

# **Binary Assembler Bot v1.0**

## **Manual**

**By Billy Rebecchi**

# Contents

BAB Instruction Set .....	3
Getting Started .....	4
Writing your first program .....	5
BAB Assembly .....	5

## BAB Instruction Set

In the table is a list of Binary Assembler Bot commands. Use this reference guide to find out the equivalent of each instruction in binary, hex and BAB assembly.

Binary	Hex	Assembly	Instruction
0000	0	MLT	Move character to the left by amount
0001	1	MRT	Move character to the right by amount
0010	2	MUP	Move character up by amount
0011	3	MDN	Move character down by amount
0100	4	STP	Stop execution of the program
0101	5	BLP	Begin loop (includes number of executions)
0110	6	ELP	End loop
0111	7	PUP	Pick up item
1000	8	DRP	Drop item
1001	9	GTX	Go to x location
1010	A	GTY	Go to y location
1011	B	CLR	Set the colour of the current grid square
1100	C	JDN	Jump forward (down) a specified number of lines
1101	D	JUP	Jump backwards (up) a specified number of lines
1110	E		
1111	F		

## Getting Started

The BAB is a basic assembler robot with a 4-bit instruction set. The robot has a number of commands that allow it to move around the grid, change the colour of a grid location, pick up and drop objects and other such commands.

To get started it is first necessary to familiarise yourself with the layout of the BAB window:

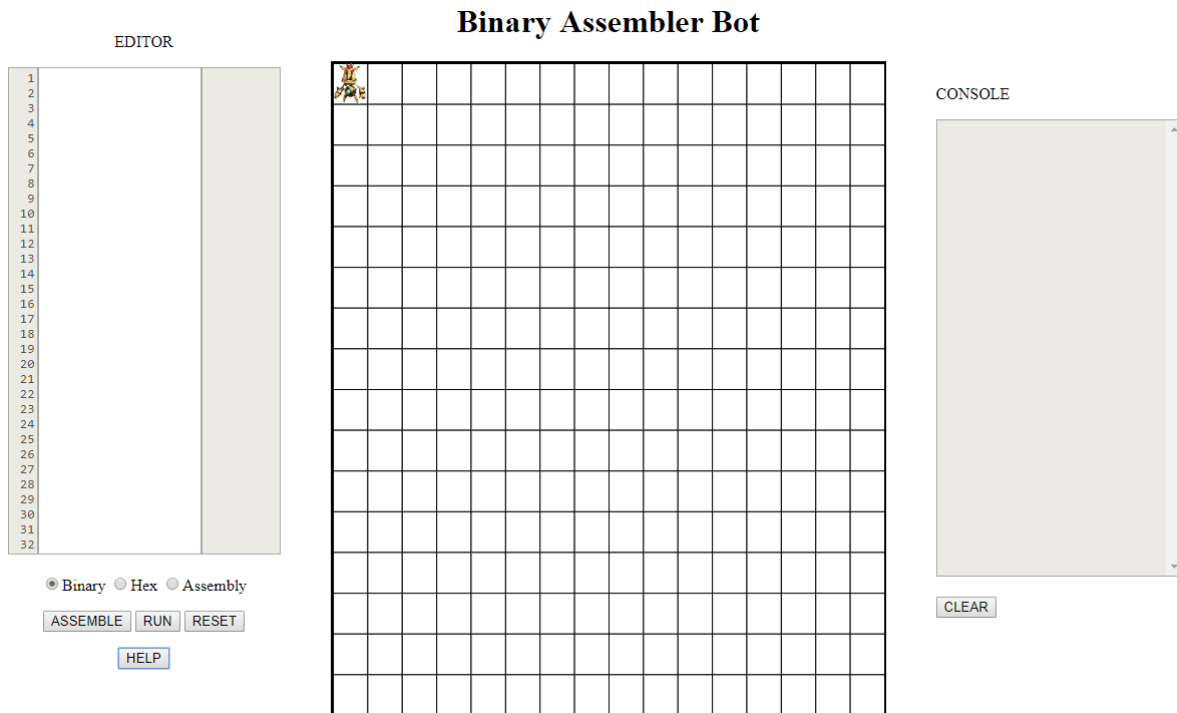


Figure 1. The BAB browser window

In figure 1, you can see the BAB browser window. This is where you can write binary, hex or BAB assembly code to control your binary assembler bot. The BAB editor is split up into three key components:

