messungDataArray:

temparray teensy.read\_byteArray von DATA\_START\_BYTE..<BUFFER\_SIZE

let datastring:String = IntDataString(data:messungDataArray) // Array der Daten von readbytearray von DATA\_START\_BYTE an

messungstring = messungstring + "\n" + datastring

writeData(name: dataname,data:messungstring) // --> messungdump

messungfloatarray:

messungfloatarray[task][DIAGRAMMDATA\_OFFSET + 0,1,2,3] = analog0,1,2,3 float

adcfloatarray:

adcfloatarray [0,1,2,3] = analog0,1,2,3 float

**Aufbereitung**

var tempwerte = [Float] ( repeating: 0.00, count: 9 )

// eine Zeile mit messung-zeit und 8 floats

--> für

tempwerte[0] = Float(diff)

// Abszisse

var werteArray = [[Float]](repeating: [0.0,0.0,1.0,1.0], count: 9 )

Array mit werten fuer einen Datensatz im Diagramm

Data mit wert, deviceID, sortenfaktor anzeigefaktor

werteArray[0] = [Float(tempzeit),1.0,1.0]

// Abszisse

tempinputDataFeldstring = tempinputDataFeldstring + "\t" + String(format:"%.\(stellen)f", wert\_norm) // Eintrag in inputDataFeld

inputDataFeld.textStorage?.append(NSAttributedString(string:(tempinputDataFeldstring + "\n")))

**MESSUNG\_DATA:**

**device: 2 messungfloatzeilenarray: \* [0.0, 0.0, 0.0, 0.0, 1401.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]\***

func werteArrayFromIntArray:

**werteArrayFromIntArray** downloadfloatzeile: **[1401.0, 0.0, 2.0, 0.0]**

**index: 0 werteArray**

**[1.0, 1.0, 1.0]**

**[28.0200005, 2.0, 10.0, 2.0]**

**[0.0, 2.0, 10.0, 1.0]**

**[0.0400000028, 2.0, 10.0, 2.0]**

**[0.0, 2.0, 10.0, 1.0]**

**[0.0, 0.0, 1.0, 1.0]**

**[0.0, 0.0, 1.0, 1.0]**

**[0.0, 0.0, 1.0, 1.0]**

**[0.0, 0.0, 1.0, 1.0]**