

Little Man Computer

0	1	2	3	4	5	6	7	8	9
516	319	519	902	516	922	519	922	117	319
10	11	12	13	14	15	16	17	18	19
218	715	523	922	602	000	032	001	097	032
20	21	22	23	24	25		27	28	29
001	097	000	010						
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Program Counter	09
opCode	319
Instruction	3
Address	19
Accumulator	033

An educational guide to computer operations using an Excel workbook

Little Man Computer Excel Workbook

Contents

INTRODUCTION	3
GETTING STARTED.....	3
MACROS.....	4
THE LITTLE MAN COMPUTER RIBBON.....	4
ENVIRONMENT	5
SECURITY	5
ENABLING MACROS	6
.....	6
LITTLE MAN COMPUTER FORM	6
CPU GROUP TEXT BOXES	7
<i>Program Counter</i>	7
<i>Instruction Register</i>	7
<i>Address Register</i>	7
<i>Accumulator</i>	7
SOURCE CODE GROUP BUTTONS.....	8
<i>Load</i>	8
<i>Save</i>	8
<i>Update</i>	8
<i>Assemble</i>	8
<i>Reload</i>	8
<i>Help</i>	9
PROGRAM GROUP BUTTONS.....	9
<i>Reset</i>	9
<i>Run</i>	9
<i>Stop</i>	9
<i>Step</i>	9
INPUT PROMPTS.....	9
LITTLE MAN COMPUTER GROUP BUTTONS	10
<i>Restart</i>	10
<i>Clear</i>	10
<i>Zero</i>	10
ASSEMBLER BUTTONS	10
<i>First Pass</i>	10

Little Man Computer Excel Workbook

<i>Second Pass</i>	10
<i>Format</i>	10
THE OUTPUT FORM	11
THE OUTPUT WINDOW	11
THE STATUS BAR	11
USING THE LITTLE MAN COMPUTER WORKBOOK	11
INPUT DEVICE.....	11
OUTPUT DEVICE	11
ERROR MESSAGES	11
SOURCE CODE	12
THE BUBBLE_SORT SAMPLE	13
TO RUN THE MODEL.....	13
LITTLE MAN COMPUTER - CPU SIMULATOR	13
THE MODEL CPU.....	13
MACHINE CODE INSTRUCTIONS	14
THE ASSEMBLER	14
VISUAL BASIC PROJECT	17
CHANGING THE VBA PROJECT PASSWORD	17
ASCII CHARACTER SET	18
Figure 1, Worksheet with Modeless Forms	3
Figure 2. Little Man Computer ribbon	4
Figure 3, Prompt to trust a document	5
Figure 4, Excel options	5
Figure 5, Show developer tab	5
Figure 6, Security options	6
Figure 7, Enable macros.....	6
Figure 8, Little Man Computer form	6
Figure 9, Source Code group buttons	8
Figure 10, Program group buttons.....	9
Figure 11, Little Man Computer group buttons.....	10
Figure 12, The Output Form.....	11
Figure 13, Worksheet and Form	12
Figure 14, VBA access	17
Figure 15, Project Properties	17

Little Man Computer Excel Workbook

Introduction

This workbook consists of three worksheets and two forms. There is a close link between the forms and two of the worksheets (called 'Code' and 'RAM'). The third worksheet contains some example programs which can be used to populate the code worksheet for assembling and execution. It is recommended that this document is read in its entirety to make best use of this workbook; it will also give a simple understanding in how a computer works.

