


	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

Spike

PH\_Setup

Functional Design Specification for Manufacturing Control  
System

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	


## APPROVALS

Functional Area	Required Approver	Date
Author / Originator	Print Name:	
	Sign Name:	
Engineering	Print Name:	
	Sign Name:	
Operations	Print Name:	
	Sign Name:	
Automation	Print Name:	
	Sign Name:	
QAV	Print Name:	
	Sign Name:	

	<p style="text-align: center;"><b>HAL Software Lifesciences</b></p>	Date: 25/10/2015	
		Doc no.:	

## Revision History

Revision	Date	Author	Description

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

## Table of Contents

APPROVALS ..... 2

Revision History..... 3

Table of Contents ..... 4

Reference Documentation..... 5

1. PH\_Setup ..... 7

    1.1. Running..... 8

        1.1.1. S0 Logic ..... 9

        1.1.2. S1 Logic ..... 9

        1.1.3. S2 Logic ..... 9

        1.1.4. S3 Logic ..... 10

        1.1.5. S4 Logic ..... 10

        1.1.6. S5 Logic ..... 11

        1.1.7. S6 Logic ..... 11

        1.1.8. S7 Logic ..... 11

        1.1.9. S8 Logic ..... 12


        1.1.10. T0 Logic ..... 12

        1.1.11. T1 Logic ..... 13

        1.1.12. T10 Logic ..... 13

        1.1.13. T11 Logic ..... 13


        1.1.14. T2 Logic ..... 13


	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

1.1.15.	T3 Logic .....	14
1.1.16.	T4 Logic .....	14
1.1.17.	T5 Logic .....	14
1.1.18.	T6 Logic .....	15
1.1.19.	T7 Logic .....	15
1.1.20.	T8 Logic .....	16
1.1.21.	T9 Logic .....	17
1.1.	Formula Name .....	17

## Reference Documentation

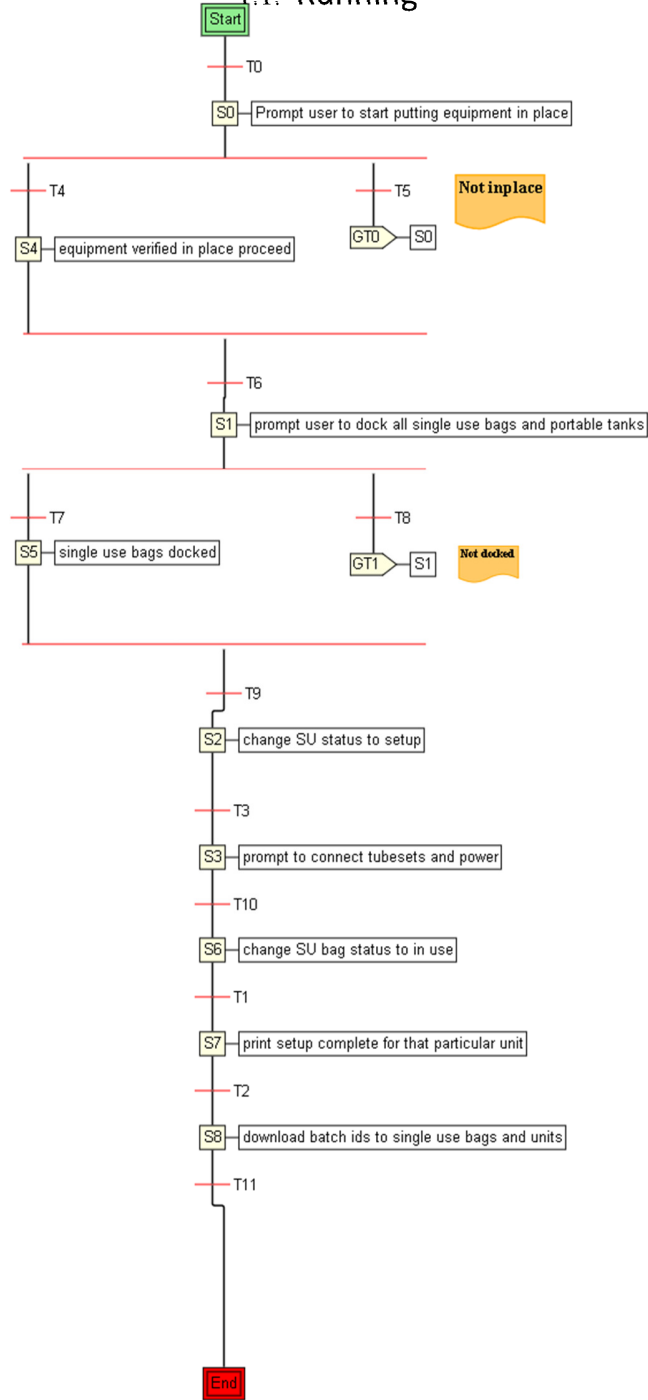
Document	Ref.	Title

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	


	<p>HAL Software Lifesciences</p>	Date: 25/10/2015	
		Doc no.:	

