Printing Variables

Last updated 6/30/22

Printing Variables

Embedded Systems typically do not print anything

- We introduce printing to the console for 3 reasons
  - For debugging our code
  - To practice our programming (so we can see what's happening)
  - To prepare for general programming situations

Printing Variables

- C has very powerful I/O capabilities

  - Accessed by including the standard I/O library

```
#include <stdio.h>
```

Remember the < brackets > are used when accessing standard library elements

Printing Variables

- Printing a variable is relatively simple

command:	printf()	
format:	“text to display %type more text”, variable	
optional:	“\n”	prints a line feed (a new line is started)
	“\t”	prints a tab
type:	i → int, f → float, c → char	

Examples:

int ave;

ave = 12;

%i replaced by value of the variable

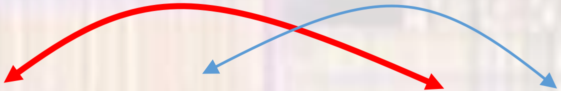
printf(“The average is %i”, ave); - prints The average is 12

printf(“%f is the average\n”, ave); - prints 12.00000 is the average and a new line is started

Printing Variables

- Each variable in a single print statement needs its own format descriptor

```
int count;
count = 21;
float ave;
ave  = 12.2;
```



```
printf(“The average is %f, with %i scores”, ave, count);
```

- prints: The average is 12.200000, with 21 scores