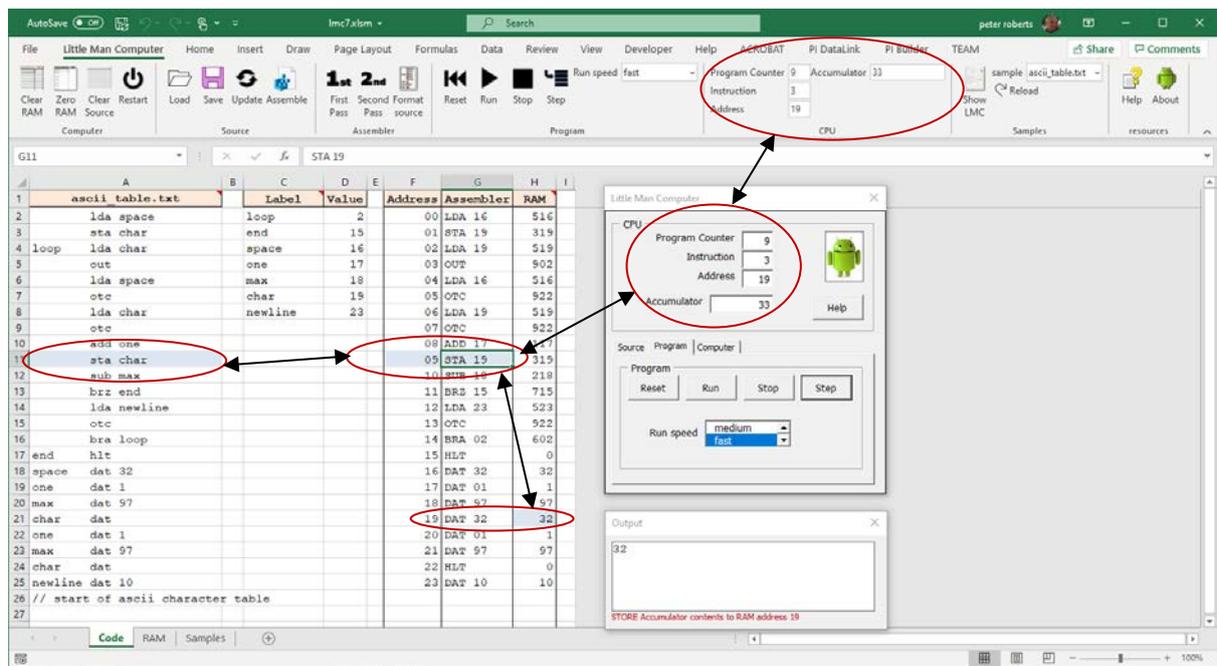


Figure 1, Worksheet with Modeless Forms

Figure 1 shows a typical display during the running of a program. The circles show the links between the form and the worksheet cells. Machine instruction translations are shown at the bottom of the output form. Error messages can be displayed here or in the application (Excel) status bar.

Getting Started

This later version of the workbook addresses some minor issues that may have been encountered in previous versions. There are also some improvements and additional features: -

- The form has a slightly different look, with buttons associated with the source code being grouped separately from the program buttons.
- A new button has been added that allows new sample code text files to be added to the **Samples** worksheet
- Some of the samples provided have been corrected and improved
- Prompts for numeric input during the execution of a program can now halt the program by clicking the Cancel button or closing the form (top right-hand 'X' button on the form). Previously it would loop for input which may be undesirable
- Cells on the code worksheet will highlight together as the program counter changes
- The latest version includes a ribbon bar which compliments the LMC form

Little Man Computer Excel Workbook

- The LMC output, which used to form part of the LMC form is now a separate form so that the ribbon can be used without the need for the LMC form (with the exception of the **Run** button which will require the LMC form to be visible)
- Error messages can now be displayed either on the status of the output form or the Application status bar (the very bottom of the Excel window). Generally, errors from the ribbon will be displayed on the application status bar and errors from the LMC form will be displayed on the output form.
- The output routine will now handle *BEL* (character 7) and *DEL* (character 127) ASCII codes. *BEL* will make a sound without adding anything to the output display; the actual sound will depend on Windows settings. *DEL* will delete the last character from the output display. Successive *DELs* will delete successive characters until none remain.

Macros

This workbook contains macros; they are unprotected and fully editable, therefore open to abuse. It may be prudent for these to be protected if students are not intended to access these. They are left exposed with the intention that students and teachers alike can examine the internal workings on this workbook and perhaps offer suggestions for improvements and observations for detected errors to the author of this workbook. Praise and recognition are also acceptable comments.

All the useful macros are available from the Little Man Computer ribbon.

The Little Man Computer ribbon

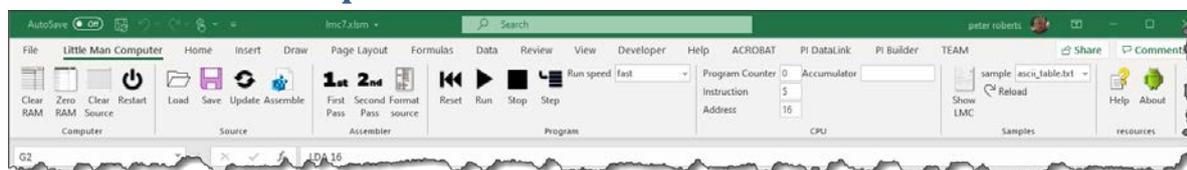


Figure 2. Little Man Computer ribbon

The buttons and features on the ribbon function the same as the buttons and features on the LMC form. See the section on the Little Man Computer form for the functions of each of the buttons. Apart from the **RUN** button it is not necessary to use the LMC form. If the **RUN** button on the ribbon is clicked, the LMC form will be displayed; this is because the ribbon will not update during the running of the macro, so the LMC form must be used to watch the computer activity; once the macro finishes the ribbon will update to match the values seen on the LMC form.

The ribbon and form will generally be in Sync (apart from use of the **RUN** button). Use of either the Form or the ribbon may be used together and separately.

The ribbon has quite a few buttons and text boxes on it, so it may be preferable to use this workbook in full-screen mode. Reducing the window size will result in the ribbon reducing and, in some cases, minimizing to save space; this is an Office feature.

Little Man Computer Excel Workbook

Environment

This Excel workbook has been tested on Windows and Mac operating systems.

On Windows™ machines, the workbook has been tested on Excel version 2007, 2010 and in Office 365.

On Mac™ machines, the workbook has been tested in Office 365.

Security

Security on Windows and Mac machines behaves differently, but on both systems the user will need to agree to the workbook macros being given the right to run. Macros need to be enabled, and on the Mac, access needs to be granted to Excel to allow some features to operate correctly; for example, clicking the help button will require Excel to access the associated PDF document which constitutes the help file.

