# Research the C-language: How are variables stored and changed in memory?

Tip:

This subject is also know as ‘pointers and references’ in the C-language.

Note:

Provide a list of used source(s) for each of the below questions.

WARNING:

You might damage (low chance) your hardware by writing to certain addresses in memory. So be careful with providing the correct memory address.

Questions:

1. How to retrieve the memory address of a byte variable?   
   byte i = 0;

byte\* p;

p = &i;

printf(“Address: %d\n”, p);

1. What type of variable should you use to store the memory address of a byte variable? How do you call this kind of variables?

Byte pointer

1. How to set a byte value at a given memory address?

\*p = 1;

1. How to retrieve the byte value that is stored at a given memory address?

I = 1;

P = &I;

printf(“Value: %d\n”, \*p);

1. Answer question 1-4, but now for an int variable type.

Zelfde maar dan een int vanplaats byte

1. Explain what the advantage is of using different variable types for storing addresses of different data types (byte,int,float,double)?
2. Demonstrate the answers to above questions by extending the Arduino code in ‘Assignments/memory-fun’ as a starting point for your program. You can control the menu via input of numbers over the serial line. Search for the ‘TODO’ comments in the code. You only need to change a small amount of code. Enjoy!
3. What happens (observe your Arduino very carefully) when:  
   - you change the **value** of address 37 (0x25 hexadecimal) to 32 (0x20 , 0b00010 0000)? Lampje knippert  
   - you change the **value** of address 37 back to 0? Blinkt hij nog eens

Explain this behavior. Hint: use the datasheet of the 328p processor and the Arduino connection diagram.  
<https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>  
  
<https://www.circuito.io/blog/arduino-uno-pinout/>