

Reading User Input

Last updated 7/8/21

Reading User Input

Embedded Systems typically do not read user input

- We introduce reading from 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

Reading User Input

- 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

Reading User Input

- Reading a variable is relatively simple

command: `scanf()`

format: `"%type", &variable`

the & indicates a pointer is being used

type: `i` → int, `f` → float, `c` → char

Examples:

```
int ave;
```

```
scanf("%i", &ave); //reads from the keyboard and stores the value in ave
```

```
float foo;
```

```
scanf("%f", &foo); //reads from the keyboard and stores the value in foo
```

```
char initial;
```

```
scanf(" %c", &initial); //reads from the keyboard and stores the value in initial  
// note the space before %c
```

Reading User Input

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

```
int count;
```

```
float ave;
```

```
char month;
```

```
printf("Enter an int for count, float for ave and character for month");
```

```
scanf("%i %f %c", &count, &ave, &month);
```

A diagram illustrating the mapping between format descriptors and variables in a scanf statement. A red arrow points from the '%i' format descriptor to the '&count' variable. A blue arrow points from the '%f' format descriptor to the '&ave' variable. A green arrow points from the '%c' format descriptor to the '&month' variable.

Reading User Input

- Using `scanf()` with characters can be tricky
 - Spaces, characters, tabs, newlines are characters

```
scanf("%c %c", &foo, &boo);
```

expects an input like `a b`

```
scanf("%c,%c", foo, boo);
```

expects an input like `a,b`

if you enter `a b` `scanf()` will hang – waiting on the `,`

if you enter `a, b` `boo` will have the value of a space

Placing a space before the `%` causes it to ignore any “whitespace” characters (space, tab, newline)