



## Test des entrées analogiques du ROBOT RARE

Dans ce cours nous présentons les entrées analogiques le schéma, la configuration programme MCC, le calcul avec le nombre de points.

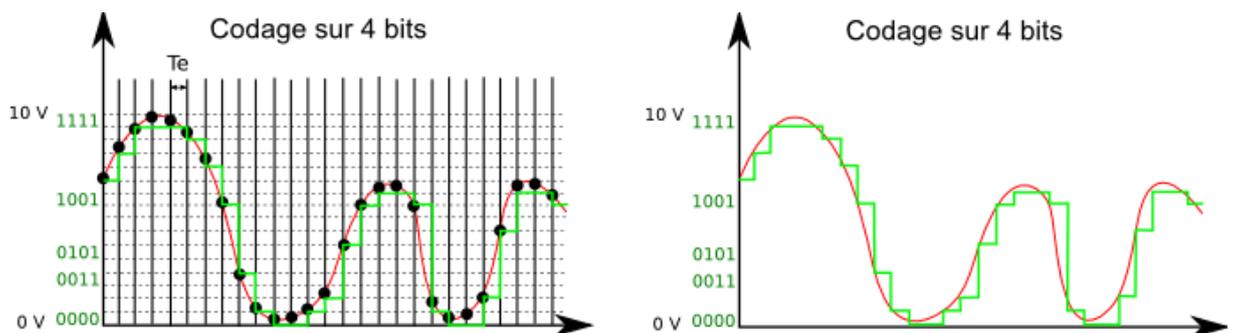
### Table des matières

|   |   |
|---|---|
| Test des entrées analogiques du ROBOT RARE.....               | 1 |
| Présentation des entrées analogiques .....                    | 1 |
| Structure du programme RAPE avec les entrées analogiques..... | 2 |
| les fichiers créés par MCC.....                               | 4 |
| Vérification sur notre robot sur la cellule centrale .....    | 5 |

### Présentation des entrées analogiques

CAN = Convertisseur Analogique Numérique (en anglais : ADC = Analog to Digital Convertor)

Un convertisseur analogique numérique est un système qui fait une correspondance entre une tension analogique et une valeur numérique.



Dans notre programme les entrées analogiques sont sur 10 bits c'est-à-dire de 0 à 1023, donc 1024bits.

5v correspondra a 1023 donc 0.0048v par point



Structure du programme RAPE avec les entrées analogiques

The screenshot displays the IDE interface for a project named '18f47k40\_rape\_multi\_V1\_01'. The project structure is shown in the left pane, with 'Source Files' expanded to show various test files. The file 'test\_Entrées\_base\_P1.c' is highlighted. The right pane shows the code editor for 'test\_Entrées', with line numbers 43 to 68 visible. The code in the editor consists of a series of comments and preprocessor directives, including '#d' and '//'. The bottom pane shows the 'main() - Navigator' window.

| Line | Code |
|------|------|
| 43   | #d   |
| 44   | #d   |
| 45   | #d   |
| 46   | #d   |
| 47   | #d   |
| 48   |      |
| 49   | //   |
| 50   | //   |
| 51   | //   |
| 52   | #d   |
| 53   | #d   |
| 54   |      |
| 55   | #d   |
| 56   | #d   |
| 57   | #d   |
| 58   | #d   |
| 59   |      |
| 60   |      |
| 61   | //   |
| 62   | #d   |
| 63   | #d   |
| 64   | #d   |
| 65   | #d   |
| 66   |      |
| 67   | #d   |
| 68   | #d   |

Toujours le programme des entrées.





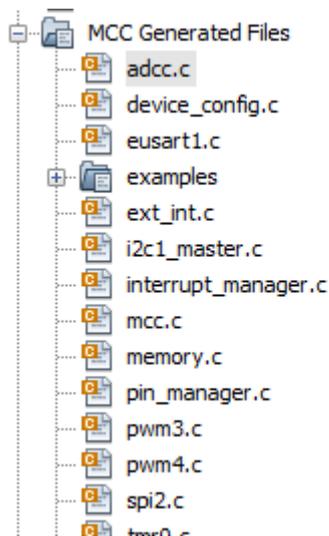
11/2023

```
//////////////////////////////////cde avance un pas//////////////////////////////////  
  
cycle (); //cycle grafcet  
  
if (fm_bit_500ms ) {  
  
    printf("\r\n");  
  
    printf("    compteur roue gauche: %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);  
}
```

```
printf("ana0 centrale,%d " , capteur_central_avant_AN0);
```

```
printf("ana3 tension,%d " , tension_batterie_AN3);
```

