



## Test des entrées RAPE sur MPLABX

Nous nous proposons dans ce cours de charger le programme du test des entrées puis de le tester sur le robot RAPE enfin de contrôler que toutes les entrées TOR et ANA sont fonctionnelles.

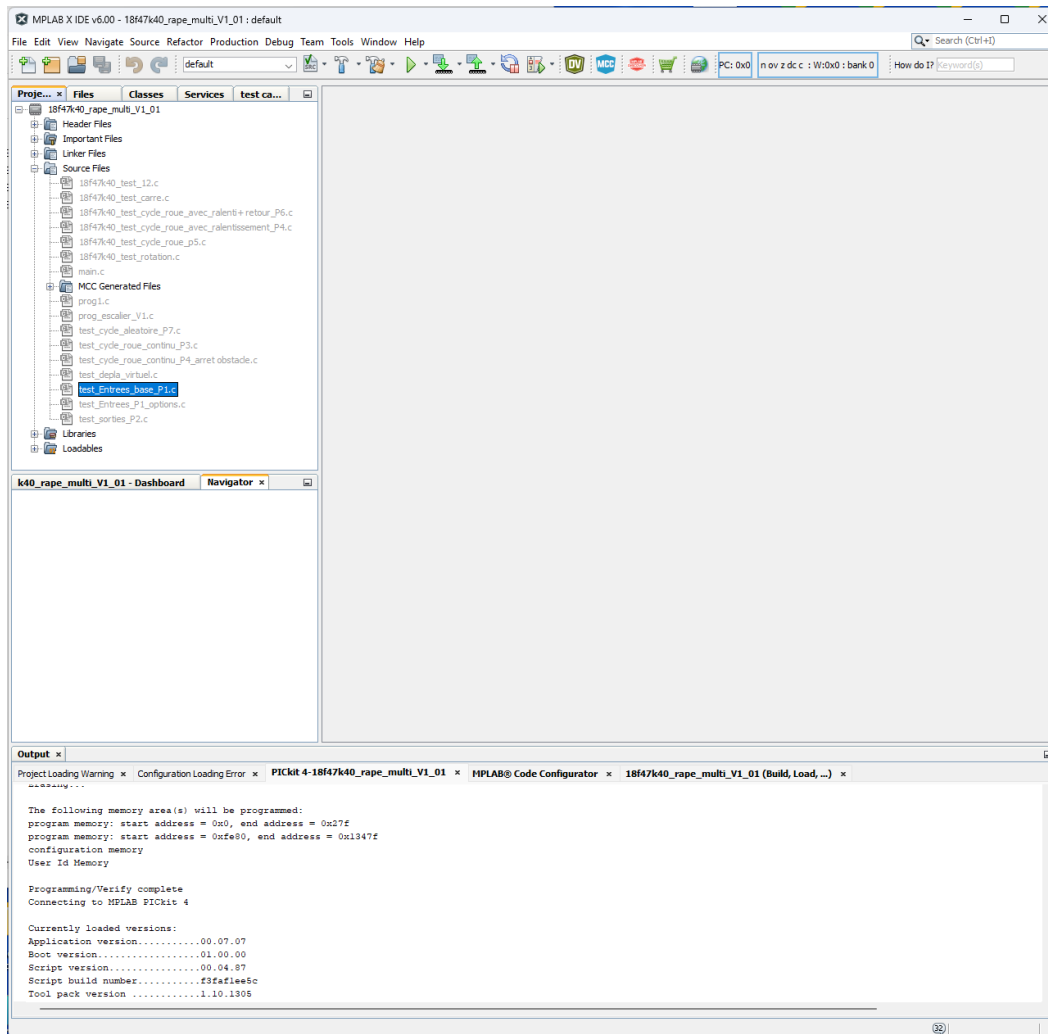
### Table des matières

Test des entrées RAPE sur MPLABX .....	1
Charger le programme des entrées .....	2
Description du programme : .....	4
Explication sur le programme test des entrées : .....	4
• Lecture des entrées analogiques.....	4
• appel des fonctions sur chaque tour de scrutation .....	4
• appel de la fonction print toutes les 500ms.....	5

