PCB 111000: Introducing the Projects

It is hoped that newcomers to the C programming language and microcontrollers will find that these projects offer them a basic introduction to:

the WinAVR development system, Programmers notepad and the makefile
the C-programming language and the ATMEGA hard ware (HW)
numbers in binary and visual form and logical and numeric operations

It is also hoped that the projects will offer
A bit of fun to share with friends
The odd useful application such as a clock/ stop watch or simple calculator
Templates upon which newcomers can develop their own projects

Note: It is not intended that these projects be substituted for a C course or demonstrate the accepted or optimum way of doing things.

The Projects

Projects are grouped into nine separate directories.

Proj_1_Led_display_32_element
Proj_2_Led_display_56_element
Proj_3_numbers_and_strings
Proj_4_Data_entry_PCI
Proj_5_Clocks_and_stop_watches
Proj_6_Logic_the_EEPROM_and_more
Proj_7_Real_numbers
Proj_8_Using_the_C_maths_library
Proj_9_System_projects

The idea is that that each project should explore one new topic, however this was often found to be too restrictive. It is therefore suggested that some projects are worth revisiting later on rather than struggling with when when things are not immediately clear.

Proj_1 and 2

These cover most of the basic C and HW features to be introduced including:
The main routine and subroutines
Program flow control and structures
Defining data items and data arrays
Communicating with a PC
Strings and pointers
Displaying numbers
Timers (T0 to T2) and the watchdog timer (WDT)
The EEPROM (memory that survives power disconnection)
UART (Receiver/transmitter) for communication with a PC
The input/output pins (I/O pins)
System Reset: Power on reset (POR), WDT reset and External reset
Interrupts: Timer, WDT, UART, PCI (pin change interrupt)
I2C bus: Which is used to share data between the two devices

PCB 111000: Introducing the Projects

Proj_3_numbers_and_strings These cover the askii code and numbers in string and binary format including:

Sending symbols, text and numeric strings to the PC
Entering integers at the keyboard and converting them to binary numbers
Sending binary numbers to the eight digit display and the PC
Scientific notation
Hex numbers
Division with rounding
Trapping overflow in multiplication

Proj_4_Data_entry_PCI These cover the operation of "pin change interrupt" including

Configuring the Atmega device for PCI
Reading the three project switches responsible for generating a PCI
Delaying and restoring PCI (putting it on hold and responding later on)
Disabling PCI, Clearing PCI and switch bounce
Entering numeric data using switch presses.

Proj_5 Clocks_and_stop_watches Mainly just for fun including:

An attempt to produce something useful
A combined clock stop watch that makes more ambitious use of the I2C bus and an excuse to study mini-OS (Operating System) code in some detail.

Proj_6_Logic_the_EEPROM_and_more Odds and ends including:

Saving strings to the EEPROM for use by a client program.
Logical and bit operations
Prime number generation
Fruit machine simulation

Proj_7_Real_numbers Explores the acquisition and processing of real numbers:

Conversion of decimals numbers between 0 and 1 to binary and back again
Converting a real number into a fixed point binary number
Using twos complement to obtain a negative number
Conversion of a real number to a floating point number with 32 bits for the decimal part and eight bits for the exponent
Basic arithmetic with fixed point and floating point numbers

Proj_8_Using_the_C_maths_library Explores the use of the C library functions to

Input strings and numbers from the keyboard and print results to the screen
Input and process data from a .csv file
Use of "sprintf" to display results on the 8 digit display.
Acquire floating point numbers from the project switches

Proj_9_System_projects These include:

Re-calibration of both devices
Re-initialising the Atmega 168 EEPROM following its use for project work
Confirming correct operation of the display one digit at a time.