

Dossier De Conception (DDC)

du projet

Kart À Hélice (KAH)

Responsabilité documentaire

| Action | NOM Prénom | Fonction | Date | Signature |
|----------------------------------|---|----------------|------------|-----------|
| Rédigé par | Lestage Margot Cassany Marie Roboam Damien Jerez Elouan Frieih Mathieu L'Haridon Mathis Larjuzan Noah Bakas Iaila (KartoFuel) | Technicien | 20/02/2024 | |
| Approuvé par | F.AUGEREAU W.D'ANNA (IUT GEII Bdx) | Chef de projet | JJ/MM/AAAA | |
| Approuvé par | S. AROUL | Client | JJ/MM/AAAA | |
| IUT Bordeaux Département GEii | Référence : KAH_DDC_EQ21 Révision : 3 – 20/02/2024 | | | 1/69 |

(Toy Corporation)

Suivi des révisions documentaires

| Indice | Date | Nature de la révision |
|--------|------------|--|
| 1 | 01/09/2021 | Publication préliminaire du DDC, document à compléter par les Techniciens |
| 2 | 17/02/2024 | Première publication : Conception préliminaire |
| 3 | 20/02/2024 | Correction des remarques |
| | | |
| | | |

Documents de références

| Sigle | Référence | Titre | Rév. | Origine |
|-------|-----------|--------------------|------|------------------|
| [CDC] | KAH_CDC | Cahier des charges | 1 | Baby Corporation |
| | | | | |
| | | | | |

Table des matières

| | |
|---|-----------|
| 1. Nature du document | 6 |
| 2. Conception préliminaire du produit | 6 |
| 2.1 Mécanique | 6 |
| 2.1.1 Mécanique - Émetteur | 6 |
| Référence du paragraphe : CPR_EMTT_ARCHI_MECA | 6 |
| Référence du paragraphe : CPR_EMTT_DIMENSIONS | 7 |
| Référence du paragraphe : CPR_EMTT_LOGO | 7 |
| 2.1.2 Mécanique - Récepteur | 8 |
| Référence du paragraphe : CPR_RCPT_ARCHI_MECA | 8 |
| Référence du paragraphe : CPR_RCPT_DIMENSIONS | 9 |
| Référence du paragraphe : CPR_RCPT_LOGO | 9 |
| 2.2 Électronique | 9 |
| 2.2.1 Électronique - Émetteur | 9 |
| Référence du paragraphe : CPR_EMTT_ARCHI_ELEC | 9 |
| Référence du paragraphe : CPR_EMTT_IHM | 10 |
| Référence du paragraphe : CPR_EMTT_KLAXON | 11 |
| Référence du paragraphe : CPR_EMTT_TRAITEMENT | 11 |
| Référence du paragraphe : CPR_EMTT_REPETITIVITE | 12 |
| Référence du paragraphe : CPR_EMTT_RETENTISSEMENT | 12 |
| Référence du paragraphe : CPR_EMTT_PUISSANCE | 13 |
| Référence du paragraphe : CPR_EMTT_INDICATEUR | 13 |
| Référence du paragraphe : CPR_EMTT_ENERGIE | 14 |
| Référence du paragraphe : CPR_EMTT_INTERRUPTEUR | 15 |
| Référence du paragraphe : CPR_EMTT_SCHEMA | 16 |
| 2.2.2 Électronique - Récepteur | 16 |
| Référence du paragraphe : CPR_RCPT_ARCHI_ELEC | 16 |
| Référence du paragraphe : CPR_RCPT_CAPTEUR | 17 |
| Référence du paragraphe : CPR_RCPT_TRAITEMENT | 18 |
| Référence du paragraphe : CPR_RCPT_SECURITE | 19 |
| Référence du paragraphe : CPR_RCPT_RETENTISSEMENT | 19 |
| Référence du paragraphe : CPR_RCPT_MOTEUR | 19 |
| Référence du paragraphe : CPR_RCPT_ROUE | 20 |
| Référence du paragraphe : CPR_RCPT_INDICATEUR | 20 |
| Référence du paragraphe : CPR_RCPT_CONNEXION | 21 |
| Référence du paragraphe : CPR_RCPT_KLAXON | 21 |
| Référence du paragraphe : CPR_RCPT_ENERGIE | 22 |

| | |
|---|----|
| Référence du paragraphe : CPR_RCPT_INTERRUPTEUR | 23 |
| Référence du paragraphe : CPR_RCPT_SCHEMA | 23 |
| 2.3 Informatique | 24 |
| 2.3.1 Informatique - Émetteur | 24 |
| Référence du paragraphe : CPR_EMTT_ARCHI_INFO | 24 |
| 2.3.2 Informatique - Récepteur | 26 |
| Référence du paragraphe : CPR_RCPT_ARCHI_INFO | 26 |
| 2.4 Coût - Délai | 28 |
| Référence du paragraphe : CPR_COUT | 28 |
| Référence du paragraphe : CPR_DELAI | 29 |
| 2.5 Conclusion de la conception préliminaire du produit | 30 |
| 3. Conception détaillée du produit | 32 |
| 3.1 Électronique | 32 |
| 3.1.1 Électronique - Émetteur | 32 |
| Référence du paragraphe : CDT EMTT IHM | 32 |
| Référence du paragraphe : CDT_EMTT_KLAXON | 33 |
| Référence du paragraphe : CDT_EMTT_PUISSANCE | 34 |
| Référence du paragraphe : CDT_EMTT_INDICATEUR | 35 |
| Référence du paragraphe : CDT_EMTT_TRAITEMENT | 35 |
| Référence du paragraphe : CDT_EMTT_ENERGIE | 39 |
| Référence du paragraphe : CDT_EMTT_INTERRUPTEUR | 42 |
| 3.1.2 Électronique - Récepteur | 43 |
| Référence du paragraphe : CPR_RCPT_ARCHI_ELEC | 43 |
| Référence du paragraphe : CDT_RCPT_ENERGIE | 43 |
| Référence du paragraphe : CDT_EXIG_RCPT_CAPTEUR | 44 |
| Référence du paragraphe : CDT_RCPT_TRAITEMENT | 46 |
| Référence du paragraphe : CDT_RCPT_INDICATEUR | 49 |
| Référence du paragraphe : CDT_RCPT_CONNEXION | 51 |
| Référence du paragraphe : CDT_RCPT_KLAXON | 52 |
| 3.2 Informatique | 54 |
| 3.2.1 Informatique - Émetteur | 54 |
| Référence du paragraphe : CDT_EMTT_TRAITEMENT | 54 |
| Référence du paragraphe : CDT_EMTT_REPETITIVITE | 56 |
| Référence du paragraphe : CDT_EMTT_RETENTISSEMENT | 57 |
| 3.2.2 Informatique - Récepteur | 58 |
| Référence du paragraphe : CDT_RCPT_TRAITEMENT | 58 |
| Référence du paragraphe : CDT_RCPT_SECURITE | 63 |
| Référence du paragraphe : CDT_RCPT_RETENTISSEMENT | 63 |

| | |
|---|-----------|
| 3.3 Coût - Délai | 64 |
| Référence du paragraphe : CPR_COUT | 64 |
| Référence du paragraphe : CPR_DELAI | 64 |
| 3.4 Conclusion de la conception détaillée du produit | 65 |
| 4. Conclusion de la conception du produit | 65 |
| 5. Matrice de conformité du produit | 65 |

1. Nature du document

Ce document est un dossier de conception et a pour but de détailler la conception du produit développé. Il apporte ainsi des preuves de la conformité du produit par rapport à l'ensemble des exigences client. Le paragraphe 3 du [CDC] décrit de façon plus détaillée la nature et le positionnement de ce document dans l'arborescence documentaire du projet.

2. Conception préliminaire du produit

Compétences GEII : C1-4

Ce chapitre décrit l'architecture fonctionnelle du produit. Il apporte les premiers éléments de preuve de la faisabilité du produit vis-à-vis des exigences client.