**La vidéo associée : installation fichier mplab test des entrées liaison rs232**

11/2023

## Charger le programme des entrées



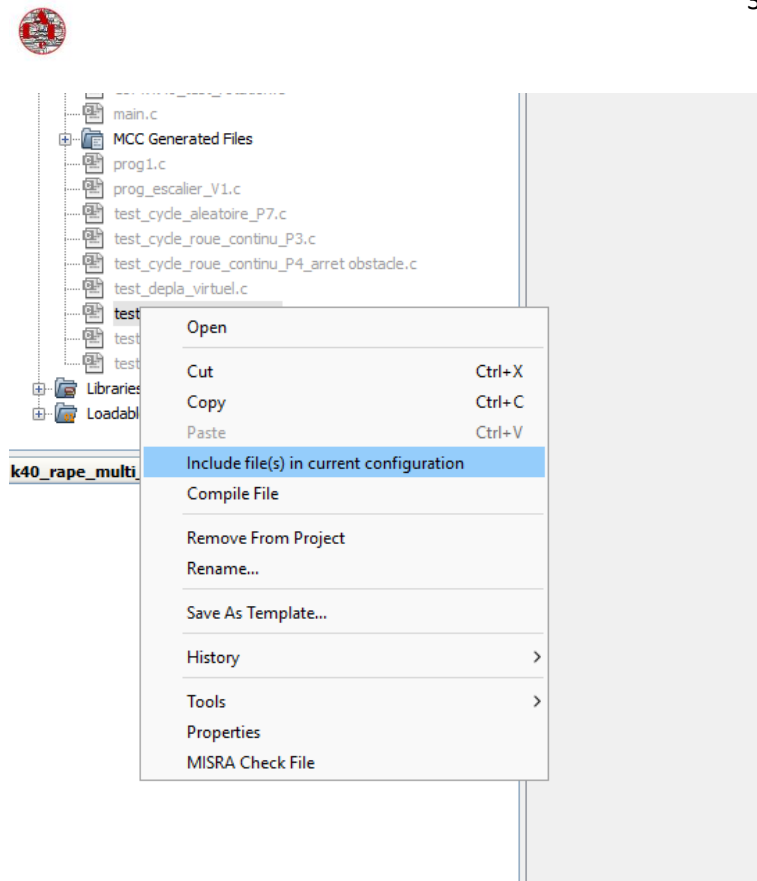
The screenshot displays the MPLAB X IDE interface for a project named '1847k40\_rape\_multi\_V1\_01'. The left-hand pane shows a hierarchical file structure with the following categories:

- Header Files
- Important Files
- Linker Files
- Source Files
  - 1847k40\_test\_12.c
  - 1847k40\_test\_carre.c
  - 1847k40\_test\_cycle\_roue\_avec\_ralenti+retour\_P6.c
  - 1847k40\_test\_cycle\_roue\_avec\_ralentissement\_P4.c
  - 1847k40\_test\_cycle\_roue\_p5.c
  - 1847k40\_test\_rotabon.c
  - main.c
- MCC Generated Files
  - prog.c
  - prog\_escaller\_V1.c
  - test\_cycle\_aleatoire\_P7.c
  - test\_cycle\_roue\_continu\_P3.c
  - test\_cycle\_roue\_continu\_P4\_arret\_obstade.c
  - test\_depla\_virtuel.c
  - test\_entrees\_base\_P1.c** (highlighted)
  - test\_entrees\_P1\_options.c
  - test\_sorbes\_P2.c
- Libraries
- Loadables

