

TRANSITION

SOURCE: STRING
READ: CHARACTER
DESTINATION: STRING
WRITE: CHARACTER
MOVE: DIRECTION

TRANSITION (IN SOURCE_STATE: STRING,
IN READ_CHARACTER: CHARACTER,
IN DESTINATION_STATE: STRING,
IN WRITE_CHARACTER: CHARACTER,
IN MOVE_DIRECTION: DIRECTION)

SOURCE_STATE(): STRING
READ_CHARACTER(): CHARACTER
DESTINATION_STATE(): STRING
WRITE_CHARACTER(): CHARACTER
MOVE_DIRECTION(): DIRECTION

for each line in file:

```
line = line.split('#')[0] // removes comments
```

✓ Check if we've got a target declaration:

```
if line[0] == '[':
```

```
    if (TempTarget.complused):
```

```
        targetList.append(tempTarget)
```

```
        TempTarget = new Target
```

```
    else:
```

```
        TempTarget = new Target
```

```
    'otherwise, split on '='
```

```
else:
```

```
    strSplit = line.split('=')
```

```
    if strSplit[0] is in funcDict.keys():
```

```
        funcDict[strSplit[0]](strSplit[1])
```

```
funcDict = {  
    "X": storeX,  
    "Y": storeY,  
    "Z": storeZ,  
    "Friend": storeFriend,  
    "Points": setPoints,  
    "FlashRate": setFlash  
}
```

Each storage method has a prototype of the form:
setFunc(string input, Target inTarget) {}

Checks:

- Check for complete [target] headers
- Check all attributes for a target are set
-

```
storeX('3.222', tempTarget) X = 3.222 # Some Comment
```

```
'X=3.222'  
['X', '3.222']
```