2.1 Mécanique

2.1.1 Mécanique - Émetteur

Référence du paragraphe : CPR_EMTT_ARCHI_MECA

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu et Mathis L'Haridon

Compétences GEII : C1-9

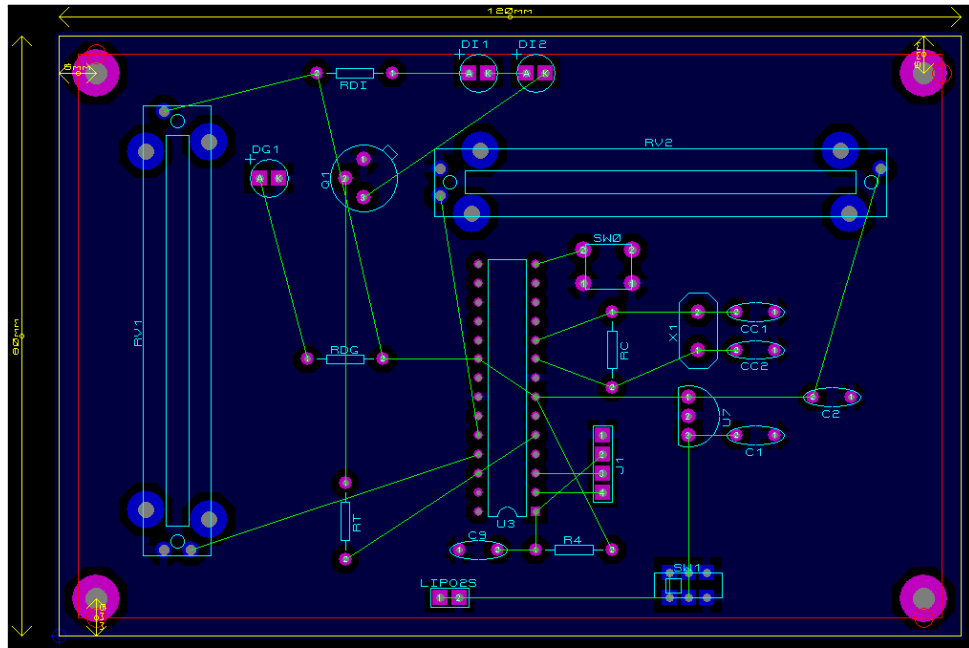


Figure 1 : Plan mécanique préliminaire de l'émetteur

Ci-dessus vous pouvez voir le schéma du positionnement préliminaire des composants de la carte de l'émetteur à échelle 1 sous le logiciel ARES (Suite Proteus). Nous respectons donc bien les exigences de mécanique en pouvant positionner tous les composants (MCU, résistances, condensateurs, transistors, connecteurs) sur cette carte avec une certaine marge.

Nous avons placé les potentiomètres sur les bords de la carte afin de faciliter l'accès aux commandes lors de l'utilisation et de se rapprocher au plus d'une manette de jeux vidéo. Le bouton d'action du buzzer est sous le potentiomètre de direction pour avoir toujours la maîtrise de la vitesse.. Les diodes infrarouges sur le devant de la carte transmettre au mieux le signal. Les autres composants sont placés de façon à simplifier le routage.

Référence du paragraphe : CPR_EMTT_DIMENSIONS

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieuh Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_EMTT_DIMENSIONS

Compétences GEII : C1-10

L'émetteur a des dimensions égales à 120mm en largeur, 80mm en longueur. Cela répond à l'exigence puisque le schéma au-dessus (Figure 1) respecte les dimensions demandées par le cahier des charges.

Référence du paragraphe : CPR_EMTT_LOGO

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_EMTT_LOGO

Compétences GEII : C1-10

L'émetteur comporte sur une de ses faces visibles le logo du département GEII de Bordeaux et le logo de l'équipe de conception. La carte comporte 96 cm² d'espace libre pour placer ces logos. Cela répond à l'exigence fournie par le cahier des charges.

2.1.2 Mécanique - Récepteur

Référence du paragraphe : CPR_RCPT_ARCHI_MECA

Rédacteur : Cassany Marie, Lestage Margot et L'Haridon Mathis

Relecteur : Frieih Mathieu, Jerez Elouan et Roboam Damien

Compétences GEII : C1-9

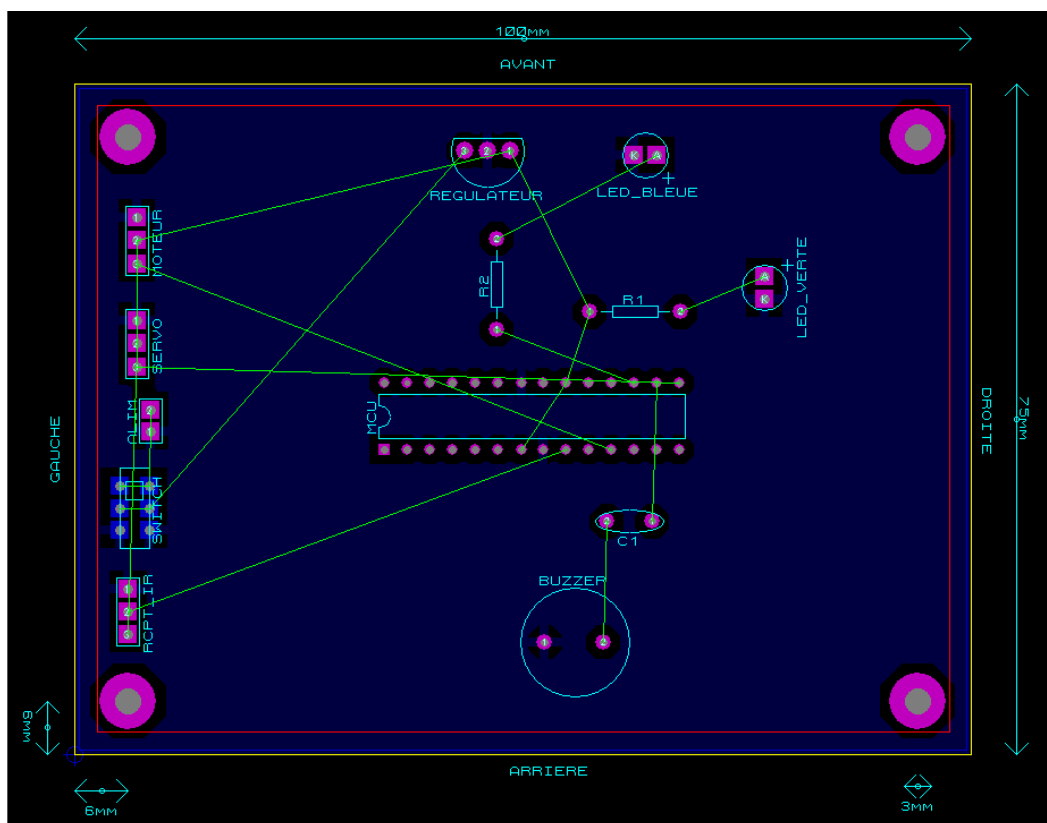


Figure 2 : Plan mécanique préliminaire du récepteur

Voici, en Figure 2, le plan mécanique préliminaire du récepteur. Sur cette carte, nous aurons cinq connecteurs (pour un moteur brushless, un servo moteur, une alimentation et un récepteur

InfraRouge (IR)) situés à gauche de la carte parce que, sur le kart, les fils arrivent par la gauche de la carte.

De plus, les composants suivants sont situés à des positions temporaires : un régulateur linéaire avec ces deux condensateurs de découplage, un interrupteur Switch, deux leds (bleue et verte), un buzzer et un MCU ATmega328-P. Les connecteurs se situent sur le côté gauche des entrées.

Référence du paragraphe : CPR_RCPT_DIMENSIONS

Rédacteur : Jerez Elouan, Roboam Damien et Lestage Margot

Relecteur : Frieh Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_RCPT_DIMENSIONS

Compétences GEII : C1-10

Le circuit imprimé du récepteur a des dimensions de 100 mm en longueur et 75 mm en largeur et comporte des trous de fixation de 3 mm situés dans chaque coin pour fixer celui-ci au reste de la mécanique du kart. Le centre de ces trous sont placés à 6 mm des bords du circuit imprimé. Cela répond à l'exigence puisque le schéma au-dessus (Figure 2) respecte les dimensions demandées par le cahier des charges.

Référence du paragraphe : CPR_RCPT_LOGO

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieh Mathieu, L'Haridon Mathis, Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_LOGO

Compétences GEII : C1-10

Sur le devant du Kart, il y a plus de 141.6 cm² d'espace pour coller l'étiquette avec le logo de l'IUT ainsi que le nom et le numéro de l'équipe. Cela réponds bien à l'exigence du cahier des charges.

