= = = GUIDE = = =

Congratulations on acquiring a DUO Decimal computer! This guide will describe the technical details of the DUO Decimal and how to write programs for the device.

For other DUO Decimal resources, please visit the DUO Decimal website:
www.ostracodfiles.com/decimal/menu.html

If you have any questions about the DUO Decimal, please contact me at this address:
esperantanaso@gmail.com

= TABLE OF CONTENTS =

= DUO DECIMAL HARDWARE =

The logic and memory of the DUO Decimal are stored in an ATTiny84 microcontroller. This chip contains 8 KB of flash memory to store firmware, 512 bytes of EEPROM for long-term memory, and 512 bytes of SRAM for short-term memory. Note that if you are assembling the DUO Decimal board, **the ATTiny84 should be plugged into the socket upside down with the semicircular indentation on the bottom**.

The crystal oscillator inside of the ATTiny runs at 8 MHz, but the frequency has been divided by 8 to preserve battery power. As a result, the DUO Decimal effectively runs at 1 MHz. To increase the speed to 8 MHz, you must change a fuse in the ATTiny. See the AVR programmer section of this document for instructions on how to change the fuse.

The ATTiny84 has 11 input/output pins total. These are used to communicate with external components and devices. 8 of the 11 pins may read analog input values. 3 of the 8 analog pins are free and are available through the breakout pins of the DUO Decimal board. These pins are labeled P0, P1, and P2.

The primary input of the DUO Decimal is a pair of tactile push buttons. To conserve ATTiny pins, the buttons are connected to a single analog pin through a resistor circuit.

The primary output of the DUO Decimal is an individual 7-segment LED display. The display is connected to the ATTiny by a resistor network chip.

A 6 pin male header is available for connecting an AVR programmer to the ATTiny. The AVR programmer may modify either the flash memory or EEPROM of the microcontroller.

A 3 volt 2032 coin cell battery powers the DUO Decimal. The battery should be placed in the battery holder with the positive side up. The power may be turned on or off using the power switch.

= DUO DECIMAL FIRMWARE =

While the DUO Decimal is running, it may be in one of several states:

- Code viewing mode
- Character entry mode
- Number viewing mode
- Digit entry mode
- Program running mode

When in code viewing mode or character entry mode, the user may view and modify code in SRAM. After the user is satisfied with the code, the user may save the code from SRAM to EEPROM, then switch to program running mode.

Programs are interpreted from EEPROM. While a program is running, SRAM is used to store variables.

The DUO Decimal recognizes a variety of keystrokes:

Keystroke 1 = press left, release left
Keystroke 2 = press right, release right
Keystroke 3 = press left, hold, release left
Keystroke 4 = press right, hold, release right
Keystroke 5 = press left, press right, release right, release left
Keystroke 6 = press right, press left, release left, release right
Keystroke 7 = press left, press right, release left, release right
Keystroke 8 = press right, press left, release right, release left
Keystroke 9 = press left and right, hold for 3 seconds, release left and right

Keystrokes 1 through 4 scroll through symbols. When the last symbol has been reached, the letter E is displayed.

Keystrokes 5 and 6 in code viewing mode scroll through text one line at a time. Perform keystroke 6 five times at the beginning of code to clear all code. Keystroke 5 in

character/digit entry mode will add the selected character to the text, while keystroke 6 will delete the previous character.

Keystroke 7 switches between code/number viewing mode and character/digit entry mode. Upon entering character/digit entry mode, the letter P will be displayed.

Keystroke 8 in code viewing mode or character entry mode causes the code to be saved to EEPROM. Keystroke 8 in number viewing mode or digit entry mode finishes viewing the number and continues control flow.

In program running mode, all keystrokes except keystroke 9 are ignored. Keystroke 9 switches between the code editor and code interpreter.

= DUO DECIMAL NUMERIC CODE =

Programs on the DUO Decimal are written in a language called DUO Decimal Numeric Code (DDNC). A DDNC program consists of decimal numbers separated by spaces and newlines. The first number of a line determines which command to execute. The remaining numbers are arguments to supply to the command.

Every variable stored in memory is a 4-byte floating-point binary number. Each variable may be addressed with a number between 0 and 84 inclusive. An indirect address is an address which points to a variable whose value points to another variable.

Below is a list of all DDNC commands:

00 (const) (dest addr)
Write constant to memory.
01 (src addr) (dest addr)
Copy between memory locations.
02 (src ind addr) (dest addr)
Copy from indirect address to destination.
03 (src addr) (dest ind addr)
Copy from source to indirect address.

10 (src ind addr) (len addr) (dest ind addr)
Copy list with given length from source to destination.
11 (list ind addr) (len addr) (list ind addr) (dest addr)
Store whether lists with given length are equal.
12 (val addr) (list ind addr) (len addr) (dest addr)
Find value in list with given length and store the index. If the value was not found, store negative one.
13 (val addr) (dest ind addr) (len addr)
Fill list which has the given length with a value.

20 (src addr) (src addr) (dest addr)
Add numbers.
21 (src addr) (src addr) (dest addr)
Subtract numbers.
22 (src addr) (src addr) (dest addr)
Multiply numbers.
23 (src addr) (src addr) (dest addr)
Divide numbers.
24 (src addr) (src addr) (dest addr)
Find modulus of numbers.

30 (num addr)
Increment number.
31 (num addr)
Decrement number.
32 (src addr) (dest addr)
Round value down to the nearest integer.
33 (num addr) (dest addr)
Generate a random number between 0 inclusive and given number exclusive.
34 (src addr) (dest addr)
Calculate the sine of value.

