.Value(dbVerTo)) _
            Then
                commandFound = True    'set flag because there is a valid command found
                commandCount = commandCount + 1    'increase command counter

                'change indicator text (add command details to the actual text)
                Indicator.PushText Indicator.Text & vbCrLf & "Command " & commandCount & vbCrLf & xlsDbase.Value(dbCommand)

                'execute the command
                If IsNull(xlsDbase.Value(dbExpAnswer)) Then    'when "expected answer" is empty (no answer expected)
                    On Error Resume Next
                    device.Write xlsDbase.Value(dbCommand)    'only SEND a command, without reading reply
                    Log.Event xlsDbase.Value(dbCommand), "Command send, no answer expected"    'log the send command
                    ErrorChecker : On Error Goto 0

                Else    'when "expected answer" is NOT empty
                    dim answer
```

## Bijlage II – Volledige broncode RadiCentre test

```

On Error Resume Next
answer = device.Query(xlsDbase.Value(dbCommand))      'send command and read back the reply
ErrorHandler : On Error GoTo 0

dim regexAnswer: set regexAnswer = new RegExp          'new regular expr. used for checking the answer
regexAnswer.Pattern = "^(" & xlsDbase.Value(dbExpAnswer) & ")$" 'answer should match COMPLETE expr.

'compose a message for the log, used for both error and checkpoint
dim msgtext
msgtext = "Command:" & aqstring.Quote(xlsDbase.Value(dbCommand)) & vbCrLf _
          & "Received :" & aqstring.Quote(Answer) & vbCrLf _
          & "Expected :" & aqstring.Quote(xlsDbase.Value(dbExpAnswer))

'test answer using the regular expression
If regexAnswer.Test(answer) Then
    Log.Checkpoint xlsDbase.Value(dbCommand), msgtext 'answer passed the check
Else
    Log.Error xlsDbase.Value(dbCommand), msgtext      'answer failed the check
End If
End If

'after command, wait a delay, read from excel. When delay is not entered in excel, wait 100 ms
dim cmdDelay
cmdDelay = xlsDbase.Value(dbDelay)
If IsNull(cmdDelay) Then 'use delay from excel database if available
    cmdDelay = 100
End If

Delay cmdDelay, Indicator.Text 'indicator will change while waiting delay, use same text to show
Indicator.PopText             'return to previous indicator text (remove command details)
End If
End If
xlsDbase.Next                 'go to next row in the excel database
Wend 'reached end of Excel file (command database)

If not commandFound Then 'not even 1 command found in database for current device
    Log.Error "Device " & device.Name & " not in xls database"
End If

Log.PopLogFolder             'return to previous level in log (next log items will NOT be child's of the device)
Indicator.PopText            'return to previous indicator text
End Sub
end class

'helper routine to create an object of class in another script file
Function GetRadiTestInst(ByRef ErrObj)
    'this function is a workaround for two limitations
    '1: It is not possible to create an instance of a class in another file using new
    '2: The VBScript Err object is in fact not global, but local on Script Unit File level
    'This Function creates an instance local, and returns that object (to get a new instance in another file)

    'When creating an instance in another file, there MUST be a reference set to the error object of that file
    'therefor this function has the errObj parameter. Creating an instance in another file is therefore only possible
    'when adding a reference to the local Err object in the parameter

    'TODO: check if parameter is a valid err object

    'check paramter on valid object (TypeName is no option for Errobject)
    If not IsObject(ErrObj) Then
        Err.Raise vbObjectError + ERLVL_CRITICAL_GLOBAL, "GetRadiTestInst", _
            "ErrObj parameter is no valid reference to the Err object"
    End If

    Dim RadiTestObj
    set RadiTestObj = new RadiTest 'create a new instance of th RadiTest class
    RadiTestObj.AddErrRef ErrObj    'add a reference to the error object of the file of the caller

    Set GetRadiTestInst = RadiTestObj 'return the new instance
End Function
'-----

'USEUNIT ClassDefinitions
option explicit 'variables must always be declared with DIM before they can be used

'global variables and objects
Dim localTest
Dim subDevList() : ReDim subDevList(SLOT_COUNT) 'array size is a const from ClassDefinitions file

