

弁護士

令和3年12月16日

送信書

大阪地方裁判所 第26民事部合議係 御中 (FAX 06-6361-4779)

原告代理人弁護士 [REDACTED] (FAX 082-962-0289)

被告代理人

弁護士 [REDACTED]

TEL 082-227-2411

FAX 082-227-6699

大阪地方裁判所 令和元年(ワ)第10940号 損害賠償請求事件

原告 森次茂廣

被告 株式会社 [REDACTED]

次回期日 令和3年12月20日 午前10時30分

送信書類

令和3年12月14日付 準備書面13

被告代理人 [REDACTED] は、上記のとおり、本書を含め合計5枚をFAXにより提出及び直送します。

本FAXを受領後は直ちに、落丁や送信ミスの有無を確認し、下記受領書部分に記載のうえ、送信者である [REDACTED] と大阪地方裁判所第26民事部の2ヶ所に本書面をそのままFAXして下さい。

(切り取らない)

受領書

大阪地方裁判所 第26民事部合議係 御中 (FAX 06-6361-4779)

被告代理人 弁護士 [REDACTED] 殿 (FAX 082-227-6699)

令和 年 月 日、上記文書を受領しました。

氏名

印

令和元年（ワ）第10940号 損害賠償請求事件

原告 森次 茂廣

被告 株式会社 [REDACTED]

## 準備書面 1 3

令和3年12月14日

大阪地方裁判所第26民事部合議係 御中

被告訴訟代理人弁護士 [REDACTED]



### 1 変換値 (InpBuf) について

#### (1) 裁判所の指摘

裁判所は、被告が令和3年11月9日付準備書面12にて示した表におけるオフセットバイナリの変換値 (InpBuf) が、通常のオフセットバイナリの場合と違うと指摘する。

#### (2) 変換値 (InpBuf) は、2進数を10進数に変換した値

しかし、上記表の変換値 (InpBuf) は、ADI16-4(FIT)GY が16bitの2進数で表現した数値を、基数変換の法則に基づき、ただ機械的に10進数の表現へ変換したものに過ぎない（以下、10進数の値を、整数値という。上記表の変換値は、整数値である。）。

#### ア 基数変換

2進数を10進数に変換する等、数字を表現する際の基数を別の基数に変える表現することを基数変換という。

ここで、2進数を10進数に変換する数式について説明する。

16Bit の2進数を10進数に変換する数式（以下「2進数を10進数に変換する数式」という。）は、次のとおり、2進数の各桁の重みを掛けたものの合計である。

2進数の15桁の数値 $\times 2^{15}$  + 2進数の14桁の数値 $\times 2^{14}$  + 2進数の13桁の数値 $\times 2^{13}$  + 2進数の12桁の数値 $\times 2^{12}$  + 2進数の11桁の数値 $\times 2^{11}$  + 2進数の10桁の数値 $\times 2^{10}$  + 2進数の9桁の数値 $\times 2^9$  + 2進数の8桁の数値 $\times 2^8$  + 2進数の7桁の数値 $\times 2^7$  + 2進数の6桁の数値 $\times 2^6$  + 2進数の5桁の数値 $\times 2^5$  + 2進数の4桁の数値 $\times 2^4$  + 2進数の3桁の数値 $\times 2^3$  + 2進数の2桁の数値 $\times 2^2$  + 2進数の1桁の数値 $\times 2^1$  + 2進数の0桁の数値 $\times 2^0$  = 10進数の整数値

ただ、2進数において最上位の bit（桁）が「1」の場合、この2進数は負の数表現しているところ、負の2進数を10進数に変換する場合は、2の補数表現を用いなければならない。具体的には、2進数の数値のビットを反転し、1を足す。ただ、この1を足した結果、最上位（16bitの場合は16桁目）の桁上がりが生じたときは無視をする。

以上の基数変換は、自然法則であり、顕著な事実である。

#### イ 本件について

ADI16-4(FIT)GY の変換データは、オフセットバイナリの 16bit 表現であり、入力電圧と 16bit 変換データとの対応関係は、次のとおりとなる。

オフセットバイナリ	
入力電圧	16bit 表現
10	1111111111111111
5	1100000000000000
0	1000000000000000
-5	0100000000000000
-10	0000000000000000

このうち、16bit の 2 進数「1111111111111111」を例に挙げる。この 16bit の 2 進数「1111111111111111」は、最上位の桁が「1」なので、負の数を表しており、これを整数値に変換するためには 2 の補数表現を用いなければならない。この 16bit の 2 進数「1111111111111111」を 2 の補数表現すると、2 進数「1111111111111111」のビットを反転させた 2 進数「0000000000000000」に 1 を足し、結果、2 進数「0000000000000001」となる。この 2 進数「0000000000000001」を、2 進数を 10 進数に変換する数式に当てはめると、「1」になる。2 進数「1111111111111111」は、負の数なので、結果、2 進数「1111111111111111」の整数値（10 進法に変換した値）は「-1」となる。

また、16bit の 2 進数「1100000000000000」を例に挙げる。この 16bit の 2 進数「1100000000000000」は、最上位の桁が「1」なので、負の数を表しており、これを整数値に変換するためには 2 の補数表現を用いなければならない。この 2 進数「1100000000000000」を 2 の補数表現すると、2 進数「1100000000000000」のビットを反転させ 2 進数「0011111111111111」となり、これに 1 を足すと、2 進数では「0100000000000000」となる。この「0100000000000000」を、2 進数を 10 進数に変換する数式に当てはめると、16384 ( $= 1 \times 2^{14}$ ) となる。2 進数「1100000000000000」は、負の数なので、結果、2 進数「1100000000000000」の整数値（10 進法に変換した値）は「-16384」となる。

以上は負の数についてだが、正の数については次のとおりとなる。16bit の 2 進数「0100000000000000」を例に挙げる。この 16bit の 2 進数「0100000000000000」は、最上位の桁が「0」なので、正の数を表している。したがって 2 の補数表現を用いることなく、2 進数を 10 進数に変換する数式により、整数値に変換することができる。この「0100000000000000」

を、2進数を10進数に変換する数式に当てはめると、16384 ( $=1 \times 2^{14}$ ) となる。

ウ 小括

以上のとおり、上記表の変換値 (InpBuf) は、ADI16-4(FIT)GY が16bitの2進数で表現した数値を、基数変換の法則に基づき、機械的に10進数の表現へ変換したものに過ぎない。

2 乙23の変換値が16bit表現であること

サイレントロボのプログラムは、Microsoft Visual Basic6.0で制作されている。そして、Microsoft Visual Basic6.0での整数型は16bitで表される整数値である。

なお、Microsoft Visual Basic.NET以降は、整数型が32bitで表される整数値に変更されている。裁判所の指摘は、異なるバージョンのプログラム言語を前提とするものである。

以 上