40 (src addr) (src addr) (dest addr)
Determine whether values are equal.
41 (src addr) (src addr) (dest addr)
Determine whether the first value is greater than the second value.

50 (src addr) (dest addr)
Calculate boolean NOT of value.
51 (src addr) (src addr) (dest addr)
Calculate boolean OR of values.
52 (src addr) (src addr) (dest addr)
Calculate boolean AND of values.

60 (dest addr)
Store the current command address.
61 (src addr)
Set the current command address.
62 (src addr)
Execute the following branch if the value is positive.
63 (src addr)
Execute the following branch if the value is not positive.
64
End the preceding branch.
65 (src addr)
Execute the following branch while the value is positive.

66
Break from the current while branch.
67 (dest addr)
Declare the following block as a subroutine and store a reference to the subroutine.
68 (src addr)
Call subroutine using the given reference.

70 (src addr)
Set current time.
71 (dest addr)
Store current time.
72 (src addr)
Sleep during the given number of milliseconds.
73 (src addr)
Set time correction factor. Default value is zero.

80 (src addr)
Display segments.
81 (src addr)
Display character.
82 (src addr)
Display digit.
83 (src addr)
Display number and pause so user may scroll through digits.
84 (dest addr)
Read keys.
85 (dest addr)
Prompt keystroke.
86 (dest addr)
Prompt number.

90 (mode addr) (pin addr)
Set mode of pin. Mode may be 0 for input or 1 for output.
91 (pin addr) (dest addr)
Digital read pin.
92 (pin addr) (dest addr)
Analog read pin.
93 (val addr) (pin addr)
Digital write pin.

= WRITING A PROGRAM =

This section will give step-by-step instructions for how to write and run a program for the DUO Decimal.

As an example, we will make a program which displays the number 25. The DDNC code for this program is below:

0 25 1
83 1

The first command stores 25 in variable at address 1. The second command displays the number at address 1.

Turn the DUO Decimal power on. The device will be in code viewing mode by default. The display will show the first character of the code. We want to clear the old program, so **perform keystroke 6 several times** until the display shows the letter E.

We now want to enter each character of the DDNC program into the computer. Perform **keystroke 7** to switch to character entry mode. You will see the letter P.

Use **keystrokes 1 through 4** to scroll through characters. The first character of our program is 0, so select the 0 character. Perform **keystroke 5** to enter the character into the code. The screen will flash blank for a fraction of a second. Continue to use **keystrokes 1 through 4 and keystroke 5** to enter the remaining characters of the program. A space on the DUO Decimal looks like an underscore, and a newline looks like a backwards L.

After you have finished entering the program, perform **keystroke 7** to return to code viewing mode. Use **keystrokes 1 through 4** to review the program.

If there is an extra character, scroll to the character after it. Perform **keystroke 7, 6, then 7** to delete the extra character.

If there is a missing character, scroll to after the position where the character should be. Perform **keystroke 7**, select the desired character, and **perform keystroke 5 then 7** to add the missing character.

After you have made any corrections to the program, save the code to EEPROM by using **keystroke 8**. The display will be blank for a couple of seconds while the code is being written to EEPROM.

After the program has been saved, you can run the program by performing **keystroke 9**. If everything goes well, the computer should display 2, the first digit of 25. Use **keystroke 2** to see the digit 5. To return to code viewing mode, perform **keystroke 8 or keystroke 9**.

Congratulations! You have written and run your first DDNC program.

= USING AN AVR PROGRAMMER =

If you have an AVR programmer, you can modify either the flash memory or EEPROM of the ATTiny84 in the DUO Decimal. An AVR programmer can also program a variety of other microcontrollers, so it is a useful tool to have. However, using an AVR programmer requires some knowledge of the command line interface. I purchased my AVR programmer from Sparkfun:

https://www.sparkfun.com/products/8702

Plug the 6 pin female socket into the 6 pin male headers on the DUO Decimal board so that the tab on the socket is facing to the right.

To write a text file into the DUO Decimal EEPROM, you must first convert the txt file into a hex file. You can find a Python script which does this on the DUO Decimal website. The script accepts a txt file as a command line argument and produces a file called eepromText.hex.

After you have obtained a hex file, you can transfer the contents to the DUO Decimal by using avrdude. The command I use is shown below:

avrdude -c stk500v2 -P /dev/cu.usbmodemfd121 -p attiny84 -U eeprom:w:eepromText.hex:i

You will need to specify the USB port to which the AVR programmer is connected. (In my case, the port name is /dev/cu.usbmodemfd121.) If you use a different AVR programmer than I do, change the programmer name (stk500v2).

After you have adjusted the command as necessary, turn on the DUO Decimal and run the command. If everything was successful, the code from the text file should be stored in the DUO Decimal EEPROM. You can scroll through the code to confirm this.

The DUO Decimal firmware is available on the DUO Decimal website as an Xcode project. If you are using a Mac, it should be easy to compile and transfer new firmware to the DUO Decimal. Change the port and programmer in the make file, execute **make**, and then **make flash**. If you are using Windows, you will need to transfer main.c to a new project in AVRStudio, change the project settings, then compile and upload.

You may also want to modify the fuses in the ATTiny. Fuses are used for a variety of chip settings. I use a free application called AVRFuses to change microcontroller fuses.

As mentioned earlier in this document, you can make the DUO Decimal run at 8 MHz instead of 1 MHz. To accomplish this, perform the following steps:

1. Uncheck the “divide clock by 8 internally” fuse.
2. Adjust the CLOCK attribute of the firmware make file.
3. Recompile the firmware.
4. Transfer the updated firmware to the DUO Decimal.

Again, if you have any questions about the DUO Decimal, contact me at this address: esperantanaso@gmail.com

Enjoy your new computer!