Sub configureSubDev(slot, devName, hwPrefix)
    set subDevList(slot) = GetRadiDevInst
    subDevList(slot).Name = devName
    subDevList(slot).slotNr = slot
    subDevList(slot).hwPrefix = hwPrefix
End Sub

Sub initConnection(source)

    Set localTest = GetRadiTestInst(Err)
    'localTest.AddErrRef Err 'set a reference to the error object of this unit file

```

## Bijlage II – Volledige broncode RadiCentre test

```
On Error Resume Next
Select case source
Case "LAN"

    'Windows saves the MAC address for an IP address.
    'When a new device (= new mac address) is connected, windows have to ask AGAIN: who has [ip]?
    'therefore, delete the existing entry for the used IP adres in the ARP table
    'to be sure, windows will defenitly always ask first who the specified IP address has.

    Sys.OleObject("WScript.Shell").Run("arp -d " & Project.Variables.IPAddress)
    delay 500
    localTest.RadiCentre.Connect "TCPIP0:." & Project.Variables.IPAddress

Case "GPIB"
    msgbox "connect gpib"
    localTest.RadiCentre.Connect "GPIB0:." & project.Variables.GPIBAddress

Case "Serial"
    localTest.RadiCentre.Connect "COM"& project.Variables.COMport
    localTest.ErrorChecker 'check to prevent err will be overwritten by next action
    localTest.RadiCentre.ConfSerial Project.Variables.COMBaud, _
        Project.Variables.COMDataBits, _
        Project.Variables.COMStopBits

Case "USB"
    localTest.RadiCentre.Connect("COM"& project.Variables.USBCOMPort)
    localTest.ErrorChecker
    localTest.RadiCentre.ConfSerial 115200, 8, 1

Case Else
    'raise customer error

    Err.Raise vbObjectError + ERR_LVL_CRITICAL, "initConnection " & source, "no valid resource name"
End Select

localTest.ErrorChecker 'check if opening session has raised any errors

localTest.RadiCentre.Query "*IDN?" 'no error after opening, so check if communication is possible

'override original error to show user the error raised by opening the session and not while testing
If Err.Number <> 0 Then
    dim eDescription: eDescription = Err.Description 'copy current error description
    Err.Clear 'clear the original error
    'raise new error and copy original error description
    Err.Raise vbObjectError + ERR_LVL_CRITICAL, _
        "initConnection " & source, _
        "error while opening. Session was created, but communication failed. " & vbCrLf & eDescription
    localTest.ErrorChecker 'error handling
End If

On Error Goto 0 'stop local error catching

Indicator.PushText source & vbCrLf 'add source name to indicator

End Sub

Sub testDevice(slotNr)

    If not IsEmpty(subDevList(slotNr)) Then
        On Error Resume Next
        set localTest.RadiCentre.device(slotNr).subDevice = subDevList(slotNr)
        LocalTest.ErrorChecker : On Error Goto 0
    End If

    localTest.testSlot slotNr
End Sub

Sub Finalize
    Indicator.PopText 'resume the previous text of the indicator (remove last resource name)
    On Error Resume Next
    localTest.RadiCentre.Query "LOCAL"
    LocalTest.ErrorChecker : On Error Goto 0
    set localTest = nothing
End Sub

'-----
```

## Bijlage III – broncode algemene routine voor het starten van RadiMation testen

```
Sub StartTSFWaitForReady(vsMenu, vsTSF)

    'first of all, start the TSF by clicking the menu
    Log.AppendFolder(vsTSF)                                'create folder in log tree

    StartTSF vsMenu, vsTSF                                'Open TSF and start by clicking the run/calculate/start test button
    dim tTestStart: tTestStart = Now                       'save the current time as start time of the test