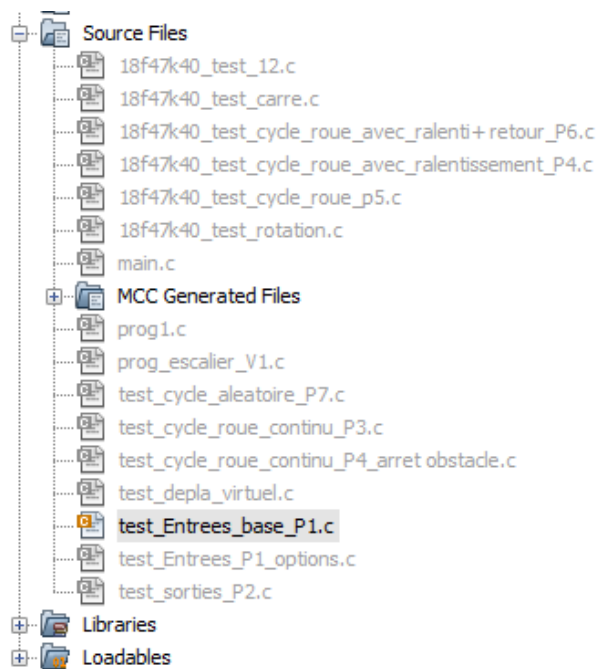
The bottom pane shows the 'Output' window with the following text:

```
ProjectLoading Warning x ConfigurationLoading Error x PICkit 4-1847k40_rape_multi_V1_01 x MPLAB® Code Configurator x 1847k40_rape_multi_V1_01 (Build, Load, ...) x  
.....  
The following memory area(s) will be programmed:  
program memory: start address = 0x0, end address = 0x27f  
program memory: start address = 0xf80, end address = 0x1347f  
configuration memory  
User Id Memory  
Programming/Verify complete  
Connecting to MPLAB PICkit 4  
Currently loaded versions:  
Application version.....00.07.07  
Boot version.....01.00.00  
Script version.....00.04.87  
Script build number.....f32af1ee5c  
Tool pack version .....1.10.1305
```

« test\_entrees\_base\_P1 »



Le programme passe en gras





11/2023

Cliquez deux fois dessus :

```

1  /**
2  * RAPE
3  *
4  * sur carte MAIN_V01-01_poc3-2023
5  * robot RAPE à programmer avec MPLAB et pic18f47k40
6  *
7  * Programme N°1
8  * premier essaï
9  * Programme de test des Entrées sur le schéma de base uniquement
10 * la led verte doit clignoter une fois le programme injecté
11 * l'affichage sur pc fonctionne et permet de voir les compteurs roue et les tensions ana
12 *
13 *
14 * La led verte doit clignoter en permanence sous les 100ms dans 0.1s
15 * l'affichage doit être présent sur l'écran du pc avec la liaison RS232 stable
16 *
17 * *****entrées TOR :
18 * les trois BP un appui led bleue
19 * le bouclier avant
20 * lorsque l'on fait tourner les roues à la main les compteurs s'incrémentent à voir sur l'affichage.
21 *
22 *
23 * *****entrées analogiques :
24 * cellule centrale tension variable de 300 à 650.
25 * tension batterie (on retire le fil du bornier) on met en série une résistance de 10ohm on lit 500 sur l'entrée ana on ramet le fil on a
26 *
27 * la consommation à vide est de 60mA
28 *
29 */
30
31 #include "mcc_generated_files/mcc.h"
32 #include <stdio.h>
33 #include <xc.h>
34 #include <stdint.h>
35 #include "mcc_generated_files/examples/i2c1_master_example.h" // sans ça pas de fonctionnement du I2C.....;
36 #include <math.h>
37
38
39

```

### Description du programme :

Dans ce petit programme de « test des entrées » nous allons tester les entrées TOR (boutons, fin de course sur bouclier avant), les entrées analogiques (cellule avant et tension batterie).

