

Swift Pro Protocol

1)Introduction

- uArm Gcode is an important part of the uArm software.
- Based on the standard gCode protocol, we add a new protocol head in front of the Gcode so that it can be more easily to use and debug.
- What's more, it is designed to be compatible with the standard Gcode. (We offer the code of decode the standard Gcode)

2)Example

- Sending command from PC
"#25 G0 X180 Y0 Z150 F200"
//move to [180,0,150] with the speed 200mm/min
- Reply from uArm "\$25 ok"

3)Commands(TBD)-

Command can be divided into two parts:

Command with underline: it's the new added protocol head.

- The command from PC starts with '#', while the command from uArm starts with '\$'.
- And the data following the symbol decided by the PC, and the reply from the uArm should have the same data which indicates it finish the command. (In the example above, PC sends the command with '#25' and uArm replies the command with'\$25')

Command without the underline: it's the standard Gcode.

Caution :

1. There should be blank space between each parameter;
2. The letters in the command should be capitalized;

GCode Command (v1.2)	Description	Feedback	Remarks
<p>1. #n is used for the debug, if you don't want to use it please remove it directly.</p> <p>(For Example: G2202 N0 V90\n)</p> <p>2. '\n' is the symbol of line feed.</p>			
Moving Command (parameters are in underline)			
#n G0 X <u>100</u> Y <u>100</u> Z <u>100</u> F <u>200</u> \n	Quick positioning, Move to XYZ(mm), F is speed(mm/min) , F= 0~200	\$n ok \n \$n Ex \n (refer to Err output)	
#n G1 X <u>100</u> Y <u>100</u> Z <u>100</u> F <u>100</u> \n	Linear interpolation, Move to XYZ(mm), F is speed(mm/min) , F= 0~200	\$n ok \n \$n Ex \n (refer to Err output)	Fix V3.2.0 interface, delete laser mode, G1,G0 control laser.
#n G2004 P <u>1000</u> \n	Delay microsecond	\$n ok \n \$n Ex \n (refer to Err output)	

# <u>n</u> G2201 S <u>100</u> R <u>90</u> H <u>80</u> F <u>100</u> \n	Polar coordinates, S is stretch(mm), R is rotation(degree),H is height(mm), F is speed(mm/min), F=0~200	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> G2202 N <u>0</u> V <u>90</u> F <u>100</u> \n	Move the motor to the position ,N is ID of joints(0~3),V is angle(0~180) , F is speed(mm/min), F=0~200	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> G2204 X <u>10</u> Y <u>10</u> Z <u>10</u> F <u>100</u> \n	Relative displacement	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> G2205 S <u>10</u> R <u>10</u> H <u>10</u> F <u>100</u> \n	Polar coordinates for relative displacement	\$n ok \n \$n Ex \n (refer to Err output)	

# <u>n</u> G2206 B <u>90</u> L <u>70</u> R <u>50</u> F <u>100</u> \n	Move the motor to the position ,B is base motor,L is left motor, R is right motor, angle(0~180) , F is speed(mm/min), F=0~200	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.5.0 or later
# <u>n</u> G2207 N0 F <u>100</u> \n	The manipulator moves axially N is the axial 0: x axis 1: y axis 2: Z axis F : velocity and direction F100: forward 100 speed F-100: Reverse 100 speed	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.9.6 or later
# <u>n</u> G2208 N0 F <u>100</u> \n	Motor joint motion N is the motor ID (0~2) F : velocity and direction F100: forward 100 speed F-100: Reverse 100 speed	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.9.6 or later
System Command (parameters are in underline)			

# <u>n</u> S1000 V <u>0</u>	Control Arm motion 0: suspend motion 1: restart motion	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.5.0 or later
# <u>n</u> S1100	motion control reset	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.5.0 or later
Setting Command (parameters are in underline)			
# <u>n</u> M17\n	Attach all the joint motors	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M204 A <u>1.3</u> \n	Set accelerations and save, A=0~5, large accelerations maybe cause out of step, suggest set as1.3	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2019\n	Detach all the joint motors	\$n ok \n \$n Ex \n (refer to Err output)	

# <u>n</u> M2120 V <u>0.2</u> \n	Set time cycle of feedback, return Cartesian coordinates, V is time(seconds)	\$n ok \n \$n Ex \n (refer to Err output) @3 X154.71 Y194.91 Z10.21 R90\n	
# <u>n</u> M2121\n	Stop feedback	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2122 V <u>1</u> \n	Report (@9 V0) when stop. V1: Enable V0: Disable	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2123 V <u>1</u> \n	closed-loop stepper system. V1: Enable V0: Disable	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.2.0 or later
# <u>n</u> M2201 N <u>0</u> \n	Attach motor, N is ID of joints(0~3)	\$n ok \n \$n Ex \n (refer to Err output)	

