# Final Black Logo BgPixelHullPixelbot Code Specification

Version 0.1

Rob Miles

# Statement format

All statements are a single, zero terminated, string.

A statement is one or more alphabetic command characters followed by a number of numeric parameters. Characters can be upper or lower case. In this document they will be expressed in upper case.

Numbers are expressed as signed decimal integers with an optional leading sign character which can be + or -. If the sign character is omitted, the value is assumed to be positive. Numbers are terminated by a non-digit character or the end of the statement. In this document a numeric value is expressed as a sequence of three lower case characters, for example ddd. The function of the sequence will be explained in the accompanying text. Any alphanumeric character (0-9 and A-Z) will be represented by the character c.

Parameters are separated by a single non-numeric character. In this document the separator is expressed as a single comma.

A comment statement starts with a hash character (#) and is ignored by the robot.

# Movement: Initial character M

## Forwards: MF

MFddd

The robot moves the number of steps given by the decimal value ddd. If the number is negative the robot moves backwards that number of steps. If the robot is already moving this command will replace the existing one. The number of steps can be omitted:

MF

The robot repeats the previous forward move. If there was no previous move the robot does not move. The movement starts moving as soon as the command is received.

The robot replies with:

OK

Note that this does not meant that the move command has been completed, rather that the robot has received and understood the command and has started moving.

## Rotate: MR

MRdd

The robot rotates clockwise the given number of steps given by the value ddd. If the number is negative the robot rotates anticlockwise that number of steps. If the robot is already moving this command will replace the existing one.

MR

The robot repeats the previous rotate. If there was no previous move the robot does not rotate.

The robot replies with:

OK

Note that this does not meant that the move command has been completed, rather that the robot has received and understood the command and has started moving.

## Check moving: MC

MC

This command can be used to determine whether the robot has completed a requested move operation. If the motors are still moving the robot replies with:

moving

If the motors are stopped the robot replies with:

stopped

## Stop robot: MS

MS

Stops any current move behaviour. The robot replies with:

OK

# Pixel control: Initial character P

## Remote Coloured Candle: PC

PCrrr,ggg,bbb

rrr red intensity in range 0-255

ggg green intensity in range 0-255

bbb blue intensity in range 0-255

Sets the pixel display to show a flickering candle of the given colour. This sets the colour of all the pixels in the display.

## Remote Set Individual Pixel: PI

PIppp,rrr,ggg,bbb

ppp number of the pixel to be set in the range 0 to n-1, where n is the number of pixels

rrr red intensity in range 0-255

ggg green intensity in range 0-255

bbb blue intensity in range 0-255

Set Individual to the given colour. The colours of the other pixels are not affected by this.

# Program Control: Initial character C

These commands are performed when the robot is running a program sequence.

## Pause when the motors are active: CA

CA

The program pauses while the motors are active. This allows a program to wait until a movement has completed.

## Delay: CD

CDddd

The program pauses for the number of *ticks* given by the decimal value ddd. The number of ticks can be omitted:

CD

The program repeats the previous delay. If there was no previous delay the program does not delay. The delay starts as soon as the command is received.

A tick is a tenth of a second.

The robot replies with:

OK

Note the reply is sent when the command has been received **not** when the robot has completed the delay. Ongoing move commands will continue to complete, and the robot will respond to other direct commands.

## Label: CL

CLcccc

This statement declares a label which can be used as the destination of a jump instruction. The label can be any number of characters and will be terminated by the end of the statement.

The robot replies with:

OK

## Jump: CJ

CJcccc

This statement causes execution of the program to continue from the given label. The label can be any number of characters and will be terminated by the end of the statement. The program must contain the label requested, or the program will stop.

If the label is found the robot replies with:

OK

If the label is not found the robot replies with:

FAIL: Destination of jump missing

## Jump when motors inactive: CI

CIccc

The program will jump to the given label if the motors are inactive.

## Measure Distance: CM

CMddd,cccc

This statement causes execution of the program to continue from the given label if the distance sensor reading is less than the given distance value. The value is given in centimetres.

The label can be any number of characters and will be terminated by the end of the statement. The program must contain the label requested, or the program will stop.

If the distance measured is greater than the given value, the program continues at the next statement.

If the label is found the robot replies with:

OK

If the label is not found the robot replies with:

FAIL: Destination of jump missing

# Remote Management: Initial character R

These commands are performed to allow a new stored program to be downloaded into the EEPROM in the robot and to control program execution by the robot drive system.

## Start Program: RS

This statement starts the execution of the current program (if present).

RS

This statement is obeyed immediately upon receipt. The program is started from the first statement in the program.

## Halt Program: RH

This statement halts the execution of the current program (if present).

RH

This statement is obeyed immediately upon receipt. The program is halted. It cannot be resumed, it must be restarted.

## Pause Program: RP

This statement pauses the execution of the current program (if present).

RP

This statement is obeyed immediately upon receipt. The program is paused. It can be resumed using the RR statement.

## Resumes Program: RP

This statement resumes the execution of the current program (if it has been paused).

RR

This statement is obeyed immediately upon receipt. The program is resumed.

## Remote Download: RM

This statement stops the execution of the current program and switches off the pixel display ready for the receipt of a new program.

RM

The program is sent as a sequence of statements that directly follow the RM command.

Note that the RM command is terminated by a Linefeed (LF) character (0x0A) as are all commands. Each statement in the program is separated from the next by the LF character. The end of the program is specified by a NULL character (0x00) which is sent **after** the LF that terminates the last statement in the program.

After the NULL character a single byte checksum is sent. This is the modulus 256 total of all of the program code characters that have been sent.

If the program was successfully received it is executed immediately.