

Question 6 : Analyser une réponse

Ce programme reçoit une trame provenant du bus CAN, puis analyse son contenu et enfin affiche les données contenues dans la trame.

Vous écrirez la fonction permettant d'analyser le contenu de la trame dont le prototype est le suivant
`void MPPT_analyser_reponse(MPPTData* donnees, CANFrame reponse);`

Fichiers CAN.c et CAN.h

Ils restent identiques

Fichier Makefile

```
reponse: reponse.c CAN.c CAN.h MPPT.c MPPT.h
gcc -Wall -Werror -Wextra -o reponse reponse.c CAN.c MPPT.c
```

Programme principal: reponse.c

```
#include <stdio.h>
#include "CAN.h"
#include "MPPT.h"

int main( )
{
    CANFrame reponse;
    MPPTData donnees;

    // Réception des données CAN
    printf("=== CAN_receive() ===\n");
    CAN_receive(&reponse);

    // Analyser la réponse du MPPT
    MPPT_analyser_reponse(&donnees, reponse);
    MPPT_print(donnees);                // Afficher les données

    return 0;
}
```

Fichier header : MPPT.h

```
#ifndef __MPPT_H
#define __MPPT_H

#include <stdbool.h>

/* Define MPPT Data structure*/
typedef struct MPPT_data {
    unsigned int mppt_id;
    bool bvlr;
    bool ovt;
    bool noc;
    bool undv;
    float u_in;
    float i_in;
    float u_out;
    char t_ambiante;
} MPPTData;

void MPPT_print(MPPTData data);
void MPPT_analyser_reponse(MPPTData* donnees, CANFrame reponse);

#endif
```

Compléter le code source ci-dessous : MPPT.c

```
#include <stdio.h>
#include "MPPT.h"

#define PRINT_BOOL(x) (x)? "true": "false"

// Affichage des données MPPT
void MPPT_print(MPPTData data)
{
    printf( "=== mppt_data ===\n" );
    printf( "mppt_id = %d\n", data.mppt_id);
    printf( "bvlr = %s\n", PRINT_BOOL(data.bvlr));
    printf( "ovt = %s\n", PRINT_BOOL(data.ovt));
    printf( "noc = %s\n", PRINT_BOOL(data.noc));
    printf( "undv = %s\n", PRINT_BOOL(data.undv));
    printf( "u_in = %5.2f\n", data.u_in );
    printf( "i_in = %5.2f\n", data.i_in );
    printf( "u_out = %5.2f\n", data.u_out );
    printf( "t_ambiante = %d\n", data.t_ambiante );
}
```