### les fichiers créés par MCC





```
adc_result_t ADCC_GetSingleConversion(adcc_channel_t channel)
{
    // select the A/D channel
    ADPCH = channel;

    // Turn on the ADC module
    ADCON0bits.ADON = 1;

    //Disable the continuous mode.
    ADCON0bits.ADCONT = 0;

    // Start the conversion
    ADCON0bits.ADGO = 1;

    // Wait for the conversion to finish
    while (ADCON0bits.ADGO)
    {
    }
}
```

Dans le adcc.h on retrouve les informations concernant cette opération

```
adc_result_t ADCC_GetSingleConversion(adcc_channel_t channel);

/**
 * @Summary
 *   Stops the ongoing continuous A/D conversion.
 *
 * @Description
 *   This routine is used to stop ongoing continuous A/D conversion.
 *
 * @Preconditions
 *   ADCC_Initialize() and ADCC_StartConversion() functions should have been called before calling this function.
 *
 * @Returns
 *   None
 *
 * @Param
 *   None
 *
 * @Example
 *   <code>
 *   ADCC_Initialize();
 *   ADCC_StartConversion(channel_ANA0);
 *   ADCC_StopConversion();
 *   </code>
 */
```

Vérification sur notre robot sur la cellule centrale

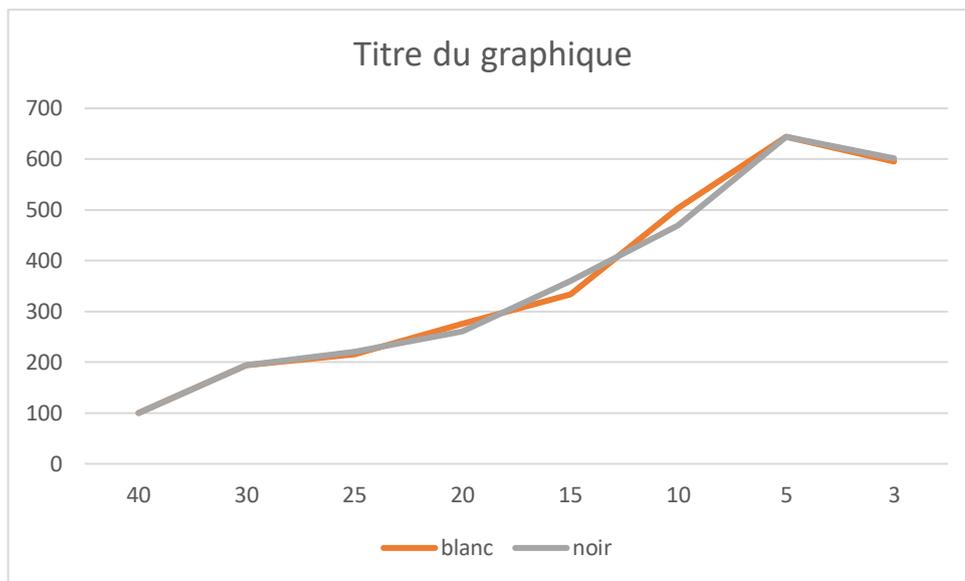
En utilisant une feuille de papier on peut établir la valeur en fonction de la distance :

| cm       | blanc lice | fond noir granuleux |
|----------|------------|---------------------|
| abscisse | blanc      | noir                |
| 40       | 100        | 100                 |



11/2023

|    |     |     |
|----|-----|-----|
| 30 | 194 | 194 |
| 25 | 215 | 220 |
| 20 | 276 | 261 |
| 15 | 334 | 360 |
| 10 | 503 | 470 |
| 5  | 644 | 644 |
| 3  | 595 | 601 |



On constate que la cellule est presque linéaire (ce qui est une bonne chose pour simplifier le programme) et quelque soit le type de matériaux en réflexion on a pas de changement sur le signal.

On pourra faire la même chose avec la batterie.