Running the workbook from a network drive (or OneDrive, for example) may require the user to make the document a trusted document; be aware that other users on the network may have access to this file and may tamper with the macros with unpredictable consequences. It is recommended that this document be trusted (click **Yes**). It is not suggested that the tick-box to allow any network files to be trusted is ticked; the user should be aware when files are being accessed from a network.

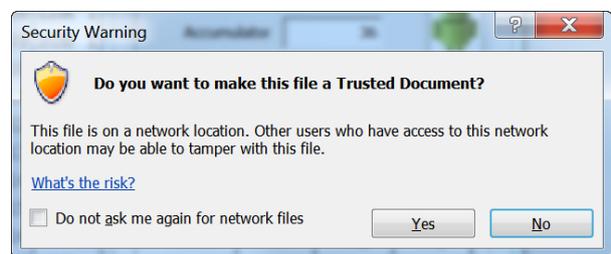


Figure 3, Prompt to trust a document

Whilst the macros can be run without using the Developer's tab, in some circumstances it may be easier. To allow access to the Developer's tab varies with the version of Excel being used. The screenshots shown in this document were made using Excel 2007; the procedure is very similar in later versions of Excel.

1. Click the top-left corner of the main excel window; a pop-up menu appears.
2. Click the button **Excel Options** in the lower-right corner of this menu; another form pops up.
3. Click the **Popular** button in the left-hand panel
4. Check the box **Show Developer tab in the Ribbon**

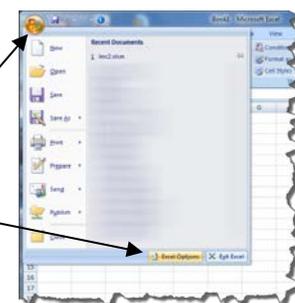


Figure 4, Excel options

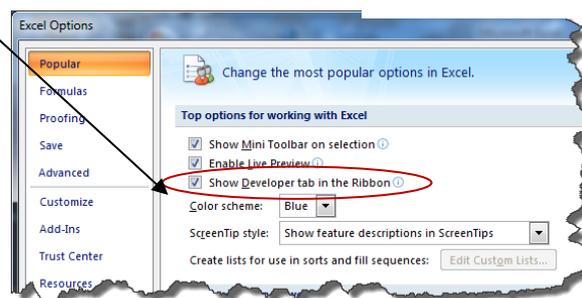


Figure 5, Show developer tab

Little Man Computer Excel Workbook

Enabling Macros

When the spreadsheet is opened, a security message will be displayed indicating that Macros have been disabled. It will be necessary for Macros to be enabled; be aware that, in general, enabling Macros may compromise the security of your machine (viruses and malware can exploit the ability to run Macros automatically). Click the **Options...** button; a dialog box will be displayed. Click the radio button **Enable this content**, then click the **OK** button.

The macro to display the Little Man computer will run and pop up a form; this form can be used along-side the spreadsheet.

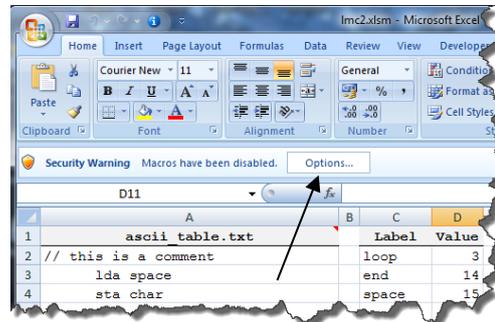


Figure 6, Security options



Figure 7, Enable macros

Little Man Computer form

The form is linked to the workbook and will manipulate the cursor according to which of the worksheets (**code** or **ram**) has been selected. Perhaps the best for demonstration purposes is to use the worksheet named **code**.

This form is shown as a *modeless* form, meaning that the worksheets can still be used while the form is in view.

If the form is ever closed (the top right-hand corner X button), it can be shown by selecting either the **code** sheet or the **RAM** sheet or by clicking the **Macros** button on the **Developer** tab; a new dialog form opens. Select the macro **showLittleManComputer** and click the **run** button. The form will close if the **samples** worksheet is selected but will re-open when either of the **code** or **RAM** worksheets are selected.

There is close interaction between the form and the worksheets (**code** or **RAM**). Changing the text in the text boxes will update the code and **RAM** worksheets accordingly. Selecting a cell in the **Address**, **Assembly** or

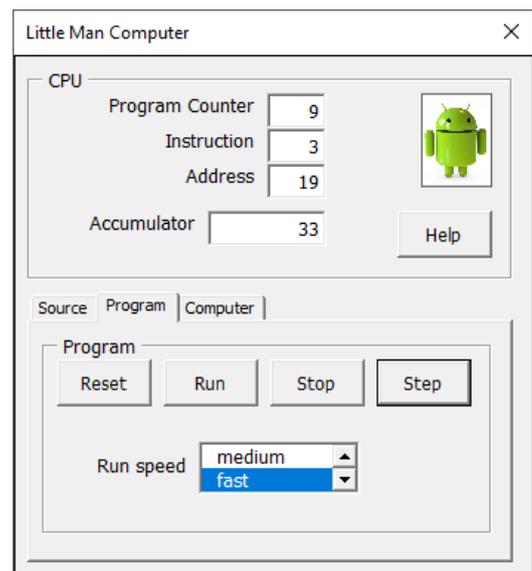


Figure 8, Little Man Computer form

Little Man Computer Excel Workbook

RAM columns of the **code** worksheet will update the **Program Counter**, **Instruction Register** and **Address Register** on the **Little Man Computer** form to match. A similar behaviour will be exhibited on the **RAM** worksheet.