# <u>n</u> M2202 <u>N0</u> \n	Detach motor, N is ID of joints(0~3)	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2203 <u>N0</u> \n	Check if the motor is attached, N is ID of joints(0~3)	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2210 <u>F1000</u> <u>T200</u> \n	buzzer,F is frequency, T is time (ms)	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2211 <u>N0</u> <u>A200</u> <u>T1</u> \n	Only support Read External EEPROM ,N1 is USR_E2PROM, A is address(0-65524), T is type (1 char,2 int,4 float)	\$n ok <u>Vx</u> \n \$n Ex \n (refer to Err output)	Support v4.7.0 or later

# <u>n</u> M2212 N <u>0</u> A <u>200</u> T <u>1</u> V <u>10</u> \n	Only support Write External EEPROM ,N1 is USR_EEPROM, A is address(0-65524), T is type (1 char,2 int,4 float)V is the input data	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.7.0 or later
# <u>n</u> M2213 V <u>0</u> \n	Default function of base buttons (0 false, 1 true)	\$n ok \n \$n Ex \n (refer to Err output)	This interface does not support temporarily
# <u>n</u> M2215\n	Reset Grbl parameter	\$n ok \n \$n Ex \n (refer to Err output)	Add reset param interface
# <u>n</u> M2220 X <u>100</u> Y <u>100</u> Z <u>100</u> \n	Convert coordinates to angle of joints	\$n ok B50 L50 R50\n (B joint 0,L joint 1,R joints 2, 0~180) \$n Ex \n (refer to Err output)	

# <u>n</u> M2221 B <u>0</u> L <u>50</u> R <u>50</u> \n	Convert angle of joints to coordinates	\$ <u>n</u> ok X <u>100</u> Y <u>100</u> Z <u>100</u> \n \$ <u>n</u> Ex \n (refer to Err output)	
# <u>n</u> M2222 X <u>100</u> Y <u>100</u> Z <u>100</u> P <u>0</u> \n	Check if it can reach,P1 polar, P0 Cartesian coordinates	\$ <u>n</u> ok V1 \n (1 reachable, 0 unreachable) \$ <u>n</u> Ex \n (refer to Err output)	
# <u>n</u> M2231 V <u>1</u> \n	pump V1 working, V0 stop	\$ <u>n</u> ok \n \$ <u>n</u> Ex \n (refer to Err output)	
# <u>n</u> M2232 V <u>1</u> \n	gripper V1 close, V0 open	\$ <u>n</u> ok \n \$ <u>n</u> Ex \n (refer to Err output)	
# <u>n</u> M2233 V <u>1</u> \n	laser V1 working, V0 stop	\$ <u>n</u> ok \n \$ <u>n</u> Ex \n (refer to Err output)	Add laser interface

# <u>n</u> M2234 V <u>1</u> \n	Enable/disable Bluetooth (1:enable, 0:disable)	\$n ok \n \$n Ex \n (refer to Err output)	This interface does not support temporarily
# <u>n</u> M2240 N <u>1</u> V <u>1</u> \n	Set the digital IO output	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.3.0 or later
# <u>n</u> M2241 N <u>1</u> V <u>1</u> \n	Set the digital IO direction (V1 Output; V0 Input;)	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.3.0 or later
# <u>n</u> M2245 V <u>btname</u> \n	Set the name of Bluetooth, 11 letters limited	\$n ok \n \$n Ex \n (refer to Err output)	This interface does not support temporarily
# <u>n</u> M2400 S <u>0</u> \n	Set the mode of arm (0: Standard 1:Laser 2:3D printing 3:Universal Holder 4: Pro 5: Plus 6: Touch Pen)	\$n ok \n \$n Ex \n (refer to Err output)	

# <u>n</u> M2401 \n	Set the current position into the reference position	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2410 \n	Set the height zero point	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2411 S <u>100</u> \n	Set the offset of end-effector (mm)	\$n ok \n \$n Ex \n (refer to Err output)	
# <u>n</u> M2412 V <u>10</u> \n	Set the offset angle of end-effector(°)	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.2.0 or later
# <u>n</u> M2413 H <u>100</u> \n	Set the offset high of end-effector (mm)	\$n ok \n \$n Ex \n (refer to Err output)	Support v4.9.0 or later
Querying Command (parameters are in underline)			
# <u>n</u> P2200 \n	Get the current angle of joints	\$ <u>n</u> ok B <u>50</u> L <u>50</u> R <u>50</u> \n \$n Ex \n (refer to Err output)	

