CSC270 Spring 2016

Circuits and Systems
Lecture Notes, Week 11

Dominique Thiébaut
dthiebaut@smith.edu
A Few Words about HW 8

Finish the Input Port Lab!
Revisiting Homework 8
Turn the Green LED On without changing the state of the others
Turn the Green LED On without changing the state of the others
Part 2

Send 50 0-1 pulses as fast as possible…
Assume E's frequency is 1MHz. Figure out a way to send 50 0-1 pulses in 100 μSec!
Part 2 Puzzle!

Figure out a way to send 50 0-1 pulses in 100 cycles!
Let’s Stop HERE!
Start the Lab!
A Few Words about HW 8

Finish the Input Port Lab!

Flexibility

General Purpose IO

Learning C

The Arduino ← download the IDE
Assume E’s frequency is 1MHz. Figure out a way to send 50 0-1 pulses in 100 uSec!
Logic

LDAA #1
STAA 8000
JMP START
General Purpose
I/O
http://www.tutorialspoint.com/compile_c_online.php

C Tutorial
References

- Online C Compiler
  - [www.tutorialspoint.com/compile_c_online.php](http://www.tutorialspoint.com/compile_c_online.php)
  - [https://ideone.com/](https://ideone.com/)

- Web Tutorial
  - [http://www.programiz.com/c-programming](http://www.programiz.com/c-programming)

- Kernighan & Ritchie
Hello World!

```c
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

```
sh-4.3$ gcc -o main *.c
sh-4.3$ main
Hello, World!
sh-4.3$ 
```
Comments

// this is a one-line comment

/* this is a comment too */
/* this is a multi-line comment */
# Variables & Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>integer variable</td>
</tr>
<tr>
<td>short</td>
<td>short integer (short int)</td>
</tr>
<tr>
<td>long</td>
<td>long integer (long int)</td>
</tr>
<tr>
<td>float</td>
<td>single precision real (floating point) variable</td>
</tr>
<tr>
<td>double</td>
<td>double precision real (floating point) variable</td>
</tr>
<tr>
<td>char</td>
<td>character variable (single byte)</td>
</tr>
</tbody>
</table>

**Qualifiers:**
- **unsigned** -> applies to ints.
- **const**    -> variable cannot be modified
Variables & Types

int -> integer variable
short -> short integer (short int)
long -> long integer (long int)
float -> single precision real (floating point) variable
double -> double precision real (floating point) variable
char -> character variable (single byte)

qualifiers:
unsigned -> applies to ints.
const -> variable cannot be modified

No Booleans!
```c
#include <stdio.h>

int main() {
    printf( "sizeof(char) = %d\n", sizeof( char ) );
    printf( "sizeof(short) = %d\n", sizeof( short ) );
    printf( "sizeof(int) = %d\n", sizeof( int ) );
    printf( "sizeof(long) = %d\n", sizeof( long ) );
    printf( "sizeof(float) = %d\n", sizeof( float ) );
    printf( "sizeof(double) = %d\n", sizeof( double ) );
    return 0;
}
```

```
sh-4.3$ gcc -o main *.c
sh-4.3$ main
sizeof(char) = 1
sizeof(short) = 2
sizeof(int) = 4
sizeof(long) = 8
sizeof(float) = 4
sizeof(double) = 8
sh-4.3$
```
#include <stdio.h>

printf( format, [variables], [literals] );

%nd    —> integer (optional n = number of columns)
%m.nf  —> float or double (optional m = number of columns,  
         n = number of decimal places)
%ns    —> string (optional n = number of columns)
%c     —> character
\n \t —> new line, tab
```
#include <stdio.h>

int main() {
    int i = 3;
    char ch = 'C'; // use ' not "

    printf("%c%d%s

", ch, i, "PO ");
    return 0;
}
```
```c
#include <stdio.h>

int main() {
    int i = 3;
    char ch = 'A';

    printf( "i = %d ch = %c ch = %d\n\n", i, ch, ch );
    return 0;
}
```

```
sh-4.3$ gcc -o main *.c
sh-4.3$ main
i = 3 ch = A ch = 65
sh-4.3$
```

(Chars are ints too!)
#include <stdio.h>

scanf( format, [variables] );

%d  —> integer (optional n = number of columns)
%f  —> float or double (optional m = number of columns, 
       n = number of decimal places)
%s  —> string (optional n = number of columns)
%c  —> character

(won't work with tutorials point's online IDE but will on https://ideone.com/)
```c
#include <stdio.h>

int main(void) {
    int i;
    float x;
    scanf("%d", &i);
    scanf("%f", &x);
    printf("i = %d  x = %1.2f\n\n", i, x);
    return 0;
}
```

Success #stdin #stdout 0s 2164KB

stdin
35
6.123456

stdout
i = 35  x = 6.12
#include <stdio.h>

char c;

c = getchar();

(won't work with tutorials point's online IDE but will on https://ideone.com/)
```c
#include <stdio.h>

int main(void) {
    char a, b, c;
    a = getchar();
    b = getchar();
    c = getchar();
    printf( "tc\n\n", a, b, c );
    return 0;
}
```
Loops

```java
while ( boolean expression ) {
    ...
}

for ( expr1; expr2; expr3 ) {
    ...
}

break; // break out of inner loop
continue; // return to beginning of
// inner loop
```
We stopped here last time...
That's the end of the Lab report
Exercise

Use a loop to print all the letters of the alphabet.
Exercise

Use a loop to print the first 10 Fibonaccis.
1, 1, 2, ...
Pointers!