    'next procedure is depending on the test
    defaultTestTimeout = #00:15:00#                       '15 minutes
    bMultiBand = (InStr(vsMenu, "Multiband") > 0)          'set boolean flag to determin multi- or singleband
    Select Case vsTSF
    Case "ESD QA Test1"                                     'define procedure, depending on test
        PerformESD                                         'Special procedure, no need to wait
    for ready
    Case "EFT QA Test1"                                     : PerformEFT                                             'Special procedure, no need to wait for ready
    Case "Surge QA Test 1"                                   : PerformSurge                                           'Special procedure, no need to wait for ready
    Case "QA Pause Test 1"                                   : PerformConfiguredToolsTSFS 'Special procedure, no need to wait for ready
    Case "RE Magnetic, QA Test 1" :                         'wait for several message boxes, before start waiting test to finish
        set p = GetRadiMationProcess
        p.WaitWindow("Magnetic Field Test", -1, 5000).Ok.ClickButton 'wait
    msgbox, click ok. Timeout = 5sec.
        WaitForTestCompletion #00:05:00#, bMultiBand        '5 minutes after last msg should be
    enough for this test

    Case "RE Magnetic, QA Test 2":
        set p = GetRadiMationProcess
        p.WaitWindow("Magnetic Field Test", -1, 5000).Ok.ClickButton 'wait
    msgbox, click ok. Timeout = 5sec.
        WaitForTestCompletion #00:10:00#, bMultiBand        'wait max 10 minutes for test to be
    finished, after last msg

    Case "QA G-Tem Test 1"
        set p = GetRadiMationProcess
        For i = 0 to 5
            p.WaitWindow("Radiated Emission GTEM", -1, 30000).Ok.ClickButton 'wait for several msgboxes, and click OK
        Next
    'wait msgbox, click ok. Timeout = 30sec.
        WaitForTestCompletion #00:10:00#, bMultiBand        'wait for test to be finished, timeout = 10
    minutes

    Case "QA Network Analyser System Calibration 1"
        set p = GetRadiMationProcess
        For i = 0 To 55
            p.WaitWindow("RadiMat", -1, 10000).Ok.ClickButton
        Next
        'wait for msgbox and click ok. Timeout = 10 sec
        WaitForCalibrationFileSave #00:01:00#, vsTSF
    'wait for completion of calibration

    Case "QA Network Analyser System Calibration 2"
        set p = GetRadiMationProcess
        For i = 0 To 15
            p.WaitWindow("RadiMat", -1, 10000).Ok.ClickButton
        Next
        'wait for msgbox and click ok. Timeout = 10 sec
        WaitForCalibrationFileSave #00:01:00#, vsTSF
    'wait for completion of calibration

    Case "QA Att System Cal 1"
        set p = GetRadiMationProcess
        For i = 0 To 1
            p.WaitWindow("RadiMat", -1, 10000).Ok.ClickButton
        Next
        'wait for msgbox and click ok. Timeout = 10 sec
        WaitForCalibrationFileSave #00:01:00#, vsTSF
    'wait for completion of calibration

    Case "QA compression System Cal 1"
        set p = GetRadiMationProcess
        For i = 0 To 1
            p.WaitWindow("RadiMat", -1, 10000).Ok.ClickButton
        Next
        'wait for msgbox and click ok. Timeout = 10 sec
        WaitForCalibrationFileSave #00:01:00#, vsTSF
    'wait for completion of calibration

    Case Else
        'generic procedure for both multi- and single band
        If InStr(vsMenu, "Calibration") > 0 Then
            'do the calibration thing
            WaitForCalibrationFileSave defaultTestTimeout, vsTSF
        Else
            WaitForTestCompletion defaultTestTimeout, bMultiBand
        End If
    End Select

    dim tTestStop: tTestStop = Now                        'save current date/time value to determine total test time

    totalTestTime = DateDiff( "s", tTestStart, tTestStop) 'calculate elapsed test time in seconds
    Log.Message "total test time in seconds: " & totalTestTime
    Project.Variables.TSFTimingCollector.Add vsTSF, totalTestTime 'add time with tsf name to datacollection

    Log.PopLogFolder                                     'set the log-level back again one level

    'when here, the test is finished!