CPU group text boxes

There are four text boxes which represent the main features of the model computer.

Program Counter

Using the cursor on either the **code** or **RAM** worksheets to select a memory address will update the Program Counter (PC).

Changing the Program Counter on the form will position the cursor on the appropriate cell in the selected worksheet (**code** or **RAM**). The PC on the form can be incremented or decremented by using the arrow keys; the up-arrow key will decrement the PC, while the down-arrow key will increment the PC (this will be more intuitive with the code worksheet in view, as the arrow keys will follow the cells on the worksheet).

Instruction Register

The instruction register is updated based on the selected memory address on either the **code** or **RAM** worksheets.

Changing the Instruction Register on the form will update the appropriate cell contents in the selected worksheet (**code** or **RAM**); for the **code** worksheet this will be both the **Assembler** and the **RAM** columns.

Address Register

The address register is updated based on the selected memory address on either the **code** or **RAM** worksheets.

Changing the Address Register on the form will update the appropriate cell contents in the selected worksheet (**code** or **RAM**); for the **code** worksheet this will be both the **Assembler** and the **RAM** columns.

Accumulator

The accumulator text box shows the current contents of the LMC accumulator; the contents of this can be changed or viewed during the running of the LMC program; it is also updated on the **RAM** worksheet; there is no cell on the **Code** worksheet representing the accumulator (the accumulator is only available on the **LMC** form).

Little Man Computer Excel Workbook

Source Code group buttons

The source code buttons are used to load, assemble and manage the source code column of the **code** and **Samples** worksheets.

Load

The **Load** button allows previously generated text files of source code to be quickly loaded into the source code column of the **code** worksheet. It prompts for the location of the file to be loaded; the file is expected to be a text file with an extension of "*.txt".

Save

The **Save** button will generate a text file from the source code column of the code worksheet. It prompts for the location and name of the file to be saved.

Update

The **Update** button will update the **Samples** worksheet with the source code from the **code** worksheet. If such code does not exist on the **Samples** worksheet, a new column is populated, and a new entry is added to the drop-down list on the source code group.

Assemble

The **Assemble** button generates the machine code for the source code found in the **code** worksheet. It populates the **symbol table** and the **RAM table** in the **code** worksheet and the **RAM** contents in the **RAM** worksheet; it will also set the program counter to zero (the first RAM location) and the accumulator to zero ready for program execution. The source code column will be reformatted to make it more easily read.

***Note:** The **ram** area is not automatically cleared on assembly; so, if there was a previous (longer) program loaded in ram, there may be remnants in 'memory' (this is typical computer behaviour). The user cannot assume, therefore, that when loading a new program that 'new' ram is necessarily empty or that it contains zero. If the user wishes to clear the memory or make all ram contents zero, there are two buttons on the **computer** tab for this purpose (the macro can also be found on the Excel developer's tab). The memory can be cleared (or zeroed) and the **Assemble** can be made again to repopulate the RAM with the object code.

Reload

The **Reload** button is used to reload the source code column of the RAM workbook with the currently selected code displayed in the drop-down list.

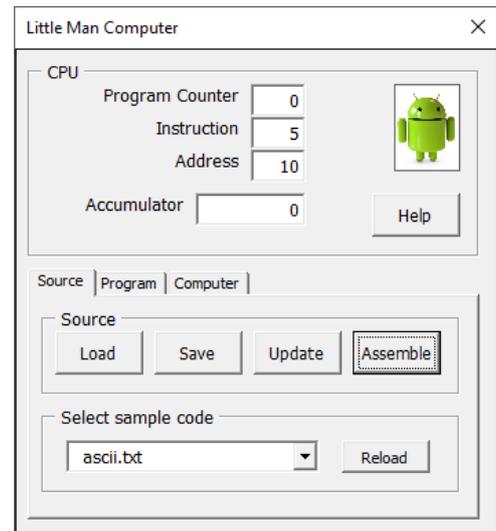


Figure 9, Source Code group buttons

Little Man Computer Excel Workbook

Help

The **Help** button will display this document if it's found in the same location as the workbook, and both the workbook and document have the same name (with only the extension being different).

Program group buttons

The program buttons are used to control the execution of the LMC program loaded into RAM. In addition, the drop-down selector can be used to set the execution speed; it has three settings.

Reset

The **Reset** button will set the *program counter to zero* (the first RAM location of the LMC program) and clear the accumulator contents (set the accumulator to zero) and clear the output display of the **Little Man Computer** form. It does NOT reset any used memory locations; if it is necessary to perform a 'full reset' this can be achieved by re-assembling the code (treat this as turning the computer off and on again).

