Instruction Codes

0: halt = forces the system to stop

1: moveForward = moves the robot forward in the current direction it is facing

2: rotateLeft = rotate left the direction where the robot is facing

3: rotateRight = rotate left the direction where the robot is facing

4: sense = sense what is in front of the robot where it is currently facing. The remainder 15 bits tells the robot which register to store the data.

5: jump = jump to a code line number. The remainder 15 bits identifies which line number to jump to.

6: jumpeq = jump to a code line number if the value of a register is equal to zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

7: jumpne = jump to a code line number if the value of a register is not equal to zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

8: jumpgt = jump to a code line number if the value of a register is greater than zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

9: jumplt = jump to a code line number if the value of a register is less than zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

10: jumpgte = jump to a code line number if the value of a register is greater than or equal to zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

11: jumplte = jump to a code line number if the value of a register is less than or equal to zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

11: jumplte = jump to a code line number if the value of a register is less than or equal to zero. The first 2 bits after the opcode determines which register to check against. The remainder 13 bits identifies which line number to jump to.

12: add = adds the values of the first and second register and stores it on the third register. The first 2 bits after the opcode represents the first register, the next 2 bits is for the second register and the next 2 bits for the third register. The remainder of the bits are ignored.

13: sub = subtracts the values of the first register to the second register and stores it on the third register. The first 2 bits after the opcode represents the first register, the next 2 bits is for the second register and the next 2 bits for the third register. The remainder of the bits are ignored.

14: mult = multiplies the values of the first and second register and stores it on the third register. The first 2 bits after the opcode represents the first register, the next 2 bits is for the second register and the next 2 bits for the third register. The remainder of the bits are ignored.

15: readmem = reads the value of the memory location and stores it to a register. The first 2 bits after the opcode represents the register, the remaining bits are used to specify the memory address.

16: writemem = writes to the memory location the value of a register. The first 2 bits after the opcode represents the register, the remaining bits are used to specify the memory address.

17: setreg = sets the value of a register to a constant value. The first 2 bits after the opcode represents the register, the remaining bits represent the constant value that will be stored.