End Sub
```

## Bijlage IV – broncode IniFile klasse

```

class IniFile
    private m_FileLocation
    private m_iniFile
    private rootSectionAlreadyExists

    Public Sub Class_Initialize()
        set m_iniFile = nothing
        rootSectionAlreadyExists = True
    End Sub

    Public Sub Class_Terminate()
        set m_iniFile = nothing
        If not rootSectionAlreadyExists Then
            removeRootSection
        End If
    End Sub

    Public Function Open(filename)
        dim result: result = false

        If aqFile.Exists(filename) Then
            m_FileLocation = filename
            createRootSection
            set m_iniFile = Storages.INI(m_FileLocation)
            result = true
        End If

        Open = result

    End Function

    Public Sub Close
        m_iniFile.Save

        If not rootSectionAlreadyExists Then
            removeRootSection
            rootSectionAlreadyExists = True
        End If
        set m_iniFile = nothing

    End Sub

    Private Sub createRootSection
        'the used storages.ini implementation nees ALWAYS a [root] section in the file.
        'This function will add this section manually, to be able to use the storages.ini method

        content = aqFile.ReadWholeTextFile(m_FileLocation, aqFile.ctANSI)

        If Instr(content, "[root]") <> 0 Then
            rootSectionAlreadyExists = True
        Else
            'no, just ad for now, and also remove afterwards
            rootSectionAlreadyExists = False
        End If

        content = "[root]" & vbCrLf & content
        aqFile.WriteToTextFile m_FileLocation, content, aqFile.ctANSI, True

    End Sub

    Private Sub removeRootSection
        'when done, the custom added [root] section can be removed again

        content = aqFile.ReadWholeTextFile(m_FileLocation, aqFile.ctANSI)
        content = aqString.Replace(content, "[root]" & vbCrLf, "", false)

        call aqFile.WriteToTextFile(m_FileLocation, content, aqFile.ctANSI, True)

    End Sub

    Public Function ReadString(SectionName, ItemName, DefaultValue)
        dim result

        If not m_iniFile is nothing Then
            result = m_iniFile.GetSubSection(SectionName).GetOption(ItemName, DefaultValue)
        End If

        'return
        ReadString = result
    End Function

    Public Function ReadAllItemsInSection(SectionName, DefaultValue)
        If m_iniFile is Nothing Then : Exit Function: End IF

        set itemMap = CreateObject("Scripting.Dictionary")
        set curSection = m_iniFile.GetSubSection(SectionName)

        'get all options in the currently selected section

        For itemIndex = 0 To curSection.OptionCount -1
            itemName = curSection.GetOptionName(itemIndex)
            'get name of current item
            itemValue = curSection.GetOptionByIndex(itemIndex, DefaultValue)
            'get value of current item

            itemMap.Add itemName, itemValue
            'add the results as key -> value to the data Map
        Next
        set ReadAllItemsInSection = itemMap
        'return the data map with the found items
    End Function

    Public Sub WriteString(SectionName, ItemName, Value)
        If not m_iniFile is Nothing Then
            call m_iniFile.GetSubSection(SectionName).SetOption(ItemName, Value)
        End If
    End Sub
End Class

```

## Bijlage V – broncode xlsListDataCollector

```

class xlsListDataCollector
private m_DataCollection           'map with the keys and values
private m_xlsPath                 'path to the xls file
private m_xlsApplication           'OLE object to access excel application
private m_xlsWorkbook             'OLE object to access excel workbook
private m_xlsWorksheet            'OLE object to access excel worksheet
private m_OnAddEventHandler        'Reference to function which will be called when Add function is called

'Default class methods -----
Public Sub class_initialize         'class constructor
    set m_DataCollection = CreateObject("scripting.dictionary")

    'xls object will be set when writing to excel
    set m_xlsApplication = Nothing
    set m_xlsWorkbook = Nothing
    set m_xlsWorksheet = Nothing
    set m_OnAddEventHandler = Nothing
End Sub

Public Sub class_terminate
End Sub

'Public Property members

Public Property Get Item(keyName)
    dim result                     'create empty variable
    If m_DataCollection.Exists(keyName) Then      result = m_DataCollection.Item(keyName)

    Item = result                  'return the (empty or filled) result variable
End Property

Public Property Let Item(keyName, Value)
    m_DataCollection.Item(keyName) = Value        'overwrite or create element
End Property

Public Property Get Count
    Count = m_DataCollection.Count                'returns the number of keys, saved in the collection
End Property

'Public member functions -----
Public Function Exists(key)
    Exists = m_DataCollection.Exists(key)          'return the result of the exist function
End Function