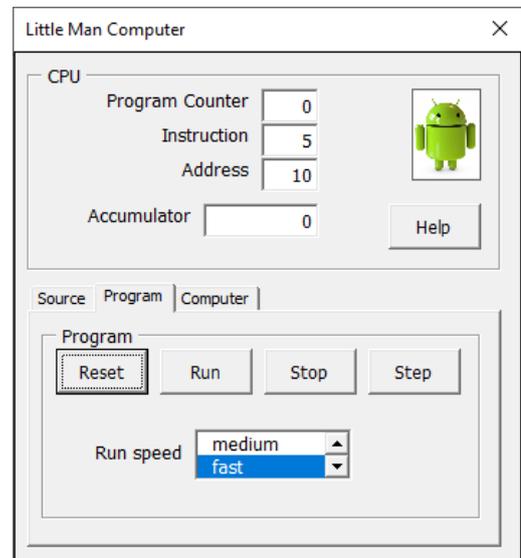


Figure 10, Program group buttons

Run

The **Run** button will start the LMC program from the current **Program Counter** position, updating the appropriate registers (**Address**, **Instruction** and **Accumulator**) as it progresses.

If the **Run** button on the Ribbon is clicked, this will force the LMC form to be displayed. This is because it's the ribbon cannot be updated during the execution of a macro; the ribbon gets updated when the macro terminates.

Stop

The **Stop** button will stop or pause the execution of the LMC program at the end of its current instruction. If a program runs away (say, for example an infinite loop executes) the stop button can be used to interrupt the execution; it can be continued by clicking **Run** or **Step**.

Step

The **Step** button will execute a single instruction in the loaded LMC program (the next instruction at the **Program Counter** location), updating the registers and RAM as necessary.

Input prompts

During the execution of the program prompts for input are made using a pop-up Windows Input box; pressing the cancel button or closing the box (top right-hand X button) will stop the program

Little Man Computer Excel Workbook

execution. Previously this would loop continuously waiting for a satisfactory response (i.e. a value from 0 to 99).

Little Man Computer group buttons

The little Man Computer buttons work with the machine rather than the programs that reside within it.

Restart

This button performs the steps that would happen when a computer is restarted; the memory would be cleared and any programs, source code, executable code and data would be cleared. This is demonstrated by leaving the **Code** worksheet empty (apart from the row headings).

Clear

This button clears the memory that would be occupied by the executable code and data. All that remains will be the source code.

Zero

This button populates the RAM cells with zeros. It will be necessary reassemble any code before the program can be executed.

Assembler buttons

These buttons will perform the individual steps in creating the machine code

First Pass

This button performs the first pass in the assembly process. It scans the source code identifying any labels and their values. The labels and their values are added to the symbol table ready for the second pass.

Second Pass

This button performs the second pass of the assembly process. It scans the source code and the symbol table and writes the machine code to RAM.

Format

This button formats the source code column for readability; it will use a non-proportional font so that the text is laid out correctly. It will require the First Pass macro to have been run because this uses the symbol table data.



Figure 11, Little Man Computer group buttons

Little Man Computer Excel Workbook

The output Form

This form represents the computer output device. This device is a serial output device, so there is no control over where the next character will be printed; it will be print at the next available position on the 'screen'.

The Output Window

The output window can contain up to five lines each containing 35 characters; once filled the window will scroll (losing any earlier output).



Figure 12, The Output Form

The Status Bar

The red text on the bottom of the form will display messages. This is mainly a translation of the next instruction to be executed but may also hold error messages.

Using the Little Man Computer workbook

A computer typically has an input stream, an output stream and an error stream. As the computer executes instructions, the input can be directed to different devices capable of providing input such as a keyboard, CD, magnetic tape, mouse, etc. The output can be directed to various output devices such as printer, disc, file, speaker or screen. If any errors occur during the execution of the program error messages are directed to a third stream such as a file or console printer for later diagnostic purposes.

Input device

This workbook represents a computer with a simple input device (represented in action by the Input dialog box prompting for input).

Output device

The output device is a serial output device (represented in action by the output form); characters are sent to this device in serial fashion. Because this device is a screen it is possible to delete the last character sent to the display; however, there is no 'intelligence' to delete other characters because detail of the screen layout is not kept. In a true screen output device, a portion of the computer memory would be used to map characters on the screen to memory locations; by writing to the appropriate memory location it would be possible to manipulate the screen display.

Error messages

If, during the execution of the program, errors are detected, messages are displayed as a message at the bottom of the output form. However, some messages may be displayed at the bottom of the application window in its status bar. Only the last error message may be visible; previous messages will be overwritten. Generally, errors emanating from the Ribbon bar will be displayed in the application status bar and errors emanating from the LMC form will be displayed in the output form status bar.

Little Man Computer Excel Workbook

Source code

The workbook comes pre-loaded with sample source code; this is in the first column of the **code** worksheet. The **samples** worksheet has many examples which can be selected from the Little Man computer form drop-down list. User-defined code can be added to this worksheet; the first row is a title row, and this will be added to the drop-down list on workbook opening, from where it can then be selected to populate the source code column of the **code** worksheet.

Once a selection is made, click the **Assemble** button to populate the rest of the **code** worksheet and populate the **RAM** worksheet.

Columns C (**Label**) and D (**Value**) in this worksheet represent a table populated by the first pass of the assembler to generate values for the labels in the source code column. In the second pass of the assembler, these values are used to convert the source code into machine code in the **RAM** column (and to the **RAM** worksheet). The **Address** and **Assembler** columns are a visual aid for the reader for comparison with the original source code.