```c
int a = 3;
int* p = 0;
```
int a = 3;
int* p = 0;

Pointers!
int a = 3;
int* p = 0;
p = &a;
int a = 3;
int* p = 0;

p = &a;
a = a + 1;
int a = 3;
int* p = 0;
p = &a;
a = a + 1;
*p = 5;
Arrays

```c
int arr[4];
arr[0] = 1;
arr[1] = 10;
arr[2] = 3;
arr[3] = 100;
```
int arr[4];

arr[0] = 1;
arr[1] = 10;
arr[2] = 3;
arr[3] = 100;

An array variable is a pointer to the first cell of the array.
int arr[4];
int *p;

arr[0] = 1;
arr[1] = 10;
arr[2] = 3;
arr[3] = 100;

p = arr;
*p = 99;
*(p+2) = 65;

An array variable is a pointer to the first cell of the array.
Exercise

Store the 10 first Fibonacci numbers in an array using indexes.
Exercise

Store the 10 first Fibonacci numbers in an array using pointers.
// prototypes
void hello();
int main();
...

void hello() {
  printf( "Hello World!\n" );
}

int main() {
  hello();
  return 0;
}
// prototypes
void printNice(int);
int main();

...

void printNice( int a ) {
    printf( "$%d.00\n", a );
}

int main() {
    int dollars = 100;
    printNice( dollars );
    return 0;
}
// prototypes
void increment();
int main();

...

void increment( ) {
    a = a + 1;
}

int main() {
    int amount = 100;
    increment( amount );
    return 0;
}
Exercise

Let's code Hanoi in C!!!
If-Statements

```java
if ( boolean expression ) {
    ... 
}
else {
    ... 
}
```
If-Statements

```java
if ( boolean expression ) {
    ...
} else {
    ...
}
```

Operators

```java
<  
>  
<= 
>= 
!= 
== 
! 
&& 
||
```
Exercise

Create a function called `swap()` that swaps the two ints it gets as arguments.
int* arr;

arr = (int *) malloc( 10 * sizeof(int) );
if ( arr == NULL ) {
    // stop
}
arr[0] = 1;
arr[9] = 100;
int* arr;

arr = (int *) malloc( 10 * sizeof(int) );
if ( arr == NULL ) {
    // stop
}
arr[0] = 1;
arr[9] = 100;
...
// do some more work...
int* arr;

arr = (int *) malloc( 10 * sizeof(int) );
if ( arr == NULL ) {
    // stop
}
arr[0] = 1;
arr[9] = 100;
...
// perform some operations on arr...

free( arr );
Strings

char name[] = "Sophia Smith";

printf( "name = %s\n", name );
char name[] = "Sophia Smith";
char *place = "Smith College";

printf( "name = %s\n", name );
printf( "place = %s\n", place );
Strings

char name[] = "Sophia Smith";
char *place = "Smith College";
char *st = (char *)
    malloc( 100*sizeof(char ) );
// initialize st...

printf( "name = %s\n", name );
printf( "place = %s\n", place );
printf( "st = %s\", st );
C Strings end with \0

```c
char name[] = "Smith";
```

Smith Smithh \0 ?
C Strings Are **Mutable**

```c
char name[] = "Smith";

name[4] = 't';
```

```
S m i t h \0
```

```
S m i t t t t \0
```
Exercise

Replace all the characters of "Smith College" by 'A'. Do it with indexes, and with pointers.
#include <string.h>

char name[] = "Sophia Smith";
char *place = "Smith College";
char *st = (char *)malloc( 100*sizeof(char ) );
strcpy( st, name );
strcpy( st, " " );
strcpy( st, place );
The nine most common functions in `<string.h>`:

- **strcat** - concatenate two strings
- **strchr** - string scanning operation
- **strcmp** - compare two strings
- **strcpy** - copy a string
- **strlen** - get string length
- **strncat** - concatenate one string with part of another
- **strncmp** - compare parts of two strings
- **strncpy** - copy part of a string
- **strrchr** - string scanning operation

See [http://www.tutorialspoint.com/c_standard_library/string_h.htm](http://www.tutorialspoint.com/c_standard_library/string_h.htm) for more info
Exercise

Write a C program that declares a string variable containing "Smith College" and replaces all the characters by the first letters of the alphabet. Your program must use a loop!
Let’s Stop HERE!
The Arduino
Why?

- **Natural progression:**
  - transistor
    - gate
    - comb. & seq. circuits
    - microprocessor
    - I/O ports
    - *microcontroller + C*
References

• **LYNDA**: [http://www.lynda.com/search?q=arduino](http://www.lynda.com/search?q=arduino)
  - Good for installation of software on Windows and Mac

• **Arduino's Site**: [https://www.arduino.cc/](https://www.arduino.cc/)

• **LadyAda**: [http://www.ladyada.net/](http://www.ladyada.net/)
Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.