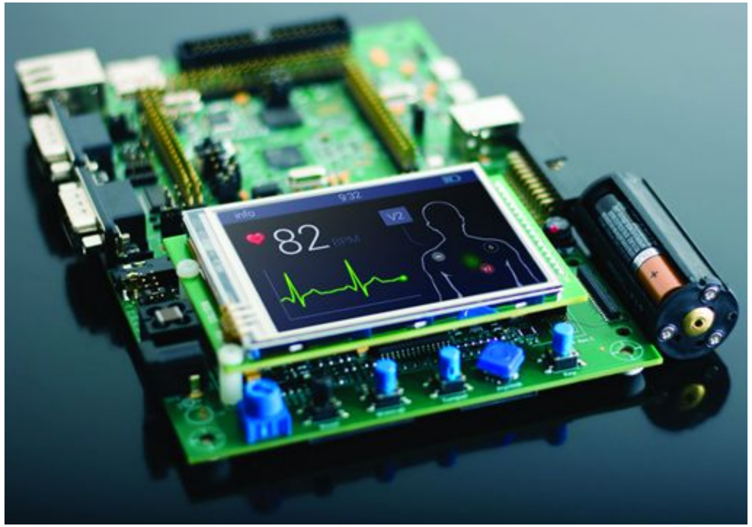
Embedded Software

CS 145/145L



Caio Batista de Melo

Defining Blocks

- whitespace doesn't matter
- blocks are defined by {}

```python
if True:
    pass
    # block of code here
```

```c
if (1) {
        // block of code here...
    // and same block here...
             // and here...
}
```

But you should still try to make your code look nice!

Variables

- static typing;
- need to declare everything beforehand!

```
x = 2
y = 2 * x
x = 2.0
z = 2 * x
```

```
int x, y;
float z;
x = 2;
y = 2 * x;
x = 2.0;
z = 2 * x;
```

Functions

- functions also need types (i.e., return type);
- parameters also need types;
- if you want to modify the arguments externally (side-effects), you need pointers!

```python
def square_num(x):
    return x * x
```

```c
int square_num(int x) {
    return x * x;
}
```

Pointers

```
void square_nums_inplace(int* x, int y) {
    *x = *x * *x;
    y = y * y;
}
int main() {
    int x = 2, y = 2;
    square_nums_inplace(&x, y);
    // x == 4 after this, but y is still 2!
    return 0;
}
```

this says we're receiving an address to an int

gets/sets the value in that address

multiplication

this gets the address of x

Arrays

- homogeneous typing
- need to declare size beforehand!
- strings are just *char* arrays, so same restrictions!
- for strings, need to make the last element == 0

```python
my_lst = []
my_lst.append(10)
my_lst.append('hello')
my_lst.append(2.5)
```

```c
int my_array[10];
my_array[0] = 10;
my_array[1] = 11;
my_array[2] = 12;
```

Loops

- for loops are similar to for in range(), there is no *for-each*
- while loops are pretty similar

```
x = 1
for i in range(1, 10):
    x *= i
```

```
int x = 1;
int i;
for (i=1; i < 10; i++) {
    x *= i;
}
```

init

condition

increment

Main Function

- your code starts in the *int main* function (like the main-block);
- it should return 0 to tell the computer that the program exited without errors
  - although in our projects we should never reach the return!

```python
def square_num(x):
    return x * x

if __name__ == '__main__':
    square_num(2)
```

```cpp
int square_num(int x) {
    return x * x;
}
int main() {
    square_num(2);
    return 0;
}
```

Source Files

```
int square_num(int);
```

C header file (square.h)

```
def square_num(x):
    return x * x
```

Python source file (square.py)

```
#include "square.h"

int square_num(int x) {
    return x * x;
}
```

C source file (square.c)

Includes

- include the .h files;
- use <> for system-library headers, “” for your own custom headers.

```c
#include "square.h"

int square_num(int x) {
    return x * x;
}
```

square.h

```c
//...
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/io.h>
//...
```

avr.h

Further Reading

Python to C

- https://www.cs.toronto.edu/~patitsas/cs190/c_for_python.html
    - https://web.cs.hacettepe.edu.tr/~bbm101/fall16/lectures/w12-c-for-python-programmers.pdf
- https://realpython.com/c-for-python-programmers/

Generic C Material

- https://www.learn-c.org
- https://www.programiz.com/c-programming
- http://www.faqs.org/docs/learnc/
- https://users.cs.cf.ac.uk/Dave.Marshall/C/

See you next time :)

Q & A