2.2 Électronique

2.2.1 Électronique - Émetteur

Référence du paragraphe : CPR_EMTT_ARCHI_ELEC

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieh Mathieu et L'Haridon Mathis

Compétences GEII : C1-3, C1-9, C1-11

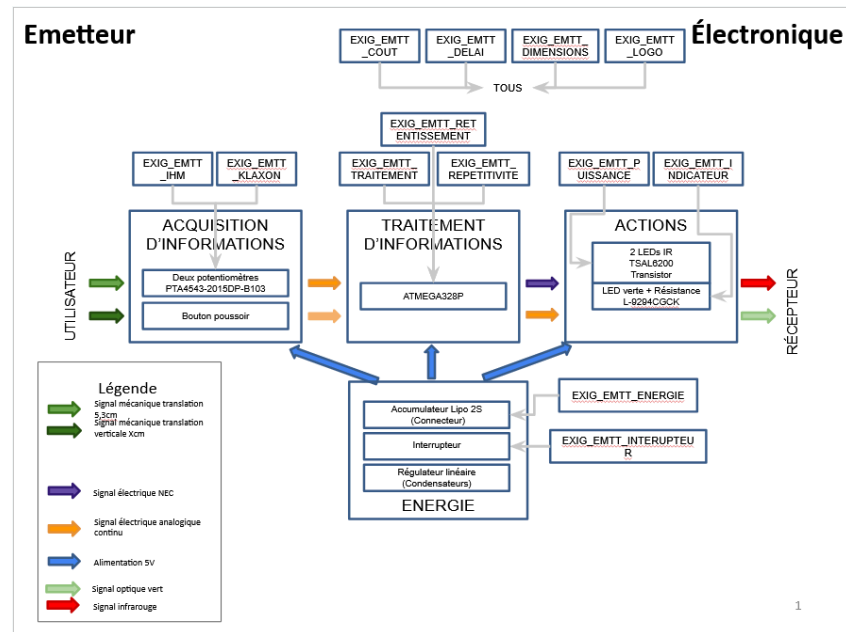


Figure 3 : Synoptique de l'architecture électronique de l'émetteur

Voici, en Figure 3, le synoptique de l'architecture électronique où nous y retrouvons les exigences avec les parties fonctionnelles avec la solution technique associée. Nous avons aussi en légende le type de signal transmis entre les différentes parties fonctionnelles.

Référence du paragraphe : CPR_EMTT_IHM

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_EMTT_IHM

Compétences GEII : C1-10, C1-11

Type d'étage retenu : Deux potentiomètres rectilignes : PTA4543-2015DP-B103

L'émetteur comporte 2 interfaces homme-machine qui permettent de contrôler respectivement la puissance du moteur et la direction des roues avant du kart. Ces 2 interfaces sont des potentiomètres de type rectiligne. Cela répond à l'exigence fournie par le cahier des charges.

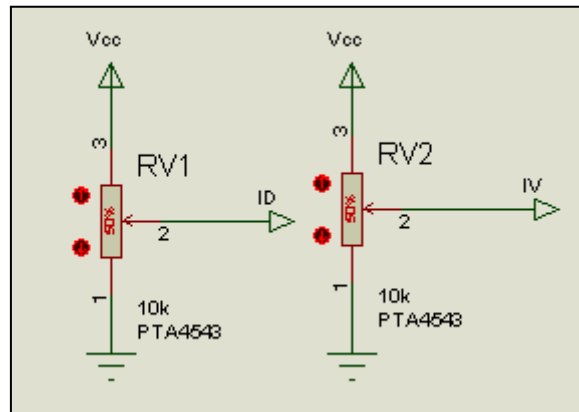


Figure 4 : Schéma électrique préliminaire des potentiomètres de l'émetteur

Référence du paragraphe : CPR_EM TT_KLAXON

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frie h Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_EM TT_KLAXON

Compétences GEII : C1-10, C1-11

Type d'étage retenu : Un bouton poussoir

L'émetteur comporte un bouton-poussoir sur lequel l'utilisateur peut appuyer pour indiquer qu'il souhaite faire retentir le klaxon du kart. Cela répond à l'exigence fournie par le cahier des charges.

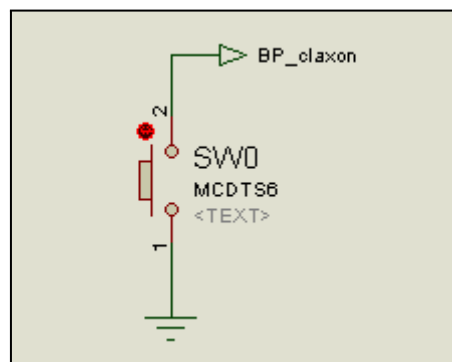


Figure 5 : Schéma électrique préliminaire du klaxon de l'émetteur

Référence du paragraphe : CPR_EM TT_TRAITEMENT

Rédacteur : Frie h Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG_EM TT_TRAITEMENT

Compétences GEII : C1-10, C1-11

Nous allons utiliser un coeur de traitement ATmega328-P parce qu'il possède plusieurs pins analogiques avec plusieurs ADC (Analog to Digital Converter), plusieurs pins PWM (Pulse Width Modulation) et une fréquence supérieur à 38 kHz pour recevoir le signal NEC. Ainsi, ce microcontrôleur répond parfaitement à l'exigence attendue.



Figure 6 : Schéma préliminaire du MCU de l'étage traitement

Référence du paragraphe : CPR EMTT REPETITIVITE

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG EMTT REPETITIVITE

Compétences GEII : C1-10, C1-11

Pour répondre à cette exigence, nous allons créer une fonction nommée “Memetrame” dans le code du programme informatique. Elle vérifiera que la nouvelle trame est bien différente de l’ancienne. Si oui, le délai inter trame sera de 108 ms. Sinon, le délai inter trame sera de 333 ms. Pour gérer le délai, nous utiliserons le CPU (Central Processing Unit) et un des trois Timer (Périphérique) du MCU ATMEGA328-P. Ainsi, cette fonction répondra parfaitement à l’exigence attendue.

Référence du paragraphe : CPR EMTT RETENTISSEMENT

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG EMTT RETENTISSEMENT

Compétences GEII : C1-10, C1-11

Nous allons utiliser le bit de poids fort de l'adresse du protocole NEC afin de signaler si le bouton du klaxon est appuyé ou non. Ce bit de poids fort étant initialement à 0, il suffit de le faire passer à 1 pour que le klaxon émette du bruit.

Référence du paragraphe : CPR_EMTT_PUISSANCE

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIFG_EMTT_PUISSANCE

Compétences GEII : C1-10, C1-11

On veut alimenter une diode infrarouge, il nous faut donc une résistance pour appliquer le courant adapté à l'intensité lumineuse demandée dans le cahier des charges. Seulement le courant de sortie de l'ATMEGA328P est de seulement 40 mA alors qu'il faut 200 mA pour obtenir une émission correspondant au cahier des charges. Il faut donc utiliser un transistor pour amplifier le courant.

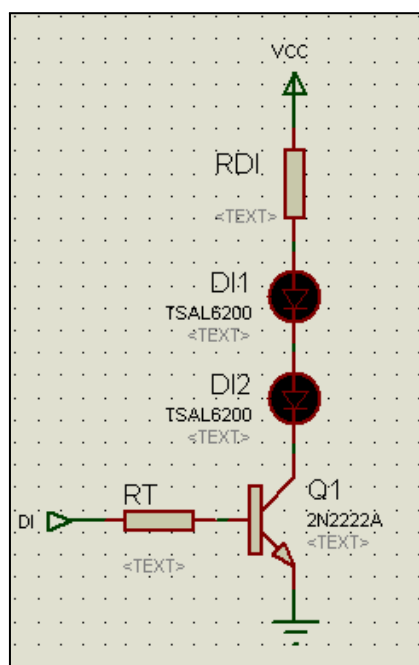


Figure 7 : Schéma préliminaire des leds infrarouges de l'étage action

Référence du paragraphe : CPR_EMTT_INDICATEUR

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_EMTT_INDICATEUR

Compétences GEII : C1-10, C1-11

On doit allumer une DEL à 50 mCd donc il faut y placer une résistance pour régler la tension de la diode et pouvoir régler l'intensité lumineuse. Nous avons une tolérance imposée de +/-20% donc il faut une résistance de série e6 ou e12. La LED choisie pour l'indicateur est de couleur vert car c'est une couleur habituellement utilisée en modélisme.

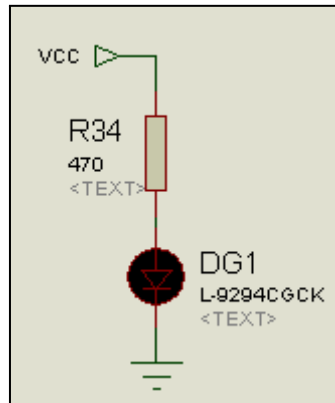


Figure 8 : Schéma électrique préliminaire de la led “indicateur”

Référence du paragraphe : CPR_EMTT_ENERGIE

Rédacteur : Frieih Mathieu et L’Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG_EMTT_ENERGIE

Compétences GEII : C1-10, C1-11

Pour répondre à cette exigence, nous devons utiliser un Lipo 2S.