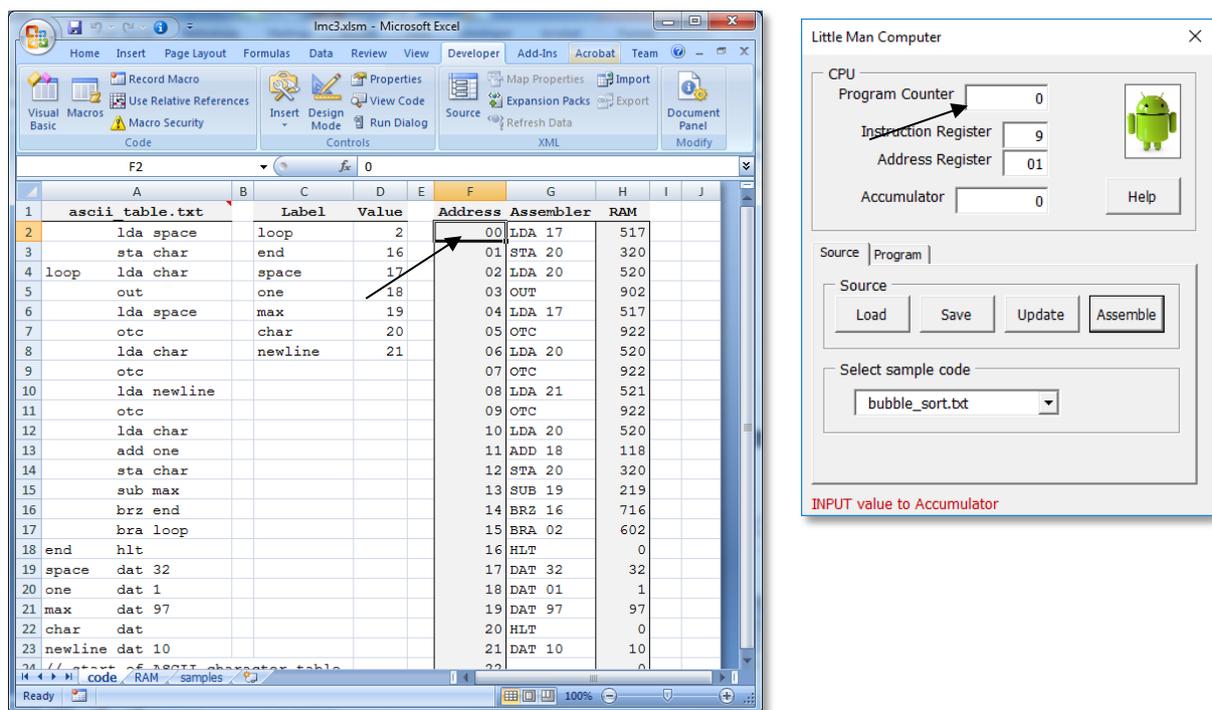


Figure 13, Worksheet and Form

Selecting a cell in the **Address** column will change the **Program Counter** of the Little Man Computer to match; setting the **Program Counter** to a number from 0 to 99 will select the appropriate address in the displayed worksheet (either **code** or **RAM**).

The **Instruction Register** and **Address Register** will update accordingly.

Little Man Computer Excel Workbook

The bubble_sort sample

The bubble-sort sample provided has some unusual features probably not to be recommended in modern computer programming standards; it includes some self-modifying code (code that can change itself during its execution) this would generate exceptions in modern computers where 'data' and 'code' are separated (partitioned). This makes it difficult to re-run it without re-assembling it. It also includes an infinite-loop situation; if, during the execution and at the prompts for numeric entry, the same number is repeated, the sort will loop trying to sort two identical numbers. It is for the student to identify and correct this 'bug'.

To run the model

- Select the **code** worksheet; this action should also show The new ribbon bar
- If the **Little Man Computer** form is not visible, click the **Show LMC** button on the **Little Man Computer** ribbon
- Select an example program from the drop-down list on the ribbon
- Click the **Assemble** button, which will convert the loaded source code file into machine code in **RAM**
- Click the **RUN** button to start execution; alternatively, the **STEP** button may be clicked to step through the program one instruction at a time.

Little Man Computer - CPU simulator

The Little Man Computer (LMC) is an instructional model of a computer, created by Dr. Stuart Madnick in 1965. It models the architecture of a simple computer and has all the basic features. See en.wikipedia.org/wiki/Little_man_computer for more information.

The web site www.peterhigginson.co.uk/LMC/ is an on-line model of LMC (with the endorsement of GCSEcomputing.org.uk); it may require JAVA to be installed for this to operate correctly, and my thanks to Peter for the inspiration for this excel version.