- **The editor** — This is where you can write, assemble and run code to control the BAB
- **The grid** — This is where the execution is displayed, showing the BAB running
- **The console** — This is where information about the BAB and assembler errors are shown

The three parts of the editor window will be discussed in more detail in the next section.

## Writing your first program

In this section we are going to write, assemble and run our first BAB program. We are going to write three versions of the same code, one in binary, one in hex and one in BAB assembly. To get started, and for more simplicity, we are going to write our first program in BAB assembly.

### BAB Assembly

The BAB assembly language is a purpose built language designed for controlling the BAB. The language is made up of three letter commands which give the bot individual instructions. Before writing our first program, it is important that we understand what makes up a BAB instruction.

A BAB instruction is composed of two parts, an **opcode** and an **operand**. An opcode is an instruction itself, such as move down or halt. An operand is some information that is passed alongside the opcode such as how much to move by. Look at the following instruction:

## MRT 5

This is the BAB assembly command we would type into the editor if we wanted the BAB to move right by 5 spaces. MRT is the opcode and is telling us to move right (Move Right). 5 is our operand and is telling us that we want to move right by a total of 5 grid spaces.

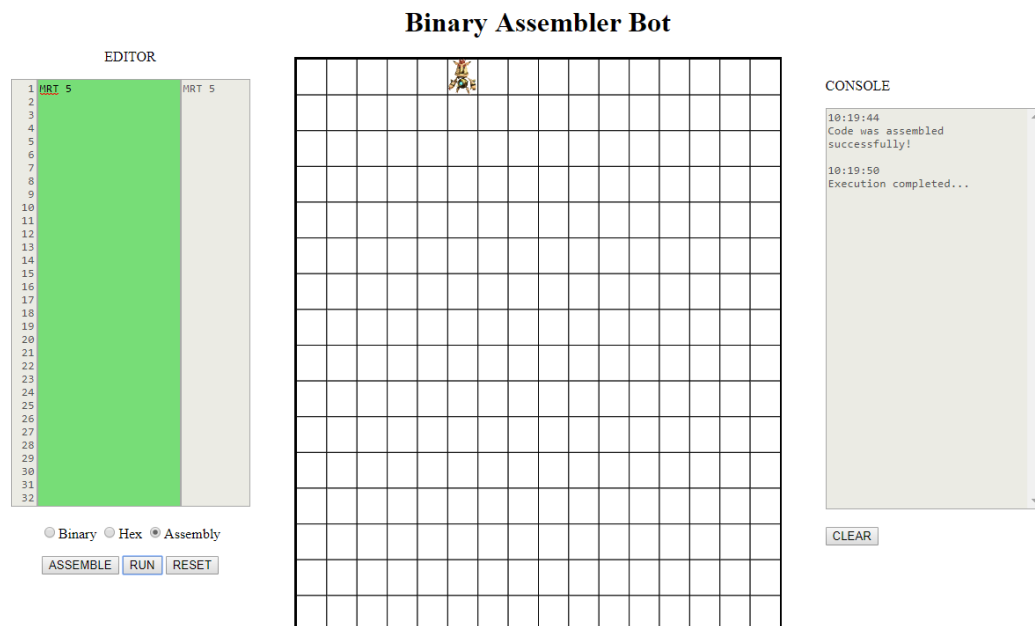


Figure 2. Successful execution of the code MRT 5

Try and run the code in the BAB editor window. Make sure to select the ***assembly*** option underneath the editor before you try to assemble, otherwise you will get an error message. Once you have assembled the code and your editor window has turned green, you will be able to run the code. Your window should look like the example in figure 2. when you have successfully run the code.