Nous devons dimensionner le Lipo 2S en fonction de la consommation de la carte, sachant qu’il est considéré comme déchargé à 20% de sa capacité et qu’il fournit 7,4V.

Nous avons déterminé un schéma électrique préliminaire de la partie énergie, auquel nous avons ajouté un régulateur linéaire et des condensateurs de découplage afin de lisser la tension fournie par l’accumulateur et abaisser la tension reçue (7,4V) afin d’alimenter l’ATMEGA328P, la led infrarouge et la led verte à 5V . (voir Figure 9)

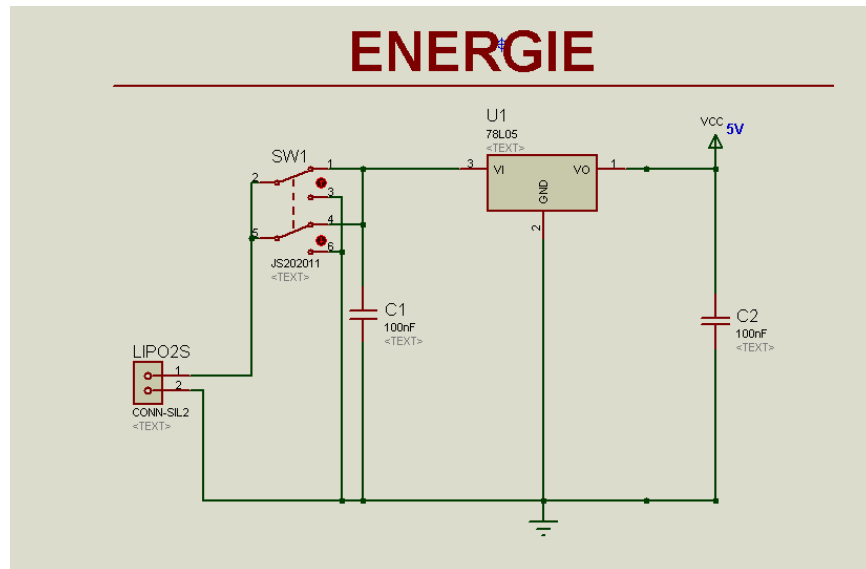


Figure 9 : Schéma électrique préliminaire de la partie énergie de l'émetteur

Référence du paragraphe : CPR_EMTT_INTERRUPTEUR

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG_EMTT_INTERRUPTEUR

Compétences GEII : C1b-22, C1b-26

Type d'étage retenu : Un interrupteur Switch JS202011CQN

Pour répondre à cette exigence, nous avons choisi d'intégrer un interrupteur entre le connecteur de l'alimentation et le régulateur.

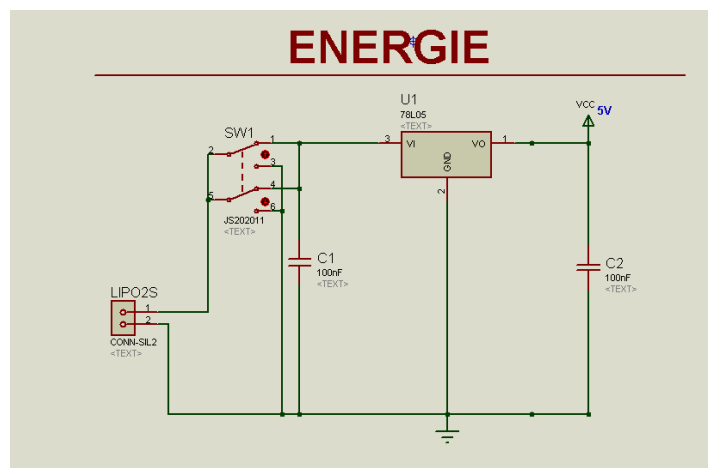


Figure 10 : Schéma électrique de la partie interrupteur de l'émetteur

Référence du paragraphe : CPR_EMTT_SCHEMA

Rédacteur : Frieih Mathieu, L'Haridon Mathis, Jerez Elouan et Roboam Damien

Relecteur : Frieih Mathieu, L'Haridon Mathis, Jerez Elouan et Roboam Damien

Compétences GEII : C1-10, C1-11

Voici, en Figure 10, le schéma électrique préliminaire de la partie émetteur du produit.

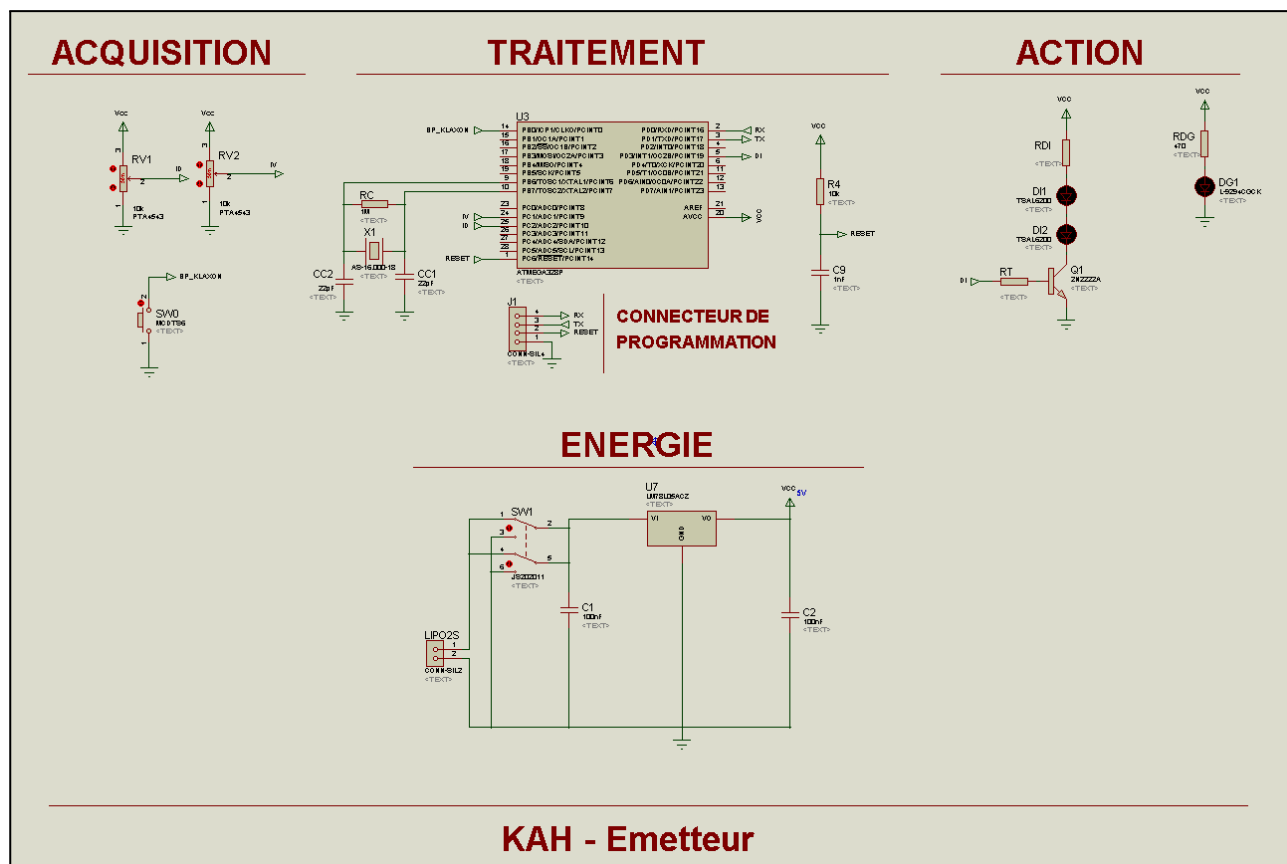


Figure 11 : Schéma électrique préliminaire de la partie émetteur du produit

2.2.2 Électronique - Récepteur

Référence du paragraphe : CPR_RCPT_ARCHI_ELEC

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Compétences GEII : C1-3, C1-9, C1-11

