```
//
//  IncomingMessage.swift
//  RGB Controller
//
//  Created by Erik Nordlund on 2/18/17.
//  Copyright © 2017 Erik Nordlund. All rights reserved.
//
//  Arm Controller includes the following open-source components:
//      • swiftBluetoothSerial: https://github.com/hoiberg/SwiftBluetoothSerial
//      • peertalk-simple: https://github.com/kirankunigiri/peertalk-simple

import Foundation

class IncomingMessage {
    func connectionVerifier() -> String {
        return "c."
    }

    func xyCommandVerifier() -> String {
        return "xy."
    }

    func zCommandVerifier() -> String {
        return "z."
    }

    func statusMessage(x: Double, y: Double, z: Int) -> String {
        let roundedX = Int(round(x*1000))
        let roundedY = Int(round(y*1000))


        let xString: String
        let yString: String
        let zString: String

        if (roundedX < 10) {
            xString = "0000\(roundedX)"
        } else if (roundedX < 100) {
            xString = "000\(roundedX)"
        } else if (roundedX < 1000) {
            debugPrint("roundedX < 1000: ", roundedX)
            xString = "00\(roundedX)"
        } else if (roundedX < 10000) {
            debugPrint("roundedX < 10000: ", roundedX)
            xString = "0\(roundedX)"
        } else {
            xString = String(roundedX)
        }

        if (roundedY < 10) {
            yString = "0000\(roundedY)"
```

```
        } else if (roundedY < 100) {
            yString = "000\(roundedY)"
        } else if (roundedY < 1000) {
            debugPrint("roundedY < 1000: ", roundedY)
            yString = "00\(roundedY)"
        } else if (roundedY < 10000) {
            debugPrint("roundedY < 10000: ", roundedY)
            yString = "0\(roundedY)"
        } else {
            yString = String(roundedY)
        }

        if (z == 0) {
            zString = "\(z)"
        } else if (z == 1) {
            zString = "\(z)"
        } else {
            debugPrint("z value not binary")
            zString = "0"
        }

        // xyz:x00.00y00.00z0!
        let message = "s:x\(xString)y\(yString)z\(zString)."
        return message
    }

    func zStatusMessage(z: Int) -> String {
        let message = "s:z\(z)."
        return message
    }

    // Configuration messages
    func configurationNameChangeVerifier() -> String {
        return "atname."//------------------------------ This isn't really
         an option right now.
    }

}
```