The model CPU

The original concept was that inside the CPU is a tiny man who runs around executing instructions encoded in a list. Inside the CPU there are:

- 100 Memory locations (numbered 0 to 99)
- An Arithmetic Unit to do calculations
- An Accumulator to store the result of the last operation or calculation
- A Program Counter to store the address (in memory) of the next instruction to do
- An Instruction Register to hold the top digit of the instruction read from memory
- An Address Register to hold the bottom two digits of the instruction read from memory
- An Input device (via a dialog box) into which a number can be typed when it's needed
- An Output device (on the **Little Man Computer** form) where any numbers output are displayed

This workbook is a representation of this model; it is entirely unprotected and is therefore open to inspection (or abuse!). The author is not responsible for any problems if changes are made to the

Little Man Computer Excel Workbook

original file. Comments for improvements are welcome and can be addressed to the author via the contact link on the web site www.petersenterprise.me.uk

The **Program Counter** holds only 2 digits (0 to 99) and the **Accumulator** holds 3 digits and a sign (-999 to 999). The Output area on the form holds a certain amount of information in five lines of 35 characters; it will then scroll (and old information is lost).

When an instruction is read from memory the top digit is used to decide what to do (a single-digit instruction code); the remaining digits form the associated address. To help us write programs more easily each instruction type is given a **mnemonic** which the **ASSEMBLE** function converts to the corresponding code.

Machine code instructions

The instructions are: -

Code	Name	Description
0	HLT	Stop (Little Man has a rest)
1	ADD	Add the contents of the memory address to the Accumulator
2	SUB	Subtract the contents of the memory address from the Accumulator
3	STA or STO	Store the value in the Accumulator in the memory address given
	DAT	This location contains numeric data; no action is taken other than to increment the Program Counter
5	LDA	Load the Accumulator with the contents of the memory address given
6	BRA	Branch - use the address given as the address of the next instruction
7	BRZ	Branch to the address given if the Accumulator is zero
8	BRP	Branch to the address given if the Accumulator is zero or positive
9	INP or OUT	Input or Output. Take from Input if address is 1, copy to Output if address is 2
9	OTC	Output accumulator as a character if address is 22. (Non-standard instruction)

Note that the LMC computer cannot tell whether memory locations contain instructions or data. Most modern computers can segment their memory (code and data being partitioned) but there must be a way to distinguish the output from one program from the instructions of another. DAT is a fake instruction to tell the assembler that this location contains data.

This implementation allows you to output both numbers (OUT) and characters (OTC). On a real computer you would only have the OTC instruction and the program would have to convert numbers to characters before printing.

The Assembler

The assembler in this workbook is a two-pass assembler; on the first pass it scans the **sourcecode** column of the **code** worksheet and identifies all the labels and calculates their values (addresses). The output from the first pass populates the **label** and **value** columns (the symbol table) of the **code**

Little Man Computer Excel Workbook

worksheet. The second pass of the assembler re-scans the **sourcecode** column of the worksheet (again) and uses the symbol table values to substitute the label for its value and then populates the **Address**, **Assembler** and **RAM** columns of the **code** worksheet with the result.

You can also put comments in the program; they are introduced with `'//'` and can appear anywhere on a line and continue to the end of the line. Blank lines (or cells), however are not permitted, and if they appear in a source code file will be ignored.

The assembler is not case-sensitive, so **Loop** and **loop** are considered identical labels; also, **LDA** and **lda** instructions are considered identical.

In addition to selecting examples from the drop-down list, the CPU can be programmed by entering numbers representing the instructions directly into memory, by typing into or modifying the assembly language area or by preparing a file containing symbolic instructions which can be loaded into the assembly language area. Click the Assemble button to convert the assembly language to numbers in **RAM**.

The drop-down list will allow you to select from the following supplied examples:

add.txt	Output the sum of two numbers. Requests input twice.
add_sub.txt	Enter three numbers. Output the sum of the first two and the third minus the first.
ascii.txt	ASCII stands for American Standard Code for Information Interchange and is the way of encoding characters so they can be stored and manipulated by computers. The original code used the numbers 33 to 126 to represent printable characters and 0-31 and 127 to represent control functions. 32 is used for space and is normally considered a printable character (even though you cannot print it). The only useful control functions on this LMC are 7 (make a sound), 10 (force a newline) and 127 (delete the last character). This program outputs the numbers 32 to 126 as characters. Other control characters will be printed as a space.
ascii_table.txt	Prints a number, a space and then the corresponding character for numbers 32 to 96.
bubble_sort.txt	Enter up to 20 numbers followed by a zero to start the sort.
fibonacci.txt	Enter a number and output that number of values from the Fibonacci series.
isSum1toN.txt	Test if the input is a triangular number; that is, one that is the sum of 1 to a number n. If the number entered is the sum of 1 to n, output n otherwise output 0.
max.txt	Input two numbers and output the higher.

Little Man Computer Excel Workbook

memory_overwrite.txt Memory overwrite illustrates one of the issues with memory being used for both instructions and data. As the program runs, numbers are entered and stored in memory locations 0 onwards. When memory location 10 is reached the instructions in the main program loop start to be overwritten with data, causing the program to stop running correctly as the LMC tries to interpret the data as an instruction. (Mike Coley)

number_store.txt This program illustrates the LMC equivalent of an array, storing each successive INPUT in successive memory locations and counting (the equivalent of the array index) how many numbers are input. The numbers are then OUTPUT in the same order they were entered when a zero is entered.