# <u>n</u> P2201\n	Get the device name	\$ <u>n</u> ok <u>SwiftPro</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2202\n	Get the hardware version	\$ <u>n</u> ok <u>V3.0.1</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2203\n	Get the software version	\$ <u>n</u> ok <u>V4.0.0</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2204\n	Get the API version	\$ <u>n</u> ok <u>V4.0.1</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2205\n	Get the UID	\$ <u>n</u> ok <u>V0123456789AB</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2206 <u>N0</u> \n	Get the angle of number 0 joint (0~3)	\$ <u>n</u> ok <u>V80</u> \n \$n Ex \n (refer to Err output)	Add get the angle of end-effector interface

# <u>n</u> P2220\n	Get current coordinates	\$ <u>n</u> ok X <u>100</u> Y <u>100</u> Z <u>100</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2221\n	Get current polar coordinates	\$ <u>n</u> ok S <u>100</u> R <u>90</u> H <u>80</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2231\n	Get the status of pump	\$ <u>n</u> ok V <u>1</u> \n (0 stop, 1 working, 2 grabbing things) \$n Ex \n (refer to Err output)	
# <u>n</u> P2232\n	Get the status of gripper	\$ <u>n</u> ok V <u>1</u> \n (0 stop, 1 working, 2 grabbing things) \$n Ex \n (refer to Err output)	

# <u>n</u> P2233\n	Get the status of limited switch	\$ <u>n</u> ok V <u>1</u> \n (1 triggered, 0 untriggered) \$n Ex \n (refer to Err output)	
# <u>n</u> P2234\n	Get the status of power connection	\$ <u>n</u> ok V <u>1</u> \n (1 connected, 0 unconnected) \$n Ex \n (refer to Err output)	
# <u>n</u> P2240 N <u>1</u> \n	Get the status of digital IO	\$ <u>n</u> ok V <u>1</u> \n (1 High, 0 Low) \$n Ex \n (refer to Err output)	Support v4.3.0 or later
# <u>n</u> P2241 N <u>1</u> \n	Get the status of analog IO	\$ <u>n</u> ok V <u>295</u> \n (return the data of ADC) \$n Ex \n (refer to Err output)	Support v4.3.0 or later

# <u>n</u> P2242\n	Get the default value of AS5600 in each joint	\$ <u>n</u> ok B <u>2401</u> L <u>344</u> R <u>1048</u> \n \$n Ex \n (refer to Err output)	
# <u>n</u> P2243\n	Get the current value of AS5600 in each joint	\$ <u>n</u> ok B <u>2401</u> L <u>344</u> R <u>1048</u> \n \$n Ex \n (refer to Err output)	Support v4.2.0 or later
# <u>n</u> P2244\n	Get the communication status of AS5600 in each joint	\$ <u>n</u> ok V0\n \$n E26 V <u>x</u> \n	Support v4.7.0 or later
# <u>n</u> P2400\n	Check current status	\$ <u>n</u> ok V <u>1</u> \n (0: Standard; 1:Laser;2:3D printing ;3:Universal Holder ;4: Pro; 5: Plus; 6: Touch Pen;) \$n Ex \n (refer to Err output)	
Even report			

@1	Ready		
@3 X10 Y20 Z10 R90\n	Timed feedback , "M2120"		
@4 N0 V1\n	Report the button event. N: 0 = Menu button, 1 = Play button V: 1 = Click, 2 = Long Press		This interface does not support temporarily
@5 V1\n	Report event of power connection		This interface does not support temporarily
@6 N0 V1\n	Report event of limit switch in end-effector		
@7 temp error	Temperature error in 3D printing		This interface does not support temporarily
@9 V0\n	Stop movement		
Err Output			
E20	Command not exist		
E21	Parameter error		
E22	Address out of range		

E23	Command buffer full		
E24	Power unconnected		
E25	Operation failure		
E26	Encoder communication failed		
E27	Motor disable		

Different modes for uArm Swift Pro

Since different types of the end-effectors have different length and height, so we designed the command M2400,

which could help us to fit the uArm into different situations easily. With this command, there is no need to concern

about how to adjust the parameters for different situations.

Currently we offer 4 kinds of mode:

M2400 S0: Standard Suction mode (end-effector tools: Servo suction)

M2400 S1: Laser mode (end-effector tools: laser)

M2400 S2: 3D printing mode (end-effector tools: hot end)

M2400 S3: Universal holder mode (end-effector tools: universal holder)

M2400 S4: Pro Suction mode (end-effector tools: flat stepper suction)

M2400 S5: Plus Suction mode(end-effector tools: standard stepper suction)

M2400 S6: Touch Pen mode(end-effector tools: universal holder)

M2400 S7: User-Defined the End-Effector Offset Mode