User Input and the NXT Display

Our robots have been getting *input* from **sensors** (light, sound, touch etc.)

Many computers and software need *input* from humans (USER INPUT).

How do we get our Robot or Computer to respond to *human input*?

List some ways that the typical home computer can get input from you:

User input is often a necessary and crucial part of software design
Doing a web search, creating a password, and video games all require *user input* computer programs are constantly getting information from humans.

A Mouse, keyboards, and touch screens are typical ways that programs can collect user input.

Similarly, screens and printers output information to the user.

ON the NXT we will use **buttons,** and a **wheel** to input information

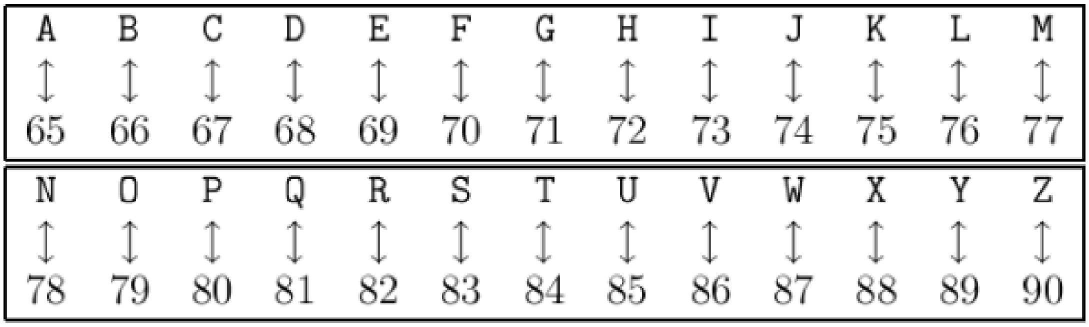
And use the NXT screen and speakers as our **output device.**

Selecting letters (Characters)

Characters or **“*char*”** are a **type a variable** like *int, string, float, etc.*

They are actually similar to an *integer* variable

“char” quantities range from0 to 256 and each number is used to represent ASCII alphanumeric codes.



A	B	C	D	E	F	G	H	I	J	K	L	M
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
65	66	67	68	69	70	71	72	73	74	75	76	77

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
78	79	80	81	82	83	84	85	86	87	88	89	90

Table 1: ASCII encodings of English capital letters.

They are declared as follows:

char a, b;

Examples:

Lets say:

char letter; ⇦ *//this is our declaration of a character variable called “letter”*

then:

letter = 100; **This char value is “d” or (little d)**
letter = 100 + 1; **This would be 101“e” or (little e)**

We can also use:

letter = ’G’; **This would be “G” (Capital G)**

*Note that when you use an ASCII value directly (like the letter G in the example above), you need to put **single** quotes around it!*

Enter the following code to see what it does:

```
task main()

{
char letter;

letter = 68;
nxtDisplayCenteredTextLine (3, "%c", letter);
wait1Msec(2000);
}
```

Now Enter this code to see what it does:

```
task main()
{
char letter;
int count;
count=1;
letter = 68;


        while (count<6)
        {
        nxtDisplayStringAt(3+(7*count), 30 , "%c", letter);
        wait1Msec(2000);
        letter = letter+1;
        count=count+1;
        }
}
```

*Print the code above with comments explaining **what the program does**, how it does it and **what each line is doing**. Use the words: ASCII, variable, display, and character in your explanation.* ***Put your name and date in the code comments***

Now using the bit of code below and/or code from your gate counter ***create a program that:***

allows a user to ***scroll*** *through the alphabet using a "wheel" device on your NXT (in order to select letters) –* ***THEY DO NOT have to select letters yet.***

When this is done, show Mr. Walzl

```
if (nMotorEncoder[Wheel] > 30)

     {
     letter=letter+1;
     nxtDisplayCenteredTextLine(3,"%c",letter);
     nMotorEncoder[Wheel]=0;
     }

if (nMotorEncoder[Wheel] < -30)

     {
     letter=letter-1;
     nxtDisplayCenteredTextLine(3,"%c",letter);
     nMotorEncoder[Wheel]=0;
     }
```

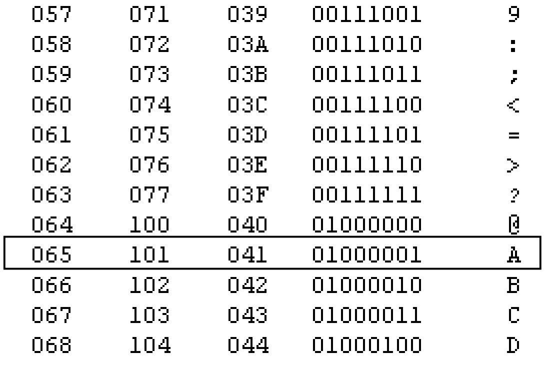
American Standard Code of Information Interchange (ASCII)

Internally, computers can only manage 0s (zeros) and 1s (ones). The circuitry involved in all modern computer processors responds <u>*only*</u> to **on** or **off.**

Using 1’s and 0’s (Binary Code) we can get computers do a number of things. The first step in communicating with a computer is creating a link between the keyboard and binary code.

To do this, *ASCII code was developed (in 1963).* It contains all the letters in the alphabet plus some additional characters. ASCII characters are always represented by the same number. For example, the ASCII code for the capital letter "A" is always represented by the order number 65, which can easily be represented using 0s and 1s in binary: *65 expressed as a binary number is 1000001.*

Here are some Examples:



Decimal			Binary	Character
057	071	039	00111001	9
058	072	03A	00111010	:
059	073	03B	00111011	;
060	074	03C	00111100	<
061	075	03D	00111101	=
062	076	03E	00111110	>
063	077	03F	00111111	?
064	100	040	01000000	@
065	101	041	01000001	A
066	102	042	01000010	B
067	103	043	01000011	C
068	104	044	01000100	D