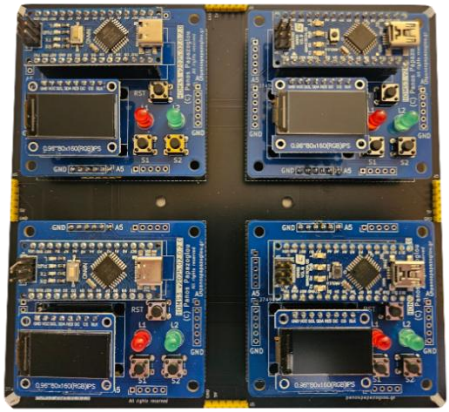
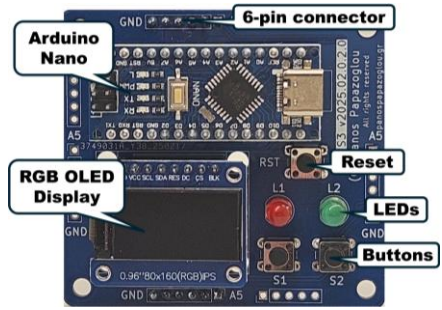
Hardware-Oriented Microcomputer Simulator Development Kit
HOMSDK v1.0

QUICK START GUIDE

The Microcomputer Development Kit (MDK) is a development platform that supports the physical implementation of custom microcomputer architectures. The user can build both internal CPU components and other microcomputer components, such as memory. The platform consists of a motherboard (2 pieces in the box) and bricks (8 pieces in the box). All bricks are identical, and the corresponding functionality can be defined by the internal software. Thus, each brick can be used as register, control unit, arithmetic and logic unit, memory, etc. For more information, visit the official web site:
https://panospapazoglou.gr

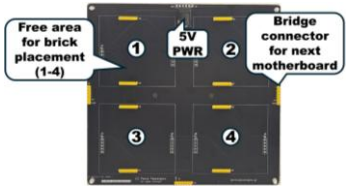




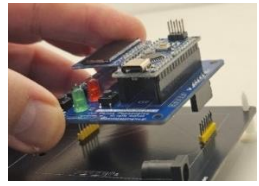
The brick! All bricks are identical, and a group of bricks constitutes a fully functioning system

The motherboard

Every brick must be installed in an available motherboard. Two motherboards can be connected using a special bridge circuit (available in the box).



Motherboard areas

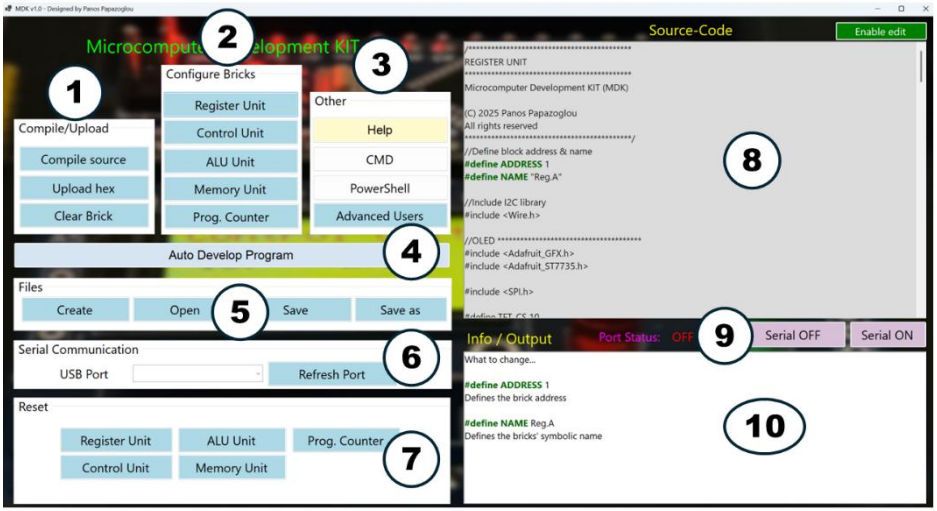


Brick installation

Bridge circuit

USING THE MDK SOFTWARE APPLICATION

The MDK bricks can be fully programmed using a specialized software which is available for download through the official MDK web site.



