

ロビプログラミング講座

ロビの動きが決まる ランダム処理

ロビのプログラムでよく出てくるものにランダム処理があります。

例えばロビにクイズ出してという毎回違った問題を出してくれるのはこのランダム処理を行っているからでランダム処理で計算した値を元に出題のプログラムを決めています。

ランダム値の取得

コンピューターでランダム値を求めるには複雑な計算をする必要がありますがロビの場合マイコンボードでこの値を計算してくれています。

完全な乱数を求めるのは複雑な計算が必要ですが擬似乱数と呼ばれるものを使っています。

```
<mem_r size="4" adr=0x0084"/>
```

```
<mem_w size="2" adr="0x0f00"/>
```

0x0084 というのは乱数を取得するためのマイコンボードのアドレス(番地)で頭に 0x が付いているのは 16 進数という事を表しています。

mem_r というのは取得の命令である事を示しています。サイズは取得する整数の大きさ(桁数)を表しバイト単位で指定します。この場合は 4 バイト=32 ビットという事になります。

乱数はこの範囲内のいずれかの値(整数)で - 2,147,483,648 ~ 2,147,483,647 のいずれかの値になります。

ランダム値の計算

2 の 32 乗の値が取得されるのでこのままではプログラムでは使えません。そのため必要とする範囲の値を求めます。ここで注意しなければいけないのはマイナスの値も乱数として取得されてしまう事です。

```
<mem_r size="2" adr="0x0f00"/>
```

```
<const data="32767"/>
```

```
<and/>
```

and というのは元の値と 32767(下 15 ビットが全て 1)で両方のビットが 1 の場合は 1 とする演算子で必要な部分だけ取り出す時によく使われます。1100011000110001 and 0111111111111111 → 0100011000110001

ロビのプログラムでは上記の計算で乱数を 0 ~ 32767 の範囲に丸めています。

次に必要な範囲にします。例えばジャンケンのように 3 つのケースが必要な場合は取得した乱数を 3 で割った余りを計算します。

```
<const data="3"/>
```

```
<mod/>
```

この計算式は 3 で割った余りの値を求めるもので 0 ~ 2 の値が計算結果になります。

ランダム値による条件分岐

計算したランダム値でプログラムの飛び先を変更する事でプログラムの動作を変えています。

```
<const data="1"/>
```

```
<sub/>
```

```
<eq/>
```

```
<jump adr="0x000005c7"/>
```

最初の 2 行で求めた乱数から 1 を引く計算を行っています。

次の eq というのは分岐するための条件式でロビのマイコンでは計算結果を常に 0 と比較します。

この場合 0 と等しい場合のみ次の行の命令が実行されます。

adr は飛び先のアドレスでプログラムの番地となります。

同様に 0 と 2 についても条件分岐の処理を行う訳ですが 0 と 1 の条件分岐を計算した場合、両方の条件分岐しない場合は必ず 2 という事になるので条件分岐ではなく無条件分岐(最後の jump 命令のみ)で行います。