Swift Pro Potocol

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
1. #n is used for the debug, if you don't want to use it please remove it directly. (For Example: G2202 N0 V90\n) 2. '\n' is the symbol of line feed.			
Moving Command (parameters are in underline)			
#n G0 X100 Y100 Z100 F200\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 X100 Y100 Z100 F100\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 P1000\n	Delay microsecond	$n ok \n $n Ex \n (refer to Err output)	
#n G2201 S100 R90 H80 F100\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)	
#n G2202 N0 V90 F100\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)	
#n G2204 X10 Y10 Z10 F100\n	Relative displacement	$n ok \n $n Ex \n (refer to Err output)	
#n G2205 S10 R10 H10 F100\n	Polar coordinates for relative displacement	$n ok \n $n Ex \n (refer to Err output)	
#n G2206 B90 L70 R50 F100\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**
System Command (parameters are in underline)			
#n S1000 V0	Control Arm motion 0:stop motion	$n ok \n $n Ex \n	**Support v4.5.0 or later**

	1:start motion	(refer to Err output)	
#n 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)			
#n M17\n	Attach all the joint motors	$n ok \n $n Ex \n (refer to Err output)	
#n M204 A1.3\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)	
#n M2019\n	Detach all the joint motors	$n ok \n $n Ex \n (refer to Err output)	
#n M2120 V0.2\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	
#n M2121\n	Stop feedback	$n ok \n $n Ex \n (refer to Err output)	
#n M2122 V1\n	Report (@9 V0) when stop. V1: Enable V0: Disable	$n ok \n $n Ex \n (refer to Err output)	
#n M2123 V1\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**
#n M2201 N0\n	Attach motor, N is ID of joints(0~3)	$n ok \n $n Ex \n (refer to Err output)	
#n M2202 N0\n	Detach motor, N is ID of joints(0~3)	$n ok \n $n Ex \n (refer to Err output)	
#n M2203 N0\n	Check if the motor is attached, N is ID of joints(0~3)	$n ok \n $n Ex \n (refer to Err output)	
#n M2210 F1000 T200\n	buzzer,F is frequency, T is time (ms)	$n ok \n $n Ex \n (refer to Err output)	

#n M2211 N0 A200 T1\n	Read EEPROM N(0~2,0 is internal EEPROM,1 is USR_E2PROM, 2 is SYS_E2PROM), A is address, T is type (1 char,2 int,4 float)	$n ok \n $n Ex \n (refer to Err output)	**This interface does not support temporarily**
#n M2212 N0 A200 T1 V10\n	Write EEPROM N(0~2,0 is internal EEPROM,1 is USR_E2PROM, 2 is SYS_E2PROM), A is address, T is type (1 char,2 int,4 float)V is the input data	$n ok \n $n Ex \n (refer to Err output)	**This interface does not support temporarily**
#n M2213 V0\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**
#n M2215\n	Reset Grbl parameter	$n ok \n $n Ex \n (refer to Err output)	**Add reset param interface**
#n M2220 X100 Y100 Z100\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)	
#n M2221 B0 L50 R50\n	Convert angle of joints to coordinates	$n ok X100 Y100 Z100\n $n Ex \n (refer to Err output)	
#n M2222 X100 Y100 Z100 P0\n	Check if it can reach,P1 polar, P0 Cartesian coordinates	$n ok V1\n (1 reachable, 0 unreachable) $n Ex \n (refer to Err output)	
#n M2231 V1\n	pump V1 working, V0 stop	$n ok \n $n Ex \n (refer to Err output)	
#n M2232 V1\n	gripper V1 close, V0 open	$n ok \n $n Ex \n (refer to Err output)	
#n M2233 V1\n	laser V1 working, V0 stop	$n ok \n $n Ex \n (refer to Err output)	**Add laser interface**
#n M2234 V1\n	Enable/disable Bluetooth (1:enable, 0:disable)	$n ok \n $n Ex \n (refer to Err output)	**This interface does not support temporarily**
#n M2240 N1 V1\n	Set the digital IO output	$n ok \n	**Support v4.3.0 or later**

		$n Ex \n (refer to Err output)	
#n M2241 N1 V1\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**
#n M2245 Vbtname\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**
#n M2400 S0\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)	
#n M2401\n	Set the current position into the reference position	$n ok \n $n Ex \n (refer to Err output)	
#n M2410\n	Set the height zero point	$n ok \n $n Ex \n (refer to Err output)	
#n M2411 S100\n	Set the offset of end-effector (mm)	$n ok \n $n Ex \n (refer to Err output)	
#n M2412 V10\n	Set the offset angle of end-effector(°)	$n ok \n $n Ex \n (refer to Err output)	**Support v4.2.0 or later**
Querying Command (parameters are in underline)			
#n P2200\n	Get the current angle of joints	$n ok B50 L50 R50\n $n Ex \n (refer to Err output)	
#n P2201\n	Get the device name	$n ok SwiftPro \n $n Ex \n (refer to Err output)	
#n P2202\n	Get the hardware version	$n ok V3.0.1\n $n Ex \n (refer to Err output)	
#n P2203\n	Get the software version	$n ok V4.0.0\n $n Ex \n (refer to Err output)	
#n P2204\n	Get the API version	$n ok V4.0.1\n $n Ex \n (refer to Err output)	
#n P2205\n	Get the UID	$n ok V0123456789AB\n $n Ex \n (refer to Err output)	

#n P2206 N0\n	Get the angle of number 0 joint (0~3)	$n ok V80\n $n Ex \n (refer to Err output)	Add get the angle of end-effector interface
#n P2220\n	Get current coordinates	$n ok X100 Y100 Z100\n $n Ex \n (refer to Err output)	
#n P2221\n	Get current polar coordinates	$n ok S100 R90 H80\n $n Ex \n (refer to Err output)	
#n P2231\n	Get the status of pump	$n ok V1\n (0 stop, 1 working, 2 grabbing things) $n Ex \n (refer to Err output)	
#n P2232\n	Get the status of gripper	$n ok V1\n (0 stop, 1 working, 2 grabbing things) $n Ex \n (refer to Err output)	
#n P2233\n	Get the status of limited switch	$n ok V1 \n (1 triggered, 0 untriggered) $n Ex \n (refer to Err output)	
#n P2234\n	Get the status of power connection	$n ok V1 \n (1 connected, 0 unconnected) $n Ex \n (refer to Err output)	
#n P2240 N1\n	Get the status of digital IO	$n ok V1\n (1 High, 0 Low) $n Ex \n (refer to Err output)	**Support v4.3.0 or later**
#n P2241 N1\n	Get the status of analog IO	$n ok V295\n (return the data of ADC) $n Ex \n (refer to Err output)	**Support v4.3.0 or later**
#n P2242\n	Get the default value of AS5600 in each joint	$n ok B2401 L344 R1048\n $n Ex \n (refer to Err output)	
#n P2400\n	Check current status	$n ok V1\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)	
事件报告			
@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		

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)