

Converting uProcess Scripts from Version 1.X to 2.X

Version 2 of the uProcess software includes many new scripting features, such as variable and math support. Inclusion of these features required changes to many of the script commands from uProcess V1. The table below gives an overview of changes from uProcess V1 to V2, see **Chapter 5** for the complete list of uProcess script commands.

Commands uProcess 1.0	Commands uProcess 2.0
<Beep>	Beep()
<Break>	Break
<Goto> StepName	Goto StepName
DeviceName: HysteresisReg Pmin Pmax units DeviceName: HysteresisReg Pmin Pmax units to ChX SyringeName: MoveWithChX (or MoveOppositeChX)	DeviceName: HysteresisReg (Pmin units, Pmax units) Ex: HysteresisReg(40 kPa, 60 kPa) DeviceName: HysteresisRegTo (Pmin units, Pmax units) SyringeName: MoveWith (DChX) (or MoveOpposite (DChX))
DeviceName: IfDone DeviceName: IfNotDone	DeviceName: IfDone() DeviceName: IfNotDone()
<log> on optional filename <log> off	log(on) or log(optional filename) log(off)
<Loop> StepName cycles	Loop StepName cycles
SyringeName: MoveTo vol units	SyringeName: MoveTo (vol units)
<Pause> duration units	Wait (duration units)
SyringeName: SetFlowRate rate units	SyringeName: SetFlowRate (rate units)
4VM01Name: SetValves x x x x	4VM01Name: SetValves (x, x, x ,x)
DeviceName: Stop	DeviceName: Stop()
<Quit>	Quit
SensorName: RegDownTo value units SensorName: RegUpTo value units	SensorName: RegDownTo (value units) SensorName: RegUpTo (value units)
SensorName: RegDownTo value units to ChX SyringeName: MoveWith ChX (or SyringeName: MoveOpposite ChX)	SensorName: RegDownTo (value units, DChX) (or RegUpTo (value units, DChX)) SyringeName: MoveWith (DChX) (or SyringeName: MoveOpposite (DChX))
SensorName: RegOff	SensorName: RegOff()
<WaitAllDone>	WaitDone()
DeviceName: WaitDone	DeviceName: WaitDone()

uProcess version 2.0 also supports several new commands including SetChannels, SetCurrent, SetOff, SetPerChannel, and SetPower. See the command descriptions in **Automated Script Functions** for more details.

Generic Formatting Updates

- Step names must be followed by a colon (":")
- Device names and step names cannot contain a dash ("-") (see **Device Naming and Numbering** for instructions on changing device names).

New Functionality: Variables, If Statements, and While loops

Variables can be defined and operated on in the script. Properties of connected devices can also be accessed in the script while it is running.

```
x1 = 20 ; declare variables
Voltage = sqrt(45) + sin(5*t) ; use mathematical expressions
position = Syringe.Volume ; access variables inherent to each device
If (uPS.Reading >=50) ; use an if statement
    SPS01: SetFlowRate (SPS01.MaxFlowRate ul/min)
While (a <=10) ; perform a while loop
{
    Beep()
    Wait(1 s)
    a = a + 1
}
```

More information on these functions and further uses and functions can be found in **Chapter 5**.

uProcess Version 1.x	uProcess Version 2.x
*Manifold = 4VM *Syringe = SPS 80 ul	*Manifold = 4VM *Syringe = SPS 80 ul
Open_Reservoir Manifold:SetValves 1 0 0 0 <WaitAllDone>	Open_Reservoir: Manifold:SetValves(1, 0, 0, 0) WaitDone()
Fill_Syringe Syringe:SetFlowRate 100.000 ul/min Syringe:MoveTo 80.000 ul <WaitAllDone>	Fill_Syringe: Syringe:SetFlowRate(100.000 ul/min) Syringe:MoveTo(80.000 ul) WaitDone()
Open_Process Manifold:SetValves 3 3 0 0 <WaitAllDone>	Open_Process: Manifold:SetValves(3, 3, 0, 0) WaitDone()
Infuse Syringe:SetFlowRate 10.000 ul/min Syringe:MoveTo 0.000 ul <WaitAllDone> <Loop> Fill_Syringe 5	Infuse: Syringe:SetFlowRate(10.000 ul/min) Syringe:MoveTo(0.000 ul) WaitDone() Loop Fill_Syringe 5