To do this the program has to alter some instructions as it runs. It does this by treating them as data (they are loaded, altered and saved) and in this program the 'data' values 350 and 550 are treated as STA (store) and LDA (load) instructions respectively by the LMC. This 'data' is altered as the program runs by adding a 1 to it so that it stores INPUT values in memory locations 50 onwards.

square.txt Output the square of a number entered.

times_table.txt Enter one number and output the times table for that number as far as the number times itself.

trinum.txt This program inputs a value (*greater or equal to zero*) and outputs which triangular number it is, or 0 if it is **not** a triangular number. For example, if the input is 15 the output will be 5 (*15 is the 5th triangular number*) and if the input is 7 the output will be 0 (*7 is not a triangular number*)

You can **STOP** and **RUN** again at any point. **STOP** pauses execution. After a **STOP**, use **RUN** or **STEP** to continue execution.

The **STEP** button executes one instruction only and is used to step through the assembled program one instruction at a time. If **STEP** is clicked while a program is running it will **pause (stop)** the program. Click **RESET** to go back to the beginning (which is always with Program Counter=0, Accumulator=0, any memory-resident data will be reset to initial values using the second pass of the assemble routine, and clear the output display on the LMC) or set the **Program Counter** to any memory address; you can also position the cursor in any **Address** cell (on either the **code** or **RAM** worksheets) to achieve the same thing.

Numbers can be entered when the running program requests it and (when the program is not running) into the individual memory (**RAM**) locations, the **Program Counter** or the Assembly Language area.

Little Man Computer Excel Workbook

Visual Basic project

To access the VBA project, select the **Developer** tab then click the **Visual Basic** button. Once the VBA project opens, click the plus sign to expand the project. This project is locked with a password in order to prevent accidental edits; enter **LMC7** (case-sensitive) in the password box and click OK to expose the VBA project.

It is not the intention of this document to explain or teach VBA (Visual Basic for Applications); there are many good books available on the subject.

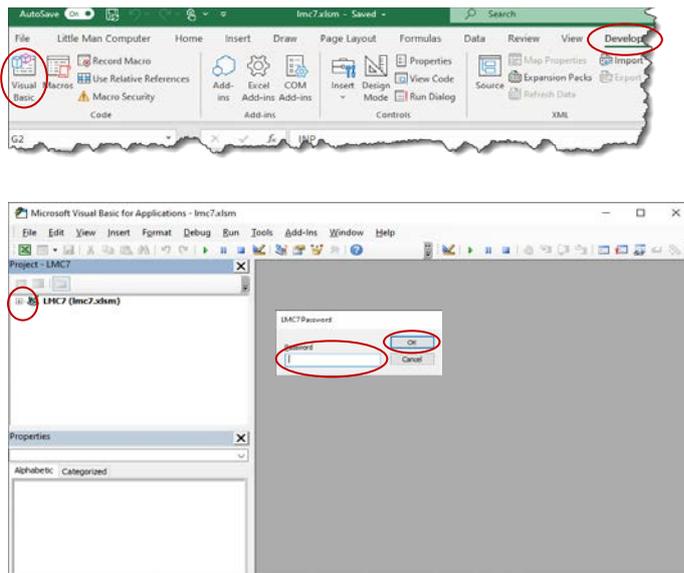


Figure 14, VBA access

Changing the VBA project password

In order to change the password, it is necessary to follow the steps above to first access the VBA project. Right-click the **LMC (lmc7.xlsm)** project and select **LMC7 properties...** from the pop-up menu. On the pop-up form select the **Protection** tab. From here the project can be unlocked by removing the tick from the box (the password will still be active) in the **Lock project for viewing** checkbox. The password can now be changed or removed entirely at the user's discretion by supplying a suitable password with confirmation before clicking **OK**.

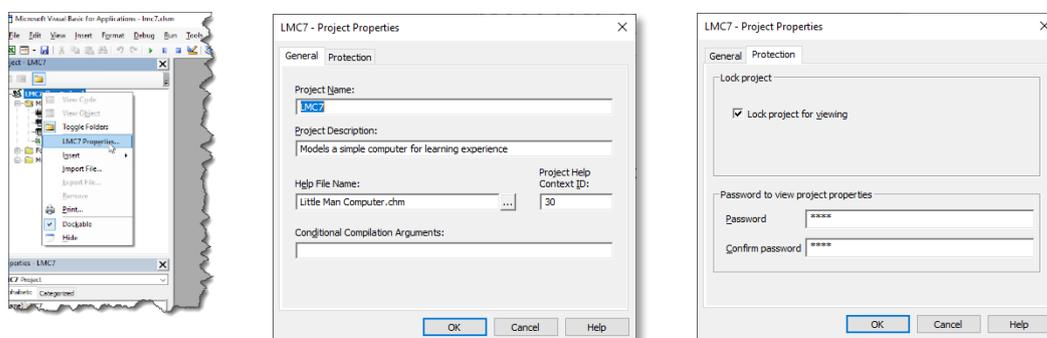


Figure 15, Project Properties

Little Man Computer Excel Workbook

ASCII Character set

Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
00	0	NUL	20	32	Space	40	64	@	60	96	`
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	\$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(48	72	H	68	104	h
09	9	TAB	29	41)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EN	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	
1D	29	GS	3D	61	=	5D	93]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL