Date: 05/12/2014

# Protocole mis en œuvre :

* [WIP] : Étude des systèmes pour l'alignement avec le pont :
* [ ] : [Laser + Capteur de luminosité](file:///P:\SI\Projet\Etude\Laser%20+%20Capteur%20de%20luminosité.odt)
* [ ] : [LED + Capteur luminosité](file:///P:\SI\Projet\Etude\LED%20+%20Capteur%20de%20luminosité.odt)
* [ ] : [Ligne au sol + Capteur de couleur](file:///P:\SI\Projet\Etude\Ligne%20au%20sol%20+%20Capteur%20de%20couleur.odt)

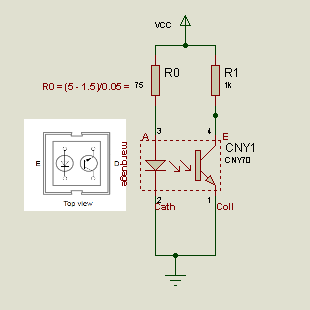
# Production finale constatée :

# Moyens utilisés :

# Avis et observations sur les résultats de l'étude :

Date: 12/12/2014

# Protocole mis en œuvre

Etude du système LED + Capteur luminosité

Mesure de l’intensité du composant CNY1 en fonction de la distance de celui-ci et de l’obstacle (cf [Tableau](#_Production_finale_constatée))

# Production finale constatée

Le transistor du CNY1 en bloqué.

# Moyens utilisés.

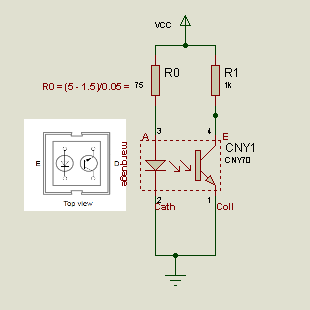
Schéma ISIS

# Avis et observations sur les résultats de l'étude.

Le phototransistor reste bloquant quelque soit la position de l’obstacle. Cependant la diode semble marcher.

Date: 09/01/2015

# Protocole mis en œuvre

Etude du système LED + Capteur luminosité

Mesure de l’intensité du composant CNY1 en fonction de la distance de celui-ci et de l’obstacle

# Production finale constatée

Tension aux bornes du CNY1 en fonction de la distance.

# Moyens utilisés.

Schéma ISIS

# Avis et observations sur les résultats de l'étude.

Le phototransistor reste bloquant quelque soit la position de l’obstacle. Cependant la diode semble marcher.

Date: 16/01/2015

# Protocole mis en œuvre :

* Début du code de gestion de l’état de charge

# Production finale constatée :

* 5 Macros :
* void initLed() // Initialise les leds
* octet getBat() // Renvoi le niveau de batterie : 0 = 0% ; 255 = 100%
* void showBatLevel() // Allume les leds en fonction du niveau de batterie
* octet isBatLevelOK(octet prct) // Renvoi true/false si la batterie est suffisante (>= prct)
* octet getTrou(octet timeout) // Renvoi true dès la présence d’un trou. Après timeout ms, renvoi false
* Le système LED + Capteur de luminosité est plutôt efficace lors de courtes distances.

|  |  |
| --- | --- |
|  | On choisit donc une valeur limite de 4.7V  Donc pour le pont de résistance on a :   * un potentiomètre de 220kΩ * une résistance de 880kΩ |

# Moyens utilisés :

* FlowCode
* Schéma ISIS
* Montage Electronique

# Avis et observations sur les résultats de l'étude :

Date: 23/01/2015

# Protocole mis en œuvre :

* Debut de la mise en place d’une routine composant pour le déplacement qui prend en compte une demande de vitesse, de sens et de rotation

# Production finale constatée :

* void move(octet vitesse, octet sens, octet degre) // Active les moteurs en fonction des trois demande

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* Le début de la routine marche, les autres tests n’ont pas pu etre effectué car FlowCode est en version gratuite

Date: 30/01/2015

# Protocole mis en œuvre :

* Redistribution des Ports du µControleur
* Etude du système moteur

# Production finale constatée :

* [Fichier papier ECIO40](Entrées-Sorties%20%20ECIO40.ods)

# Moyens utilisés :

# Avis et observations sur les résultats de l'étude :

Date: 06/02/2015

# Protocole mis en œuvre :

* Production du diaporama pour la soutenance

# Production finale constatée :

# Moyens utilisés :

# Avis et observations sur les résultats de l'étude :

Date: 06/03/2015

# Protocole mis en œuvre :

* Test, Reprogramation, Validation des routines composants, sur le µControleur

# Production finale constatée :

* Les routines sont maintenant toutes opértationelle

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* Les paramètres de la routine move sont maintenant des ENTIER cepandant les nombres doivent rester dans l’intervalle 0 - 255

Date: 12/03/2015 & 13/03/2015

# Protocole mis en œuvre :

* Re-répartition des ports pour l’integration d’un moteur pas à pas

# Production finale constatée :

* La routine : RIEN Align() //Align le robot avec le trou

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* La routine Align doit encore etre testé (Le batit sera nécéssaire pour des test optimaux)

Date: 20/03/2015

# Protocole mis en œuvre :

* Test des moteurs Pas-a-Pas et Servomoteurs
* Re-câblage de la carte des entrées sur port B, qui permet d’avoir une masse et une entrée 5V commune

# Production finale constatée :

* Une nouvelle carte Input Port B

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* La masse commune est obligatoire pour certaine lecture analogique

Date: 03/04/2015

# Protocole mis en œuvre :

* Améliotration du cycle de pose du pont

# Production finale constatée :

* La programme Flowcode est presque terminer

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* RaS

Date: 09/04/2015

# Protocole mis en œuvre :

* Création d’une nouvelle macro : RIEN verrin(OCTET monte) ; //Monte ou dessent les verins selon la valeur de *monte*
* Création d’une nouvelle macro : OCTET getTrouState(OCTET nb) ; //Renvoi la valeur binaire d’une ou des deux capteur de trou en ADC. Si *nb* vaut 1 ou 2 : renvoit la capteur 1 ou 2. Si il vaut 0 : Renvoit 0bXY ou X et Y représente l’état logique binaire des deux capteurs

# Production finale constatée :

* La routine verrin est opérationelle
* La routine getTrouState n’est pas encore finit.

# Moyens utilisés :

* FlowCode

# Avis et observations sur les résultats de l'étude :

* Le routine verrins évite une duplication de code x4