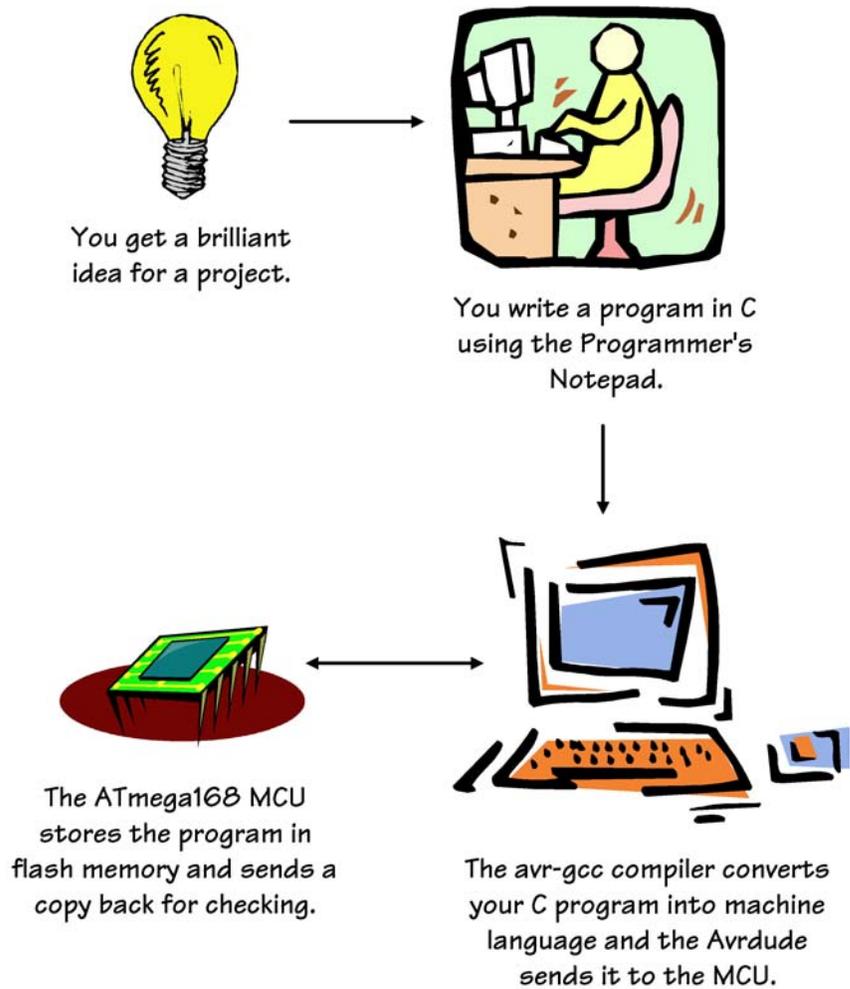# Programming the MCU
How to get your ideas into that chip!

The ATmega168 MCU has nonvolatile memory (flash memory) on board to hold the program. Therefore, when we program an MCU we say that we "flash it."

Graphically it looks like this:

You get a brilliant
idea for a project.

You write a program in C
using the Programmer's
Notepad.

The ATmega168 MCU
stores the program in
flash memory and sends a
copy back for checking.

The avr-gcc compiler converts
your C program into machine
language and the Avrdude
sends it to the MCU.

All of the software necessary for programming in the Windows environment is available in a suite of programs called Win-AVR. It can be downloaded from SourceForge at:
http://sourceforge.net/projects/winavr/files/WinAVR/20100110/

The task of programming the MCU consists of:

1. Writing the program in C using Programmer's Notepad. (Programmer's Notepad is better than just a simple text editor because it has syntax checking, indenting and color-coding of the text. All these features make your code easier to read and easier to troubleshoot.)
2. Compiling the program in a form that the MCU can understand (we're talking binary, like 1's and 0's). The C compiler called avr-gcc is also included in Win-AVR.
3. Running Makefile to download the compiled program to the MCU via a special USB to Serial cable called a flash programmer. The software downloader is called Avrdude. After the compiled program is sent to the MCU it is read back from the MCU to make sure it loaded successfully.

Set up a separate folder for each project. This folder will have the Makefile for that project and the program.c file. The name of the "program.c" file will be whatever you want to call it - typically the name you have chosen for the project.

Makefile

The "Makefile" tells the compiler which files to compile and in what order. It also tells the compiler what COM port the Nerdkits USB cable is on. Running the Makefile will program the MCU.

Rev. 9/22/15

Important DOS Commands

1. You can run the Makefile from Programmer's Notepad by selecting it from the "Tools" dropdown menu, but to get a better understanding of what's really going on I suggest running it from the DOS command line.
2. I have these files on my D drive. You may have them somewhere else. To get to the D drive (on the DOS command line) just type "d:" and hit enter.
3. If you type "dir" and hit enter you will see all the files in the current directory.
4. "cd directory" changes the current directory to what you entered in as "directory" This is an easy way to "drill down" to the directory you want.
5. Once you see the program.c and makefile you want, you can type "make" and hit enter to run the makefile.

First Program: initialload.c

This first program in the Nerdkits package simply puts up a message on the display. This initialload program is already in flash memory when you buy the kit, so it's a great way to get started. Here is the C program listing:

```
// initialload.c
// for NerdKits with ATmega168
// mrobbins@mit.edu

#define F_CPU 14745600

#include <stdio.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <inttypes.h>
#include "../libnerdkits/delay.h"
#include "../libnerdkits/lcd.h"

// PIN DEFINITIONS:
```

```c
//
// PC4 -- LED anode

int main() {
        // LED as output
        DDRC |= (1<<PC4);

        // turn on LED
        PORTC |= (1<<PC4);

        // fire up the LCD
        lcd_init();
        lcd_home();

        // print message to screen
        // 20 columns wide:
        // 01234567890123456789
        lcd_line_one();
        lcd_write_string(PSTR(" Hi Rick! "));
        lcd_line_two();
        lcd_write_string(PSTR("********************"));
        lcd_line_three();
        lcd_write_string(PSTR(" This is a test "));
        lcd_line_four();
        lcd_write_string(PSTR(" 9/16/15 "));

        // turn off that LED
        PORTC &= ~(1<<PC4);

        // busy loop
        while(1) {

        // do nothing
        }
        return 0;
}
```

Rev. 9/22/15

What Do I Expect to See?

The display has 4 rows, and each row is 20 columns wide. I expect to see the message:

Hi Rick!
****
This is a test
9/15/15

On my computer the Makefile and initialload.c are on my D drive. This is the path to the folder containing these two files:

D:\Nerdkits\Software\Code\initialload\

To run the Makefile for this project from the DOS command line, do the following:

1. Navigate to the folder that has the Makefile.
2. Once you're there, list the files just to make sure the Makefile is actually there. The "dir" command tells you all the files in the current directory (folder).
3. Type "make" and press enter to run the make command.
4. If all goes well, one of the last lines should say, "avrdude: ---- bytes of flash verified" where the "----" is a number like 7808.

Verify That the MCU was Programmed Successfully

Turn the power off, switch over to the "Run" mode (switch on pin 14), turn the power back on again and see what the display says. If it says what you were expecting, you are good to go!