Report Assignment 2

Assembly Forensics

Group Member

Platzer Andreas 11771913

Reindl Hanner 01532129

Manuel Sammer 11903022

# Getting Ready

We used the GNU Tools ARM Embedded arm-none-eabi binary command line executeables to compile the assembler program.

Nothing happened with the Raspberry Pi as predicted.

# More Blinking!

## What does it do?

It blinks! To be more precise: It sends SOS in morse code very fast, in an endless loop.

“\*.\*.\*....\*\*\*.\*\*\*.\*\*\*....\*.\*.\*........” repeat!

\* => One “dot\_length” unit, LED ON

. => One “dot\_length” unit, LED OFF

## How to get the Symbol Table and Assembler Code

Assembler Code:

"C:\Program Files (x86)\GNU Tools ARM Embedded\8 2019-q3-update\bin\arm-none-eabi-objdump" -d kernel.elf > kernel\_dump.asm

Symbol Table:

"C:\Program Files (x86)\GNU Tools ARM Embedded\8 2019-q3-update\bin\arm-none-eabi-objdump" -t kernel.elf > kernel\_dump\_symbols.txt

## Flowchart with Explanation of Code

## Pseudocode:

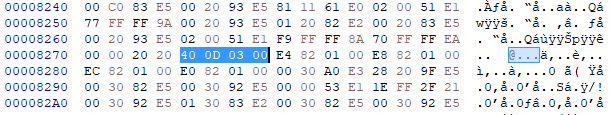
main()  
{  
 dot\_length = 200000;  
 gpio = 0x20200000;  
 gpio[1] |= 0x40000;

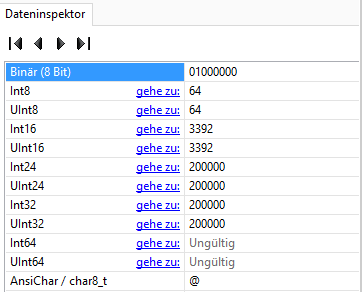
while(1)  
 {  
 morser = 0;  
 r2 = morser;  
 //Short Blinnking 3times  
 for(morser = 0; morser <= 2; morser++)  
 {  
 gpio[10] = 0x10000; //Turn on LED  
 for(timer = 0; timer <= dot\_length; timer++);  
 //LED On for duration of one “dot\_length” unit  
 gpio[7] = 0x10000; //Turn off LED  
 for(timer = 0; timer <= dot\_length; timer++);  
 //LED Off for duration of one “dot\_length” unit  
 }  
 for(timer = 0; timer <= 3\*dot\_length; timer++);   
 //Wait 3 dot\_length units (break between characters)  
 for(morser = 0; morser <= 2; morser++)  
 {  
 gpio[10] = 0x10000; //Turn on LED  
 for(timer = 0; timer <= 3\*dot\_length; timer++);  
 //LED On for duration of three “dot\_length” units  
 gpio[7] = 0x10000; //Turn off LED  
 for(timer = 0; timer <= dot\_length; timer++);  
 }  
 for(timer = 0; timer <= 3\*dot\_length; timer++);  
 //Wait 3 dot\_length units (break between characters)  
 for(morser = 0; morser <= 2; morser++)  
 {  
 gpio[10] = 0x10000; //Turn on LED  
 for(timer = 0; timer <= dot\_length; timer++);  
 gpio[7] = 0x10000; //Turn off LED  
 for(timer = 0; timer <= dot\_length; timer++);  
 }  
 for(timer = 0; timer <= 7\*dot\_length; timer++);   
 //Space between words in morse code is 7 dot\_length units  
 }  
}

# Too much Blinking!

At the offset 0x8274, there is a DEC value of 200000. This one gets saved once into the memory address of “dot\_length” (0x000182e4) (see flowchart) which gave us the idea of simply editing the value at the offset 0x8274 to 400000 via HxD Editor which worked instantly.

Before edit:





Edited:

