

Pseudo Random Binary Sequence (C++)

Séquence Binaire Pseudo Aléatoire (programme C++)

C++ programs for the generation of PRBS are provided. These programs have been written by Adaptech [<http://www.adaptech.com>] and they can be freely used provided that the author (Adaptech) is mentioned.

On donne dans ce qui suit une présentation des programmes pour la génération des séquences binaires pseudo aléatoires utilisées pour l'identification. Ces programmes élaborés par Adaptech [<http://www.adaptech.com>] sont dans le domaine public et peuvent être utilisés librement (sous réserve de la mention de l'auteur).

A.7.1. Présentation de la génération des SBPA sous forme de programme

SBPA appliquée autour d'un point de fonctionnement.

Description des variables :

N = nombre de registres,

REG(.) = registre à décalage,

K1, K2 = bits à additionner du registre,

NE = nombre d'échantillons à acquérir,

P = entier diviseur de fréquence,

UI = point de fonctionnement du processus,

U(.) = vecteur contenant les entrées (excitations) à envoyer aux processus

AS = amplitude de la SBPA

Programme :

```

initialisation des bits à additionner
K1 = N - 1
K2 = N
IF N=5 THEN K1 = 3
IF N=7 THEN K1 = 4
IF N=9 THEN K1 = 5
IF N=10 THEN K1 = 7
IF N=11 THEN K1 = 9
initialisation du registre à décalage
FOR I=1 TO N
    REG(I) = 1
NEXT I
NE = INT(NE/P) * P      NE = multiple de P
initialisation du vecteur U par groupes de P éléments
FOR K=1 TO NE STEP P
    UIU = 1 or -1 (bit du registre à décalage)
    UIU = -REG(K1) * REG(K2)
    IF N=8 THEN UIU = -REG(2) * REG(3) * REG(4) * REG(8)
    initialisation de P éléments du vecteur U
    FOR J=K TO (K+P-1)
        U(J) = UIU * AS + UI
        (+/-1 * amplitude SBPA) + point de fonctionnement
    NEXT J
    décalage du registre
    FOR I=N TO 2 STEP -1
        REG(I) = REG(I-1)
    NEXT I
    REG(1) = UIU
NEXT K

```

A.7.2. Programme en C++ pour la génération d'une SBPA

```

//-----
// Fonction : Creer_sbpa( )
// Génère la Séquence Binaire Pseudo-Aléatoire dans un tableau.

```

```

// Auteur : Adaptech
//Création du fichier : 14/02/2000
//-----
void Creer_sbpa(float ValUnit, float ValAmpli, float ValDecal, int ValLgReg,
               int ValDivi, int nombre_valeur, float *signal_sbpa )
{
/* Les paramètres d'entrée sont :
    ValUnit : Valeur initiale du signal
    ValAmpli : Amplitude du signal
    ValDecal : Offset
    ValLgReg : valeur du registre N (entre 2 et 11)
    ValDivi : valeur du diviseur de fréquence p (entre 1 et 3)
    nombre_valeur : nombre de valeur de sbpa =  $p \cdot 2^{(N-1)}$ 
    Après création du signal, le paramètre de sortie est le vecteur signal_sbpa qui contient
    les valeurs de la sbpa à appliquer à chaque instant d'échantillonnage. */
int  k1, k2, i, j, uiu, sbpa[11];

k1 = ValLgReg-1;
k2 = ValLgReg;
if (ValLgReg == 5) {k1 = 3;}
if (ValLgReg == 7) {k1 = 4;}
if (ValLgReg == 9) {k1 = 5;}
if (ValLgReg == 10) {k1 = 7;}
if (ValLgReg == 11) {k1 = 9;}

for (i=1; i<=11; i++)
    sbpa[i] = 1;
i = 1;
while(i <= nombre_valeur){
    uiu = -sbpa[k1]*sbpa[k2];
    if (ValLgReg == 8)
        uiu = -sbpa[2]*sbpa[3]*sbpa[4]*sbpa[8];
    j = 1;
    while(j<=ValDivi){
        signal_sbpa[i-1] = uiu * ValAmpli + ValDecal + ValUnit;
        i++;
        j++;
    }
    for (j=ValLgReg; j >= 2; j--)
        sbpa[j] = sbpa[j-1];
}
}

```

536 Titre de l'ouvrage

```
        sbpa[1] = uiu;  
    }  
}
```

A.7.3. Notes et indications bibliographiques

Plus de détails sont disponibles dans :

Adaptech (2001b) : *Guide d'intégration SBPA (séquence binaire pseudo aléatoire)*, Adaptech, 4 Rue de la Tour de l'Eau, Saint Martin d'Hères, France.