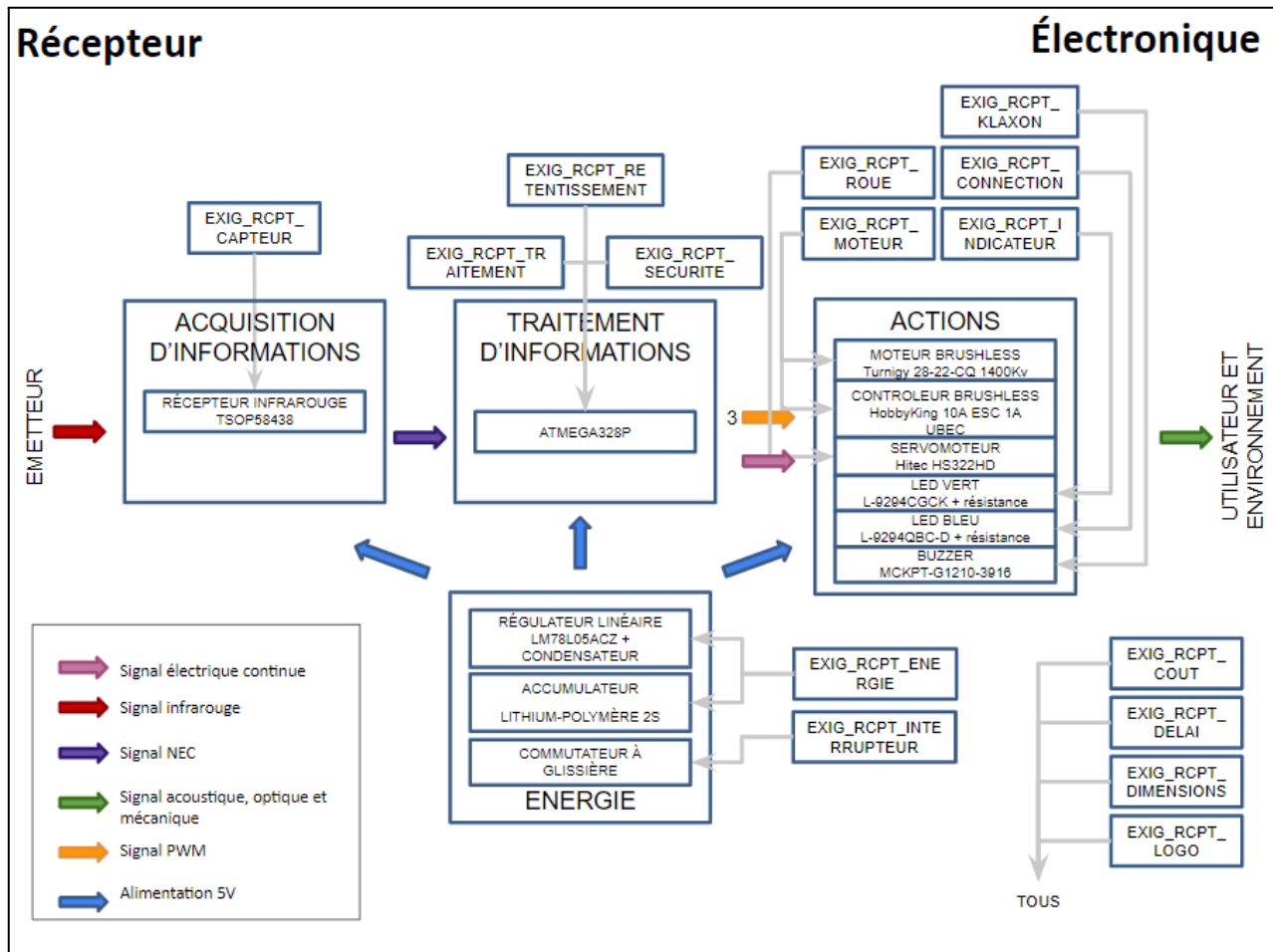


Figure 12 : Synoptique de l'architecture électronique de la partie récepteur du produit

Référence du paragraphe : CPR_RCPT_CAPTEUR

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_CAPTEUR

Compétences GEII : C1-10, C1-11

L'exigence EXIG_RCPT_CAPTEUR décrète que le produit doit réceptionner les trames protocolaires NEC émises par l'émetteur à l'aide d'un composant de réception infrarouge. Pour respecter l'exigence, nous avons sélectionné le récepteur infrarouge TSOP58436 qui liste le protocole NEC comme un des meilleurs protocoles à utiliser.

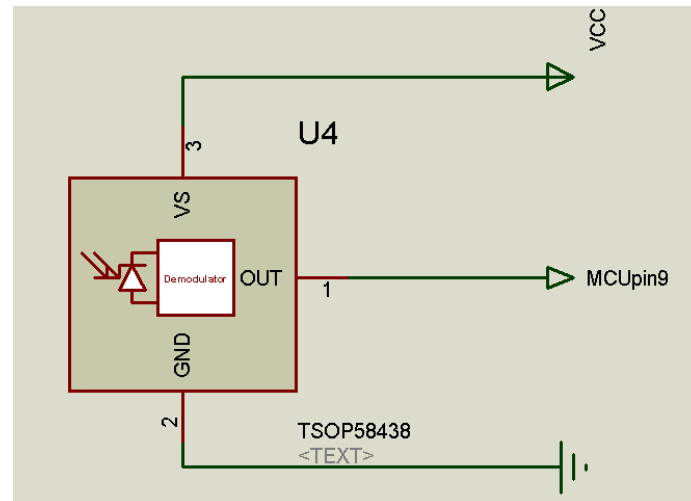


Figure 13 : Schéma électrique préliminaire du récepteur infrarouge

Référence du paragraphe : CPR_RCPT_TRAITEMENT

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées par pré-conception : EXIG_RCPT_TRAITEMENT

Compétences GEII : C1-10, C1-11

Nous avons retenu le microcontrôleur ATMEGA328P pour répondre à l'exigence de traitement car il nous faut un microcontrôleur avec au minimum 4 ports PWM (buzzer, servomoteur, moteur, LED bleu) ainsi qu'une fréquence supérieur à 38 kHz pour recevoir le signal NEC. Le microcontrôleur permet de répondre à cela et nous avons plus de connaissance sur ce microcontrôleur.

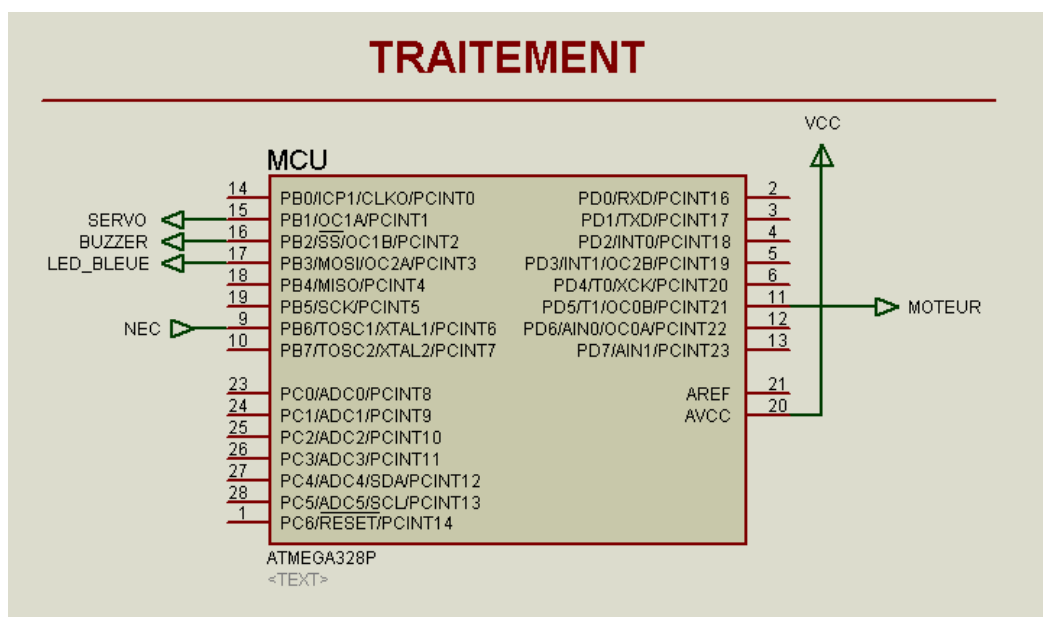


Figure 14 : Schéma électrique préliminaire de la partie traitement du récepteur

Référence du paragraphe : CPR_RCPT_SECURITE

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées par pré-conception : EXIG_RCPT_SECURITE

Compétences GEII : C1-10, C1-11

Nous avons pris en compte cette exigence dans la partie informatique de la conception, la puissance moteur est fixé à 0 par sécurité. Le microcontrôleur ATMEGA328P permet de réaliser ce type de fonction, nous utiliserons un périphérique "Timer" du MCU et le CPU.

Référence du paragraphe : CPR_RCPT_RETENTISSEMENT

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées par pré-conception : EXIG_RCPT_RETENTISSEMENT

Compétences GEII : C1-10, C1-11

Nous avons pris en compte cette exigence dans la partie informatique de la conception, le récepteur sera capable d'extraire l'information de retentissement du klaxon dans la partie adressage. Le microcontrôleur ATMEGA328P permet de réaliser ce type de fonction en récupérant la trame NEC reçue par le démodulateur infrarouge et en identifiant l'information avec un périphérique "Timer" et en utilisant le CPU.

Référence du paragraphe : CPR_RCPT_MOTEUR

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_MOTEUR

Compétences GEII : C1-10, C1-11

Nous utilisons le contrôleur brushless HobbyKing 10A ESC 1A UBEC, pour le piloter il faut que nous générons un signal PWM à l'aide de l'ATMEGA328P et de la fonction nommée PiloterMoteur() qui convertit l'information vitesse reçu en un signal PWM qui est ensuite transmis au contrôleur brushless. Nous utiliserons le CPU (Central Processing Unit) et un Timer (Périphérique) de l'ATMEGA328-P pour bien répondre à cette exigence.

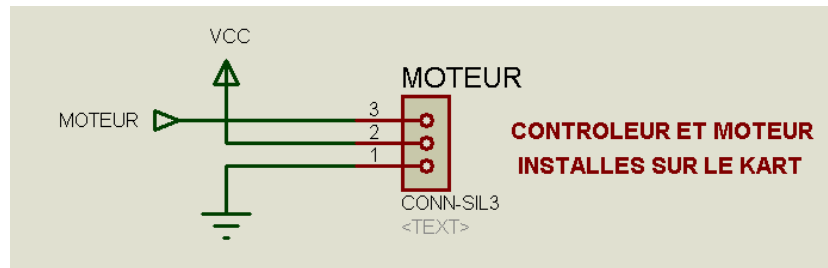


Figure 15 : Schéma électrique préliminaire du connecteur moteur

Référence du paragraphe : CPR_RCPT_ROUE

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_ROUE

Compétences GEII : C1-10, C1-11

Pour générer le signal PWM qui permette de contrôler l'angle du servo moteur nous avons décidé d'utiliser une fonction programmée dans l'ATMEGA328P nommée `PiloterServoMoteur()` qui convertit l'angle du servomoteur reçu dans la trame NEC en un signal PWM envoyé au servomoteur. Nous utiliserons le CPU (Central Processing Unit) et un Timer (Périphérique) de l'ATMEGA328-P pour bien répondre à cette exigence.

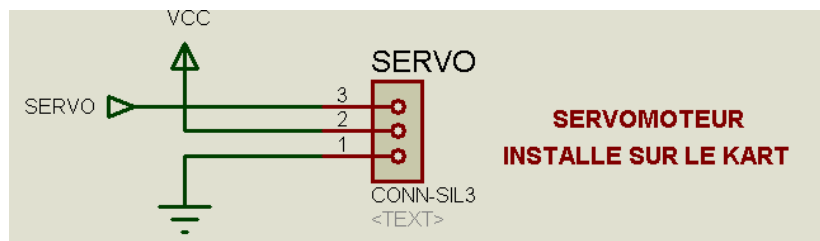


Figure 16 : Schéma électrique préliminaire du connecteur servo

Référence du paragraphe : CPR_RCPT_INDICATEUR

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_INDICATEUR

Compétences GEII : C1-10, C1-11

Pour le voyant informant que le récepteur est sous tension. Nous avons retenu la LED L-9294CGCK pour sa couleur verte qui est la couleur habituellement utilisée en modélisme pour l'indication de la mise sous tension et le fait que c'est la seule LED de cette couleur que nous avons au stock. La LED sera connectée directement à l'alimentation de la carte ce qui lui fournira un

courant suffisant. Nous avons aussi besoin d'une résistance de protection placée en série pour pouvoir réguler le courant qui la traverse.



Figure 17 : Schéma électrique préliminaire de l'indicateur de tension

Référence du paragraphe : CPR_RCPT_CONNEXION

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_CONNEXION

Compétences GEII : C1-10, C1-11

Pour cette exigence, nous avons retenu la LED L-9294QBC-D de couleur bleu. Cette couleur est habituellement utilisée en modélisme pour le transfert de données. La sortie de l'ATMEGA328P est capable de délivrer 20 mA ce qui est suffisant pour notre application. Nous avons aussi besoin d'une résistance de protection placée en série pour pouvoir réguler le courant qui la traverse.

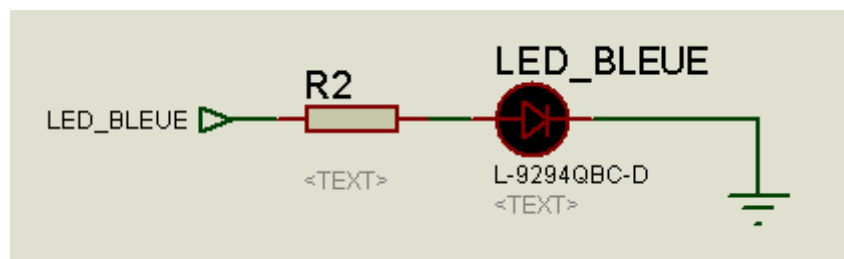


Figure 18 : Schéma électrique préliminaire de l'indicateur de connexion

Référence du paragraphe : CPR_RCPT_KLAXON

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_KLAXON

Compétences GEII : C1-10, C1-11

Pour le klaxon nous avons retenu le buzzer piézoélectrique MCKPT-G1210-3916 qui possède une tension de fonctionnement de 3 à 30 Vpp et un courant maximal de 2mA ce qui est bien dans les capacités de notre cœur de traitement. Le signal de 4 kHz sera généré par une fonction programmée sur l'ATMEGA328P nommée PiloterKlaxon() qui sera aussi responsable de transmettre le signal au buzzer. Nous utiliserons le CPU (Central Processing Unit) et un Timer (Périphérique) de l'ATMEGA328-P pour bien répondre à cette exigence.

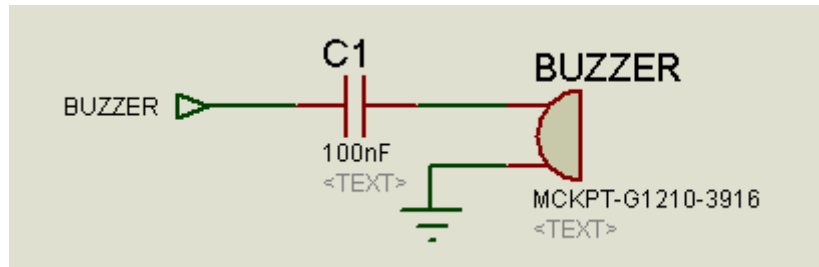


Figure 19 : Schéma électrique préliminaire du buzzer

Référence du paragraphe : CPR_RCPT_ENERGIE

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées par pré-conception : EXIG_RCPT_ENERGIE

Compétences GEII : C1-10, C1-11

Pour répondre à cette exigence, nous devons utiliser un Lipo 2S comme indiqué dans le cahier des charges.

Nous devons dimensionner le Lipo 2S dans la partie détaillée de la conception en fonction de la consommation de la carte, sachant qu'il est considéré comme déchargé à 20% de sa capacité et qu'il fournit 7,4V.

Nous avons déterminé un schéma électrique préliminaire de la partie énergie, auquel nous avons ajouté un régulateur linéaire intégré dans le contrôleur brushless, des condensateurs de découplage seront ajoutés dans la partie découplage afin de lisser la tension fournie par l'accumulateur et abaisser la tension reçue (7,4V) à 5V afin d'alimenter l'ATMEGA 328P correctement ainsi que les leds (bleu et verte), le buzzer et le démodulateur infrarouge.

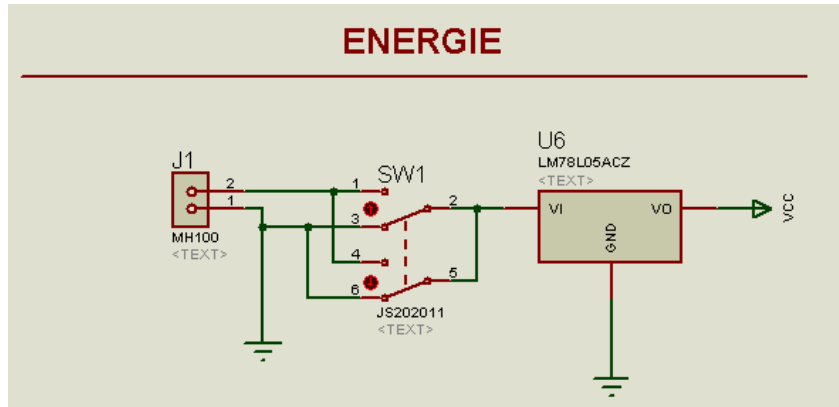


Figure 20: Schéma électrique préliminaire de l'étage énergie

Référence du paragraphe : CPR_RCPT_INTERRUPTEUR

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées par pré-conception : EXIG_RCPT_INTERRUPTEUR

Compétences GEII : C1-10, C1-11

Pour répondre à l'exigence interrupteur, un bouton poussoir est intégré au kart et se situe entre le LI-PO2s et le controleur brushless.

Référence du paragraphe : CPR_RCPT_SCHEMA

Rédacteur : Lestage Margot, Cassany Marie et L'Haridon Mathis

Relecteur : Bakas Laila et Larjuzan Noah

Compétences GEII : C1-10, C1-11

Voici, en Figure 22, le schéma électrique préliminaire de la partie émetteur du produit.

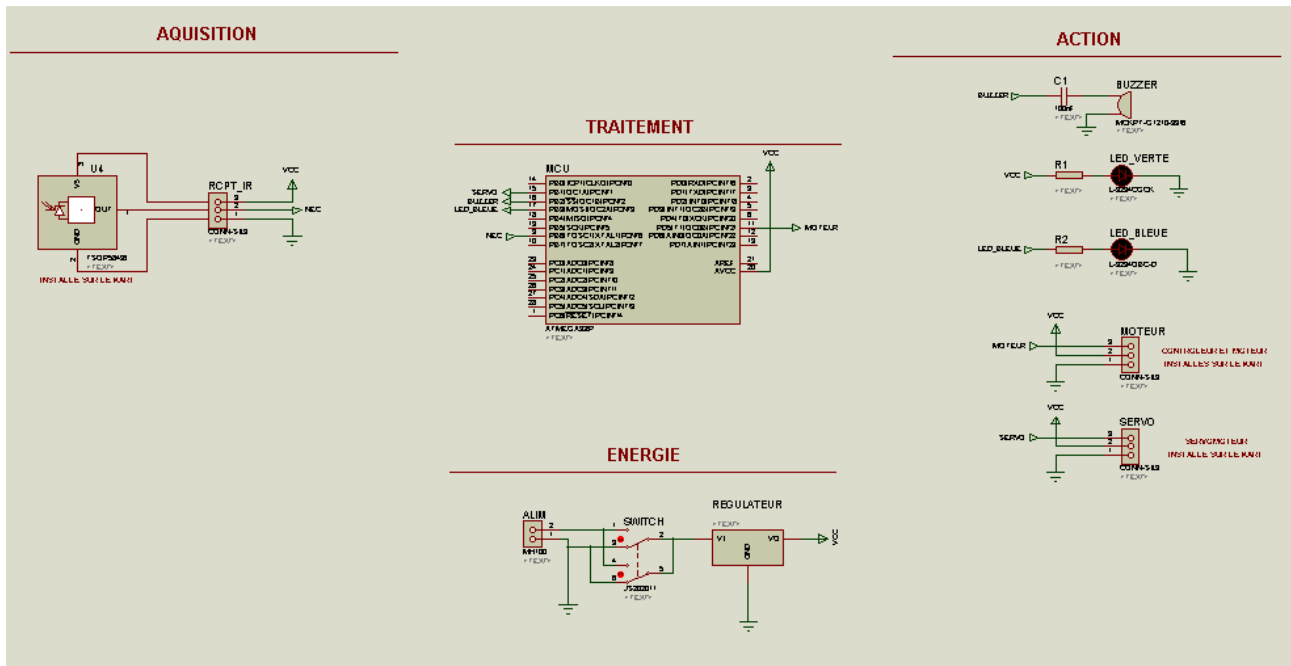


Figure 21 : Schéma électrique préliminaire du récepteur

2.3 Informatique

2.3.1 Informatique - Émetteur

Référence du paragraphe : CPR_EMTT_ARCHI_INFO

Rédacteur : Larjuzan Noah, Bakas laila, Cassany Marie, Lestage Margot et L'Haridon Mathis

