Recap

Week 8

Michi Fallegger
Mario Felder

Hochschule Luzern
Technik & Architektur

5. November 2014

Shell Connections

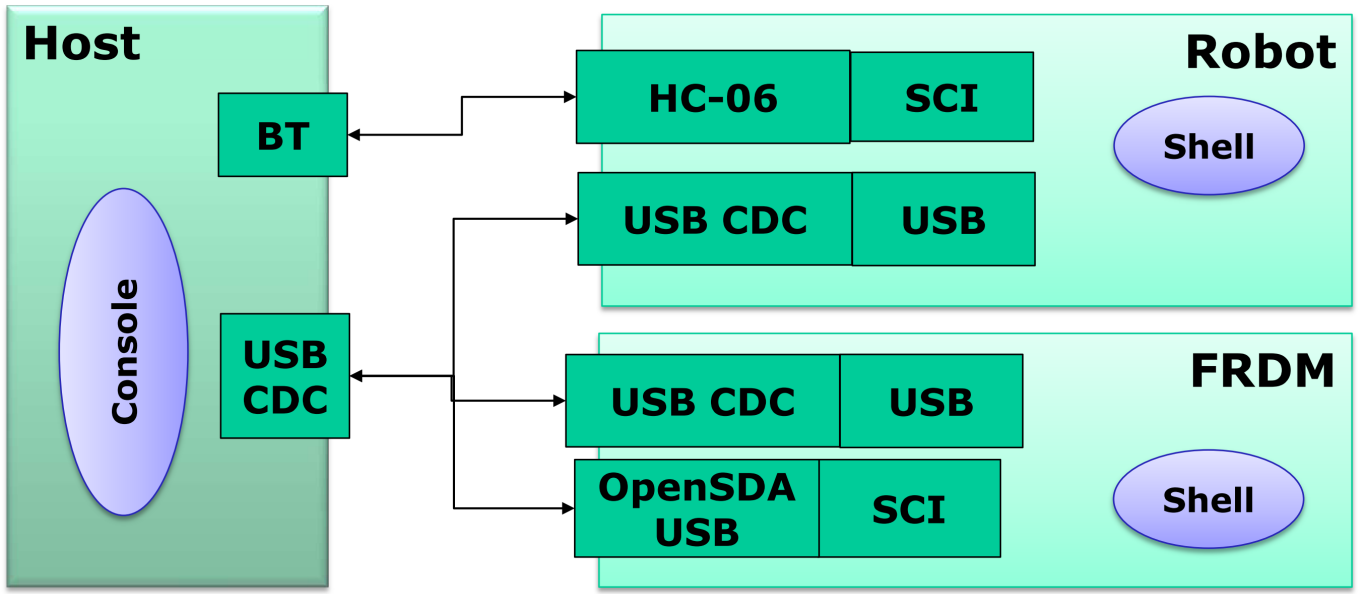

Abbildung: Shell Connections

Main Task
Part I

```
1  static portTASK_FUNCTION(ShellTask, pvParameters) {
2  #if PL_HAS_USB_CDC
3    static unsigned char cdc_buf[48];
4  #endif
5  #if PL_HAS_BLUETOOTH
6    static unsigned char bluetooth_buf[48];
7  #endif
8    static unsigned char localConsole_buf[48];
9  #if CLS1_DEFAULT_SERIAL
10   CLS1_ConstStdIOTypePtr ioLocal = CLS1_GetStdio();
11 #endif
12
13   (void)pvParameters; /* not used */
14 #if PL_HAS_USB_CDC
15   cdc_buf[0] = '\0';
16 #endif
17 #if PL_HAS_BLUETOOTH
18   bluetooth_buf[0] = '\0';
19 #endif
20   localConsole_buf[0] = '\0';
21 #if CLS1_DEFAULT_SERIAL
22   (void)CLS1_ParseWithCommandTable((unsigned char*)CLS1_CMD_HELP, ioLocal,
         CmdParserTable);
23 #endif
```

Main Task
Part II

```
1   for(;;) {
2 #if CLS1_DEFAULT_SERIAL
3     (void)CLS1_ReadAndParseWithCommandTable(localConsole_buf,
          sizeof(localConsole_buf), ioLocal, CmdParserTable);
4 #endif
5 #if PL_HAS_USB_CDC
6     (void)CLS1_ReadAndParseWithCommandTable(cdc_buf, sizeof(cdc_buf),
          &CDC_stdio, CmdParserTable);
7 #endif
8 #if PL_HAS_BLUETOOTH
9     (void)CLS1_ReadAndParseWithCommandTable(bluetooth_buf,
          sizeof(bluetooth_buf), &BT_stdio, CmdParserTable);
10 #endif
11    FRTOS1_vTaskDelay(50/portTICK_RATE_MS);
12  } /* for */
13 }
```