Eventuellement allez voir sur le cours « **1-4 structure d'un programme RAPE** »

### Explication sur le programme test des entrées :

- Lecture des entrées analogiques

Dans un premier temps on va lire les entrées analogiques et on stocke les valeurs dans deux variables : « capteur\_central\_avant\_AN0 » et « tension\_batterie\_AN3 » que l'on pourra utiliser plus loin dans le programme le texte capteur\_central.....ce sera plus simple que ADCC.....

```
capteur_central_avant_AN0 = ADCC_GetSingleConversion(channel_ANA0);
```

```
tension_batterie_AN3 = ADCC_GetSingleConversion(channel_ANA3);
```

- appel des fonctions sur chaque tour de scrutation

```
FM_tempo_100ms();
```

```
FM_tempo_500ms();
```

```
commande_sorties()
```

```
cycle (); //cycle grafcet
```



- appel de la fonction print toutes les 500ms

```
if (fm_bit_500ms) {  
    printf("\r\n");  
    printf("  cmpt_roue_GA: %d ", cmpt_roue_GA); //  
    printf("  cmpt_roue_DR: %d ", cmpt_roue_DR);  
    printf("ana0 centrale,%d  ", capteur_central_avant_AN0);  
    printf("ana3 tension,%d  ", tension_batterie_AN3);  
}
```

*Dans ce cas on affiche sur l'écran du pc avec le logiciel « TERA TERM » les valeurs des variables*

*Exemple : la variables est cmpt\_route\_GA et on affiche le texte qui est entre guillemet avec deux point et la valeur de la variable en décimal.*

En sous-programmes appelés par la fonction main :

```
void cycle ()  
{  
    //test des leds  
    // if (bp1) {sortie_led_bleue =1;} else {sortie_led_bleue=0;} autre écriture possible  
    sortie_led_bleue = bp1 || bp2 || !det_AV_ID; //  
    if (fm_bit_100ms) {sortie_led_coeur_ve = !sortie_led_coeur_ve;}  
}
```

*Dans cette fonction « cycle » on affiche la led bleue si on appui sur le bouton 1 ou le bouton 2 ou si il y absence de l'entrée du détecteur en effet l'entrée à une résistance de rappel à 5v donc on aura toujours 1 sauf que le détecteur sera activé on aura zéro c'est une sécurité positive si le fil casse sur le détecteur le robot considérera qu'il y toujours un obstacle il nous faudra par la suite gérer ce cas.*

```
void commande_sorties(void)  
{  
    sortie_mot_brosse = 0;  
    sortie_mot_IN1_recul_DR = 0;  
    sortie_mot_IN2_avance_DR= 0;
```



```
sortie_mot_IN3_recul_GA = 0;

sortie_mot_IN4_avance_GA= 0;

}
```

*Les sorties ne doivent être commandées qu'une seule fois cela évite d'avoir des sorties qui scintillent. Dans ce cas toutes les sorties à zéro pour qu'elles ne soient pas en l'air et soient commandées de manière intempestive.*

```
void FM_tempo_100ms(void)
```

```
{

    if (bit_100ms & !inter_100ms ) {fm_bit_100ms = 1;} //
    else {fm_bit_100ms = 0;}
    /* bit intermédiaire */
    inter_100ms = bit_100ms;
    return;

}
```

```
void FM_tempo_500ms(void)
```

```
{

    if (bit_500ms == 1 & inter_500ms == 0) {fm_bit_500ms = 1;} //
    else {fm_bit_500ms = 0;}
    /* bit intermédiaire */
    inter_500ms = bit_500ms;
    return;

}
```

*Ce sont des fronts montants qui sont appelés, en début de programme.*



### Test pratique sur le robot.

- 1- Faire le transfert du programme dans le pic.
- 2- Appuyez sur BP1 la LED bleue s'allume
- 3- Appuyez sur BP2 la LED bleue s'allume
- 4- Appuyez sur le pare choc avant la led bleue doit s'allumer
- 5- Brancher la liaison rs232, lancer TERA TERM les informations venant du pic doivent apparaitre
- 6- Placer sa main devant le robot environ 650 au début puis diminution
- 7- Retirer le fil brun du + batterie puis placer une résistance de 100 ohms la valeur doit passer de 1023 à 500 pts.

### Remarque sur les dysfonctionnements:

- si le voyant vert ne clignote pas rapidement, une cause possible est le non chargement du programme.
- Si le voyant bleu est allumé en permanence le détecteur sur le pare choc avant est mal branché ou bloqué mécaniquement en position enfoncée ou le/les bp en normalement fermés.