Through this software you can program each brick for the desired functionality. Predeveloped programs are available for each brick, but program customization can be also performed by the user. The main software offers the following operations (areas 1 to 10):

[1] Compile source code (predeveloped programs or your own programs), upload to Arduino using the USB port and clear bricks

[2] Load predeveloped brick codes for implementing Registers, Program Counter, Control Unit, ALU, memory unit, etc

[3] Advanced users can download special software tools or set upload parameters. A command prompt or a powershell can be also opened for entering Windows system instructions manually. There is also an available help text

[4] Open a special window for inserting automatically custom programs in memory brick that will be executed by the final experimental system on motherboard(s)

[5] Create your own programs or use the predeveloped programs and save to files

[6] USB port settings for Arduino communication

[7] Reset predeveloped programs to their default state if you have made any changes to the original versions

[8] Text area where the source code is displayed

[9] Get response from Arduino for monitoring system operation

[10] The info/console area displays information about the loaded source code, presents the results of the compilation/upload operations and the Arduino response

GET STARTED! – Creating your first microcomputer system

In this example, a small microcomputer system will be developed, including a Control Unit, a Memory unit and two registers (four bricks in total). Do not install any brick in the motherboard yet. The corresponding programming will be performed for each brick individually. After the programming of all bricks, the motherboard will be used for creating the final operational system.

A. Programming the bricks

Step 1. Press [Register Unit] button **(area 2)**, for loading the source code of a register (the source code will be displayed in area 8). At the same time, area 10 shows information text about the source code as well as modification suggestions. The source code is locked and if you want to make modifications, press **[Enable edit]**. If changes are made, save the modified version to the same file (**[save]**) or to a different file (**[save as]**) (**area 5**).

Step 2. Compile the source code for generating the executable file by pressing **[Compile source] (area 1)**. Compilation results are shown in **area 10**. If there are no errors after compilation, proceed to next step, otherwise make corrections, save the file and perform the compilation again.

Step 3. Connect the **brick** that will be used as **Register** to USB port and press **[Refresh Port]** (**area 6**).

Step 4. Select USB port for Arduino (usually any other port than COM1) (**area 6**)

Step 5. Press the **[Upload hex]** (**area 1**) button. Now the embedded software for register A is loaded in the corresponding brick.

Repeat steps 1 to 5 for the other bricks (buttons: **[Prog. Counter], [Memory Unit],** and **[Control Unit]**).

B. Installing the bricks

Now install each programmed brick into the motherboard in random locations (see the first indicated picture of this guide).

C. Verify the connected bricks

Now all the necessary bricks are installed in the motherboard. The Control Unit brick performs the instruction execution by fetching the corresponding data from memory brick. Moreover, this brick supports extra operations such as the automatic brick detection. Thus, the Control Unit brick can be used for identifying all the connected bricks.

Connect Control Unit brick to USB port (If you don't know which brick is the Control Unit, connect external power supply or USB connection to any brick and read the displayed information). Press **[Refresh Port]** button (area 6), select COM port and press the button **[Serial ON]** (area 9). Now the USB port status is **ON** (area 9).

Press the physical button <**S1**> (IDF selection from control unit menu) for unit auto detection and <**S1**> again to start. The scanning starts and the **register A is detected**. The auto detection results are also displayed in area 10 of the MDK software window. Follow the menu instructions for scanning the rest of the connected units. When the scanning process is completed, a corresponding message is displayed. Press physical button <**S1**> to return to main menu.

D. Instruction execution

The Memory Unit has a preloaded single Instruction (**MOV A,3** => loading the integer number 3 in register A) for execution (if the *Auto Develop Program* is used, the memory contents are overwritten with the new code after upload). For starting instruction execution, select **[EXE]** from the Control Unit brick and **watch the instruction execution both in hardware bricks and console output** (area 10).

Advanced users

By pressing the button **[Auto Develop Program]**, a special window is opened for inserting instructions in order to form an operational program that will be executed by the developed hardware system. Through this window, each instruction can be inserted directly into memory by selecting the corresponding starting address. When the program development is completed, the user must upload it in the Memory unit brick.