Parse Methode

```
1  uint8_t CLS1_ParseWithCommandTable(const uint8_t *cmd, CLS1_ConstStdIOType *io,
       CLS1_ConstParseCommandCallback *parseCallback){
2    uint8_t res = ERR_OK;
3    bool handled, silent;
4
5    if (*cmd=='\0') { /* empty command */
6      return ERR_OK;
7    }
8    /* parse first shell commands */
9    handled = FALSE;
10   silent = (bool)(*cmd=='#');
11   if (silent) {
12     cmd++; /* skip '#' */
13   }
14   res = CLS1_IterateTable(cmd, &handled, io, parseCallback); /* iterate through
         all parser functions in table */
15   if (!handled || res!=ERR_OK) { /* no handler has handled the command? */
16     CLS1_PrintCommandFailed(cmd, io);
17     res = ERR_FAILED;
18   }
19   if (!silent) {
20     CLS1_PrintPrompt(io);
21   }
22   return res;
23 }
```

Iterate Table

```
1  uint8_t CLS1_IterateTable(const uint8_t *cmd, bool *handled,
       CLS1_ConstStdIOType *io, CLS1_ConstParseCommandCallback *parserTable)
2  {
3    uint8_t res = ERR_OK;
4
5    if (parserTable==NULL) { /* no table??? */
6      return ERR_FAILED;
7    }
8    /* iterate through all parser functions in table */
9    while(*parserTable!=NULL) {
10     if ((*parserTable)(cmd, handled, io)!=ERR_OK) {
11       res = ERR_FAILED;
12     }
13     parserTable++;
14   }
15   return res;
16 }
```

Parse Table

```
1  static const CLS1_ParseCommandCallback CmdParserTable[] = {
2    CLS1_ParseCommand, /* Processor Expert Shell component, is first in list */
3    SHELL_ParseCommand, /* our own module parser */
4    #if FRTOS1_PARSE_COMMAND_ENABLED
5      FRTOS1_ParseCommand, /* FreeRTOS shell parser */
6    #endif
7    #if BT1_PARSE_COMMAND_ENABLED
8      BT1_ParseCommand,
9    #endif
10   NULL /* Sentinel */};
```

Command Parser

Shell Parser

```
1  static uint8_t SHELL_ParseCommand(const unsigned char *cmd, bool *handled,
       const CLS1_StdIOType *io) {
2    uint32_t val;
3    const unsigned char *p;
4
5    if (UTIL1_strcmp((char*)cmd, CLS1_CMD_HELP)==0 || UTIL1_strcmp((char*)cmd,
       "Shell help")==0) {
6      *handled = TRUE;
7      return SHELL_PrintHelp(io);
8    } else if (UTIL1_strcmp((char*)cmd, CLS1_CMD_STATUS)==0 ||
         UTIL1_strcmp((char*)cmd, "Shell status")==0) {
9      *handled = TRUE;
10     return SHELL_PrintStatus(io);
11   } else if (UTIL1_strncmp(cmd, "Shell val ", sizeof("Shell val ")-1)==0) {
12     p = cmd+sizeof("Shell val ")-1;
13     if (UTIL1_xatoi(&p, &val)==ERR_OK) {
14       SHELL_val = val;
15       *handled = TRUE;
16     }
17   }
18   return ERR_OK;
19 }
```

Question 1

Why is there a *NULL* at the end of the *CmdParserTable*-Array?

Question 1
Solution

Why is there a *NULL* at the end of the *CmdParserTable*-Array?

Solution

The array will be iterated until a *NULL* occurs.

Question 1

Why is the Command compared by the length of *sizeof*(”*cmpString*”) − 1?

Question 2
Solution

Why is the Command compared by the length of *sizeof*("*cmpString*") − 1?

Solution

The *sizeof*() method returns the length of a string includes the '\0'-symbol.

Question 3

Which settings you should care for by the communication between computer and device?

Question 3
Solution

Which settings you should care for by the communication between computer and device?

Solution

- Baudrate
- Databits
- Stopbit
- Parity