**1. PH\_Setup**

1.1. Running





	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

### 1.1.1.1. S0 Logic

```
#####
# Step: PH_Setup/Running/S0
#####

def main():
    """Main Function"""

    # Message the user...
    mycurrentequipment = unit.Name

    mytempstring = 'Please assemble:' + mycurrentequipment + ' equipment in designated area
and plug into mains and control system. Hint - change in place signal to true'

    unit.MessageModule.NewMessage('Equipment assembly', mytempstring)
```

### 1.1.1.2. S1 Logic

```
#####
# Step: PH_Setup/Running/S1
#####

def main():
    """Main Function"""

    # Message user to dock equipment...
    #
    mycurrentequipment = unit.Name

    mytempstring = 'Please dock:' + mycurrentequipment + 'Hint - Set single use state = docked
using SetDockingState on SU faceplate'


    unit.MessageModule.NewMessage('Equipment docking', mytempstring)
```

### 1.1.1.3. S2 Logic

```
#####
# Step: PH_Setup/Running/S2
#####

def main():
    """Main Function"""

    # Change unit single use bag status...
    mysingleuseequipmenttag1 = getStepParameter('singleusetag1')
```

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

```

mysingleuseequipmenttag2 = getStepParameter('singleusetag2')
mysingleuseequipmenttag3 = getStepParameter('singleusetag3')

#
opc(mysingleuseequipmenttag1.Value).SetSingleUseState(SingleUseStatus.BagSetup)

# allow for the use case where the bioreactor has more than one single use bag

if mysingleuseequipmenttag2.Value != 'dummy':
    opc(mysingleuseequipmenttag2.Value).SetSingleUseState(SingleUseStatus.BagSetup)

if mysingleuseequipmenttag3.Value != 'dummy':
    opc(mysingleuseequipmenttag3.Value).SetSingleUseState(SingleUseStatus.BagSetup)

```

#### 1.1.1.4. S3 Logic

```

#####
# Step: PH_Setup/Running/S3
#####

def main():
    """Main Function"""

    # Message the operator to change single use state to tubesetinstall...

    unit.MessageModule.NewMessage('Connect tubesets & power', 'connect the tubesets and
power. Hint - change single use status to tubesetinstall..')

```

#### 1.1.1.5. S4 Logic

```


#####
# Step: PH_Setup/Running/S4
#####

def main():
    """Main Function"""

    # notify the user...

    info('Equipment verified in place, proceed')

```

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

### 1.1.1.6. S5 Logic

```
#####
# Step: PH_Setup/Running/S5
#####

def main():
    """Main Function"""

    # Step code...
    info('Equipment verified as docked, proceed')
```

### 1.1.1.7. S6 Logic

```
#####
# Step: PH_Setup/Running/S6
#####

def main():
    """Main Function"""

    # get the hardware model from the formula definition...

    mysingleuseequipmenttag1 = getStepParameter('singleusetag1')
    mysingleuseequipmenttag2 = getStepParameter('singleusetag2')
    mysingleuseequipmenttag3 = getStepParameter('singleusetag3')

    #
    opc(mysingleuseequipmenttag1.Value).SetSingleUseState(SingleUseStatus.InUse)

    #allow for the use case where there is more than 1 single use bag in a unit.

    if mysingleuseequipmenttag2.Value != 'dummy':
        opc(mysingleuseequipmenttag2.Value).SetSingleUseState(SingleUseStatus.InUse)

    if mysingleuseequipmenttag3.Value != 'dummy':
        opc(mysingleuseequipmenttag3.Value).SetSingleUseState(SingleUseStatus.InUse)
```

### 1.1.1.8. S7 Logic

```
#####
# Step: PH_Setup/Running/S7
#####

def main():
    """Main Function"""
```



```
# setup is complete
info('Setup is now complete for: '+ unit.Name)
```

### 1.1.1.9. S8 Logic

```
#####
# Step: PH_Setup/Running/S8
#####

def main():
    """Main Function"""

    # get batchid

    mybatchid = getStepParameter('batchid')

    mysingleuseequipmenttag1 = getStepParameter('singleusetag1')
    mysingleuseequipmenttag2 = getStepParameter('singleusetag2')
    mysingleuseequipmenttag3 = getStepParameter('singleusetag3')

    # download the batch id

    opc(mysingleuseequipmenttag1.Value).BatchId = mybatchid.Value
    unit.BatchId = mybatchid.Value

    if mysingleuseequipmenttag2.Value != 'dummy':
        opc(mysingleuseequipmenttag2.Value).BatchId = mybatchid.Value


    if mysingleuseequipmenttag3.Value != 'dummy':
        opc(mysingleuseequipmenttag3.Value).BatchId = mybatchid.Value
```

