Presentation Assignment 2
Pointers & Arrays
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Stack:

```
0x0911
0x0910
0x090f
0x090e
0x090d
0x090c
0x090b
```
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t *p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Caution: The exact placement of the variable on the stack depends on the compiler and the chosen optimization level!
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

![Stack diagram]

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```c
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: uint8_t \( x \)
- Pointer: uint8_t * \( y \)
- Address-of operator: &\( x \)
- Indirection operator: *\( y \)

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t * p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

```
01 uint8_t a = 23;
02 uint8_t b = 42;
03 uint8_t *p = &a;
04 *p = 66;
05 p = &b;
06 *p -= 40;
07 uint8_t c = *p;
```

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Pointers

- Variable: `uint8_t x`
- Pointer: `uint8_t *y`
- Address-of operator: `&x`
- Indirection operator: `*y`

Caution: ATmega328PB has 8-bit registers and 16-bit addresses
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```

```
Stack ↓
```

```
0x090a
0x0909
0x0908
0x0907
0x0906
0x0905
0x0904
0x0903
0x0902
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;  // Highlighted line
12 y = y+2;
13 z = *y;
14 z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```c
08 uint8_t x[] = {2, 4, 8, 16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```
In depth: Arrays

- Constant pointer: `uint8_t a[]`
- Variable pointer: `uint8_t *b`
- Current element: `*b`
- x-th element: `b[x]`
- x-th element: `*(b+x)`

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7];
```
Constant pointer: `uint8_t a[]`
Variable pointer: `uint8_t *b`
Current element: `*b`
x-th element: `b[x]`
x-th element: `*(b+x)`

```c
08 uint8_t x[] = {2,4,8,16};
09 uint8_t *y = x;
10 uint8_t z = x[1];
11 z = *y;
12 y = y+2;
13 z = *y;
14 z = x[7]; // $$$
```
Hands-on: Pointers

No Screencast
Call-by-value vs. call-by-reference
Pointer and arrays
Pointer arithmetic
struct for GPS coordinates
Array of GPS coordinates
Function pointers

Can be compiled for the SPiCboard (serial console), the SPiCsim or Linux

Source code:
https://sys.cs.fau.de/extern/lehre/ss24/slp/uebung/material/pointer.c