(Unofficial) API Guide for mDrawBot mScara

Version 0.8 2015.11.03 rgrokett

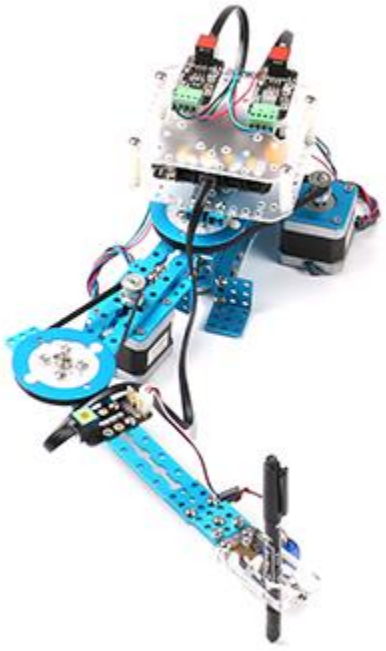
This is a guide to the drawing API used by the MakeBlock mDrawBot kit mScara robot.
http://www.makeblock.cc/

It's a reverse engineering of their scara.ino and mDraw v1.1 python from Github (Aug.25,2015 commit).

https://github.com/Makeblock-official/mDrawBot

This documents purpose is to provide a guide for creating alternative programs to mDraw, such as linux command line tools or interconnection to other CNC type plotter programs.

Note that since mScara isn't an X/Y Plotter, the coordinates require conversion to the arm coordinates to correct for distortion, same as your brain has to do to control your hand. This is done in the scara.ino software, but could require you to fine tune X/Y coordinates sent to mScara to account for distortion.



When you use the "Update Firmware" function in the mDraw program, you are actually downloading a compiled hex version of the scara.ino program to the Arduino Orion. This program handles the conversion of a subset of CAD Gcode into the stepper motor arm & pen motion.

Abbreviation	Description
CAD	Computer Aided Design
CNC	Computer (or computerized) numerical control

	https://en.wikipedia.org/wiki/Numerical_control
G-Code	G-code is a language in which people tell computerized machine tools how to make something. https://en.wikipedia.org/wiki/G-code

Once the firmware is updated into Arduino Orion, it will remain even after power off. When you power up the Arduino, it will automatically run the program again, open its USB serial port and await the Gcode commands shown below.

List of available mScara commands:

Command	Sub	Function	Example
M1		Pen Up/Down	M1 140 = *Down* M1 160 = *Up*
M2		Save Pen Up/Down positions	M2 U160 D140
M3		Set Stepper Aux Delay	M3 0
M4		Set Laser Power	M4 50 to M4 100
M5		Save Robot EPROM settings	M5 A0 B0 M168 N206 D50 *Note: requires COM port to be closed/reopened to reinitialize*
	A	Motor A Direction (forward/reverse)	A0 or A1
	B	Motor B Direction (forward/reverse)	B0 or B1
	M	Arm L1 length mm	M168
	N	Arm L2 length mm	N206
	D	Speed 50-80%	D50 to D80
M10		Return EPROM settings	*Send*: M10 *Recv*: M10 MSCARA 168 206 -373.99 0.00 A0 B0 S60 U160 D140 Arm L1 Arm L2 Home X Home Y plus below:
	A	Motor A Direction (forward/reverse)	A0/1
	B	Motor B Direction (forward/reverse)	B0/1
	S	Speed 50-80%	S50 to S80
	U	Pen Up position	U160
	D	Pen Down position	D140

G1		XYZ Move	G1 X-205.55 Y175.75 A0 *Note that positions are not true X/Y due to arm movement. Requires math conversion.*
	X	X position	X0.0 to X-375.0 (approx. range in mm)
	Y	Y position	Y0.0 to Y375.0 (approx. range in mm)
	Z	Z position (not used)	
	F	Stepper speed mm/min	Defaults to M5 D command
	A	Stepper aux delay factor	A0
G28		Move to Home	G28
P1		Return current X Y position	

Below is a sample of a typical transmission from the PC to the Arduino Orion. The Orion converts these commands into the pen motion.

Sample sequence

COMMAND	DESCRIPTION
M4 0	*Set Laser power to 0 (optional)*
M10	*Retrieve EPROM settings*
M1 130	*Pen Up*
G1 X-205.55 Y175.75 A0	*Move to location*
M1 90	*Pen Down*
G1 X-195.55 Y185.33 A0	*Draw to location*
... (more G1 pos)	
M1 130	*Pen Up*
G1 X-221.53 Y123.25 A0	*Move to location*
M1 90	*Pen Down*
...	
G1 X-231.53 Y123.25 A0	*Last Draw point*
M1 130	*Pen Up*
G28	*Move to Home*

Example Usage:

An example of using these commands follows below. To send these commands, I used a Linux server connected to the Orion via USB. This is for Command Line interface, not for GUI.

***NOTE**: If you use a Raspberry Pi, you must either leave the Orion's Power Switch turned ON or use a Powered USB Hub. The Raspi does not deliver enough current to keep the Orion powered up by itself.*

***NOTE**: You must have previously installed and operated the mScara using mDraw software on a PC so that you know the mScara is working properly before proceeding. Also sets the default EPROM settings.*

1. Plug in the USB cable between the Linux server and your Arduino Orion.
   Note that this port must be able to run at 115200 baud. Not all servers can do that!

2. On the Linux server:
   `$ ls –lart/dev/tty*`
   Look for the last entry (newest date). This would be the USB serial port.
   `crw-rw---- 1 root dialout 188,  0 Nov  1 19:01 /dev/ttyUSB0`
   On Raspi, it would be something like `/dev/ttyACM0`

3. Create a test python program to talk to the port: (substitute your port)
   `$ nano porttest.py`

```
import serial, time
ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=20)
time.sleep(2)  # just give some time for port open
ser.write('M10' + '\n')
print ("EPROM:"+ser.readline())
print (ser.readline())
print ("FINISHED")
ser.close()
```

   Be sure the mScara motor switch is turned on and previously tested using mDraw.
   Then run the test:

   `$ sudo python ./porttest.py`

   You should see something like:
   `EPROM:M10 MSCARA 168 206 -373.99 0.00 A0 B0 S60 U160 D135`

   `OK`

4. There are several other test programs available in the API guide github directory at:
   https://github.com/rgrokett/mdrawbotAPI