Public Sub Add(keyName, Value)
    'adding existing item should overwrite existing value.
    'Therefore, don't use the dictionary.add method, which will raise an error on overwrite
    'just use item(key) = value, which will overwrite if exists, or also add if not exists

    If not m_OnAddEventHandler is Nothing Then      'if 'user' set an event handler for OnAdd event
        m_OnAddEventHandler(Me)
    'Call the event function, with reference to THIS object as sender
    End If

    m_DataCollection.Item(keyName) = Value          'add the value to the data collection
End Sub

Public Property Set OnAdd(EventHandler)
    'because of using property, script engine already checked that parameter is a valid object
    'because it's an object, user should have used GetRef, which already validated the procedure
    set m_OnAddEventHandler = EventHandler
End Property

Public Sub SaveDataToXls(pathToFile, rowName)
    xlsOpen pathToFile                          'open the xls file

    'write the data
    Dim RowIndex, ColIndex
    RowIndex = GetRowIndex(rowName)              'get index of row with specified name (add row if it not exists)

    For Each keyName In m_DataCollection.Keys
        ColIndex = GetColIndex(keyName)          'get the column index of column with name of the key (created if not exists)
        m_xlsWorksheet.Cells(RowIndex, ColIndex) = m_DataCollection(keyName)      'write value of current key to excel
    Next

    xlsClose
End Sub

'Private member functions

Private Sub xlsOpen(pathToFile)
    set m_xlsApplication = CreateObject("Excel.Application")

    dim fso                      'local var for FileSystemObject, used to check if file exists
    Set fso = CreateObject("Scripting.FileSystemObject")

    If fso.FileExists(pathToFile) Then
        set m_xlsWorkbook = m_xlsApplication.Workbooks.Open(pathToFile)          'open existing file
    Else
        set m_xlsWorkbook = m_xlsApplication.Workbooks.Add                      'create new workbook
        m_xlsWorkbook.SaveAs(pathToFile)                                         'and save that file
    End If

    set m_xlsWorksheet = m_xlsWorkbook.sheets(1)                                'set ref to 1st sheet
End Sub

Private Function GetRowIndex(rowTitle)
    Dim RowIndex
    Dim foundRange
    Set foundRange = m_xlsWorksheet.Columns(1).Find(rowTitle, , , xlWhole)        'search for the text in the first column,
    xlWhole = exact match only!

    If foundRange is Nothing Then          'if the specified row title is not found
        'search for the first empty row. search from bottom to top
        Set bottomCell = m_xlsWorksheet.Cells(m_xlsWorksheet.Rows.Count, 1)      'get the very last cell in column one
        RowIndex = bottomCell.End(xlUp).Row + 1                                  'get index of cell below first used cell (from bottom to top)
        m_xlsWorksheet.Cells(RowIndex, 1) = rowTitle                            'enter the desired name in the new (empty) row title
    Else
        'the specified row title is found

```

## Bijlage V – broncode xlsListDataCollector

```

       RowIndex = foundRange.Row                'return the rowindex of the found row
    End If

    getRowIndex = RowIndex                        'return the found value
End Function

Private Function getColIndex(colTitle)
    Dim ColIndex
    Dim foundRange
    Set foundRange = m_xlsWorksheet.Rows(1).Find(colTitle, , , xlWhole) 'search for the text in the first row, xlWhole = exact match
only!

    If foundRange is Nothing Then
        dim mostRightCell
        Set mostRightCell = m_xlsWorksheet.Cells(1, m_xlsWorksheet.Columns.Count) 'get the very most right cell in the 1st
row

        ColIndex = mostRightCell.End(xlToLeft).Column + 1 'get index of cell next to first used cell (search from right to left)
        m_xlsWorksheet.Cells(1, ColIndex) = colTitle 'enter the desired title in the newly found column title
    Else
        'column title already exists
        ColIndex = foundRange.Column 'get the column index of the found column
    End If

    getColIndex = ColIndex
End Function

Private Sub xlsClose
    m_xlsWorkbook.Save 'save the changes
    m_xlsWorkbook.Close True 'close the workbook, and save changes
    m_xlsApplication.Quit 'exit the application
End Sub
End Class
```