Thema, Ziele: Debugger für Embedded Systems

Aufgabe 1: Codepatching

siehe./Loesung/A1

a) 0x20100150
b) 0xEAFBFFAA
c) Patch:
   1. sichert alle Register ausser PC
   2. führt „patchedCode“ C-Funktion aus
   3. stellt original opcode an „PATCH_ADR“ wieder her
   4. lädt alle Register ausser PC
   5. springt zu „PATCH_ADR“

   codeInsertion:
   1. sichert original opcode von „PATCH_ADR“ Adresse
   2. lädt branch opocode in „PATCH_ADR“ Adresse
d) asm(“LDR PC, =0x201000a0”);

Aufgabe 2: Disassembler

siehe ./Loesung/A2

Opcode	Instr	NextAdr
0xEAFBFFF9	B	0x20000000
0xEBFBFFF9	BL	0x20000000
0x1AFCFFAC	B	0x21140008
0xE3A00000	(MOV)	0x21200154 (PC+4Byte)

Aufgabe 3: Funktionsweise eines „on chip“ Debuggers

siehe ./Loesung/A4

a) Just do it
b) Just do it
c) Disassemblierung und danach Codepatching, vom OCD Stub ausgeführt
d) Der GDB selbst (PC seitig) nimmt das Codepatching vor. Im Gegensatz zum „single step“ weiss der GDB wo er den BKPT setzen muss, da der User die gewünschte Stelle (Adresse) angibt. Somit ist hier kein Disassembling nötig.