### 1.1.1.10. T0 Logic

```
#####
# Transition: PH_Setup/Running/T0
#####

def main():
    """Main Function"""

    # Transaction code...
```

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

### 1.1.1.11. T1 Logic

```
#####
# Transition: PH_Setup/Running/T1
#####

def main():
    """Main Function"""

    # Transaction code...
```

### 1.1.1.12. T10 Logic

```
#####
# Transition: PH_Setup/Running/T10
#####

def main():
    """Main Function"""

    # Transaction code...
```

### 1.1.1.13. T11 Logic

```
#####
# Transition: PH_Setup/Running/T11
#####


def main():
    """Main Function"""

    # Transaction code...
```

### 1.1.1.14. T2 Logic

```
#####
# Transition: PH_Setup/Running/T2
#####

def main():
    """Main Function"""
```

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

```
# Transaction code...
```

### 1.1.1.15. T3 Logic

```
#####
# Transition: PH_Setup/Running/T3
#####

def main():
    """Main Function"""

    # Transaction code...
```

### 1.1.1.16. T4 Logic

```
#####
# Transition: PH_Setup/Running/T4
#####

def main():
    """Main Function"""

    # Transition code...
    transition(False)

    # get physical equipment names from formula definition

    myinplacesensor = getStepParameter('inusetag')

    # check if vessel is in place

    if opc(myinplacesensor.Value).Value == True:
        info(myinplacesensor.Value + " vessel is in place")
        transition(True)
```

### 1.1.1.17. T5 Logic

```
#####
# Transition: PH_Setup/Running/T5
```




```
#####  
  
def main():  
    """Main Function"""  
  
    # Transition code...  
    transition(False)  
  
    # get physical equipment names from formula definition  
  
    myinplacesensor = getStepParameter('inusetag')  
  
    #check that the vessel is NOT in place  
  
    if opc(myinplacesensor.Value).Value == True:  
        info(myinplacesensor.Value + " vessel is not in place")  
        info(' Put Equipment into position.')        transition(True)
```

### 1.1.1.18. T6 Logic

```
#####  
# Transition: PH_Setup/Running/T6  
#####  
  
def main():  
    """Main Function"""  
  
    # Transaction code...
```

### 1.1.1.19. T7 Logic

```
#####  
# Transition: PH_Setup/Running/T7  
#####  
  
def main():  
    """Main Function"""  
  
    # Transiton code...  
    transition(False)  
  
    # get physical equipment names from formula definition  
  
    mysingleuseequipmenttag1 = getStepParameter('singleusetag1')  
    mysingleuseequipmenttag2 = getStepParameter('singleusetag2')  
    mysingleuseequipmenttag3 = getStepParameter('singleusetag3')
```

	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

```

# Check for the use case where a bioreactor has more than one piece of equipment
# that must be docked. - Happens on the 2000L production reactor. Note that this is
poor unit class design

if (opc(mysingleuseequipmenttag1.Value) == DockingStatus.Docked) and
(mysingleuseequipmenttag2.Value == 'dummy'):
    info(mysingleuseequipmenttag1.Value + " single use bag is docked")
    transition(True)

if mysingleuseequipmenttag2.Value != 'dummy':
    warn('Warning! This unit has additional single use bags to be installed!:')
    if (opc(mysingleuseequipmenttag2.Value) == DockingStatus.Docked) and
(opc(mysingleuseequipmenttag3.Value) == DockingStatus.Docked):
        info(mysingleuseequipmenttag2.Value + " single use bag is docked")
        info(mysingleuseequipmenttag3.Value + " single use bag is docked")
        transition(True)

```

### 1.1.1.20. T8 Logic

```

#####
# Transition: PH_Setup/Running/T8
#####

def main():
    """Main Function"""

    # Transaction code...
    transition(False)


    # get physical equipment names from formula definition

    mysingleuseequipmenttag1 = getStepParameter('singleusetag1')

    # The following logic continuously prompts the user to dock the additional mixer tanks
on the large 2000L bioreactor

```



	<b>HAL Software Lifesciences</b>	Date: 25/10/2015	
		Doc no.:	

```

if (opc(mysingleuseequipmenttag1.Value).Value != DockingStatus.Docked):
    info(mysingleuseequipmenttag1.Value + " single use bag not docked")
    transition(True)
    info('Equipment not docked, retry')

```

### 1.1.1.21. T9 Logic

```

#####
# Transition: PH_Setup/Running/T9
#####

def main():
    """Main Function"""

    # Transaction code...

```

## 1.2. Formula Name

<b>Formula Name</b>