Relecteur : Friehe Mathieu, Jerez Elouan et Roboam Damien

Compétences GEII : C1-3, C1-9, C1-10, C1-11

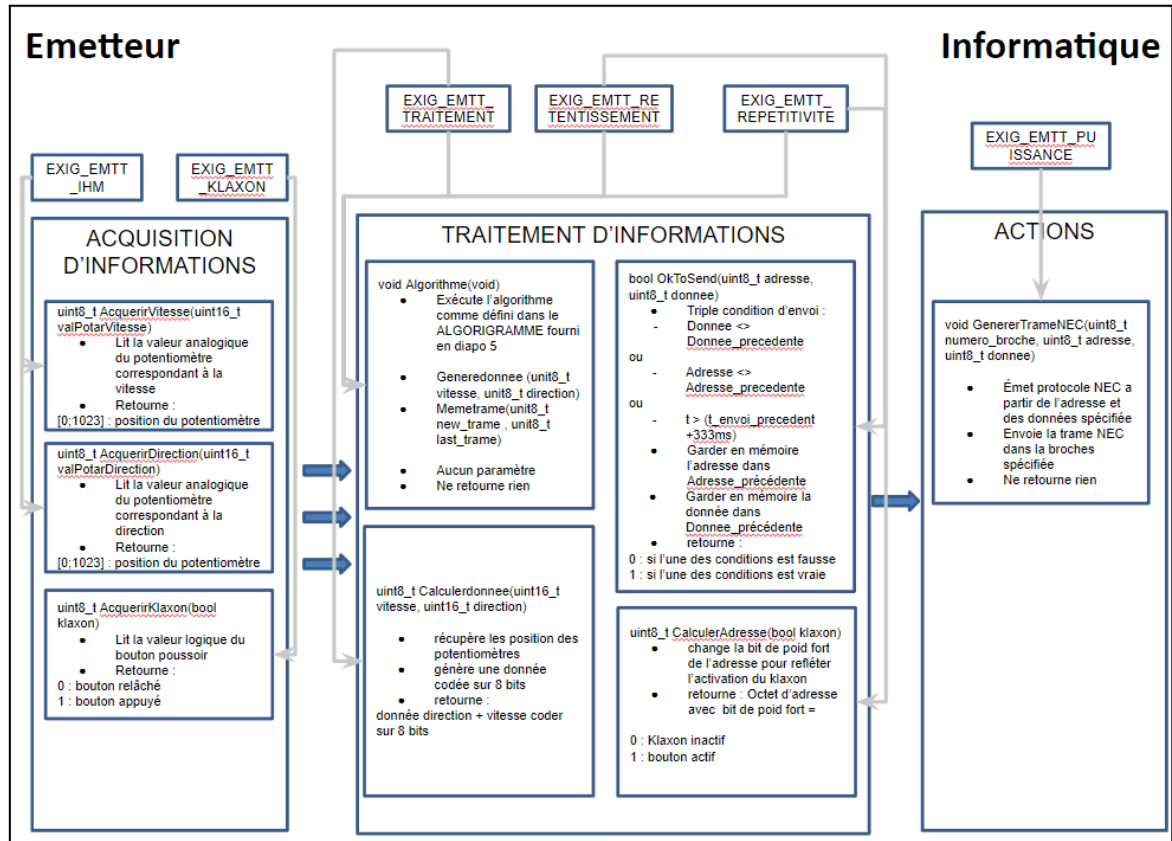


Figure 22 : Synoptique de l'architecture informatique de l'émetteur

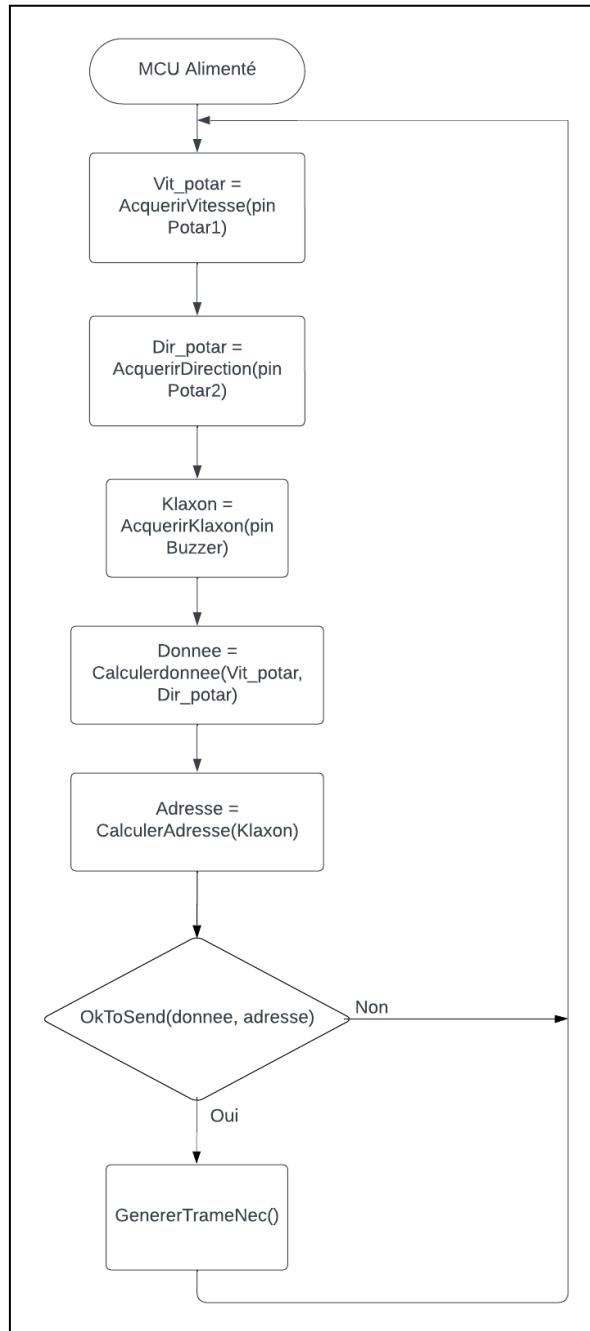


Figure 23 : Algorithme de traitement de l'émetteur

2.3.2 Informatique - Récepteur

Référence du paragraphe : CPR_RCPT_ARCHI_INFO

Rédacteur : Frieih Mathieu, L'Haridon Mathis, Jerez Elouan et Roboam Damien

Relecteur : Larjuzan Noah, Bakas laila, Lestage Margot et Cassany Marie

Compétences GEII : C1-3, C1-9, C1-10, C1-11

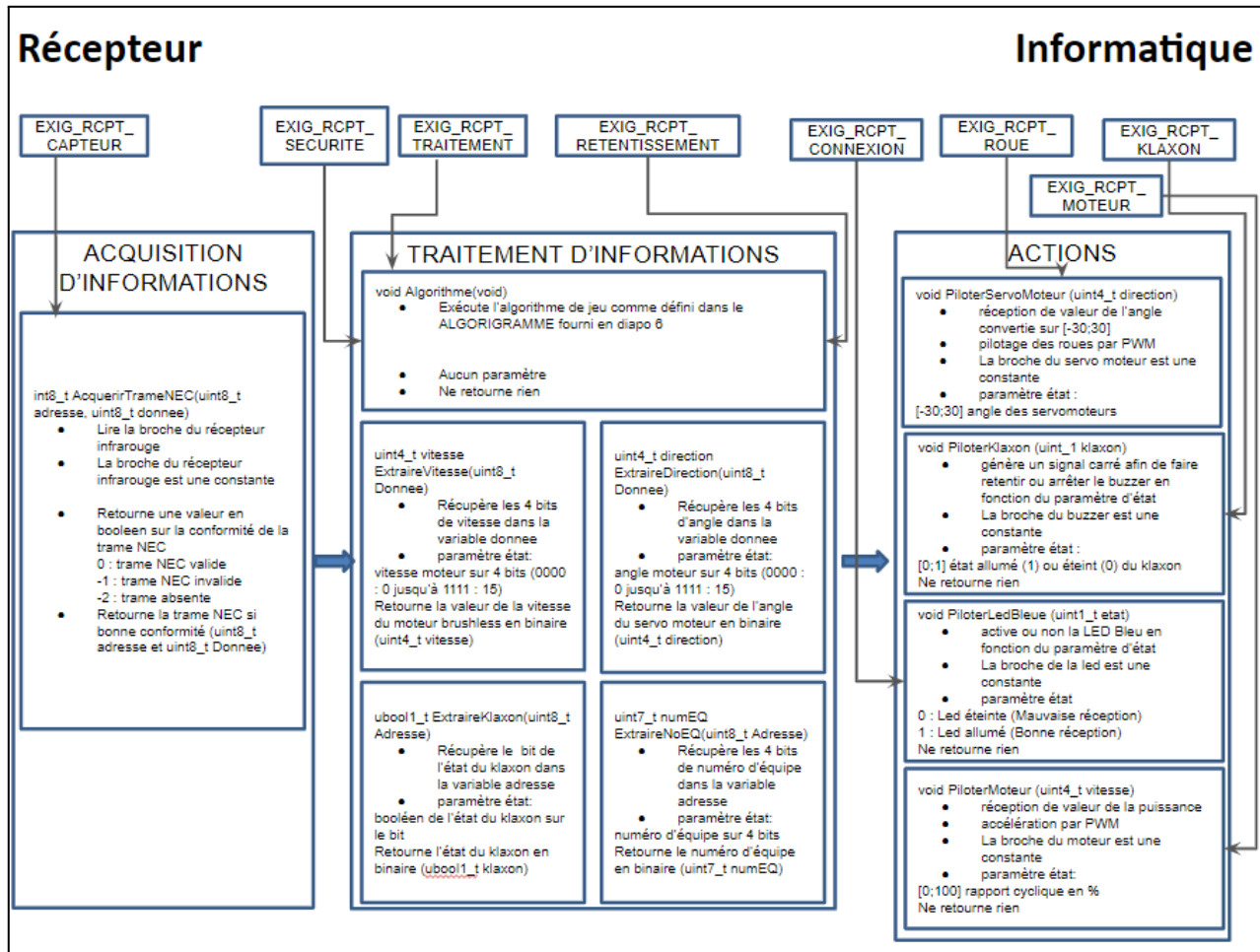


Figure 24 : Synoptique de l'architecture informatique du récepteur

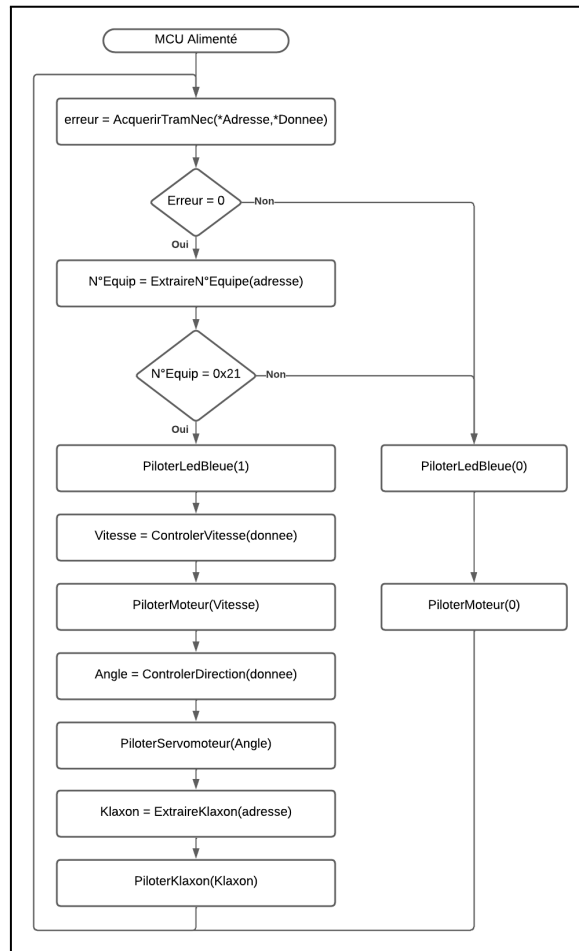


Figure 25 : Algorithme de traitement du récepteur

2.4 Coût - Délai

Référence du paragraphe : CPR_COUT

Rédacteur : Lestage Margot et Roboam Damien

Relecteur : L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_COUT

Compétences GEII : C1-10

Voici un tableau récapitulant le coût préliminaire du projet : [KAH_CDP](#).

Le coût du projet est de 82.70 euros ce qui respecte l'exigence.

КАН

[illegible]

Figure 26 : Tableaux du coût préliminaire du projet

Référence du paragraphe : CPR DELAI

Rédacteur : Bakas Laila et Lestage Margot

Relecteur : L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG DELAI

Compétences GEII : C1-10

D'après le planning prévisionnel, nous sommes bien dans les temps pour réaliser notre projet, pour le moment notre planning prévisionnel est en bonne voie pour être le planning définitif.

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 29/69 |
|----------------------------------|---|-------|

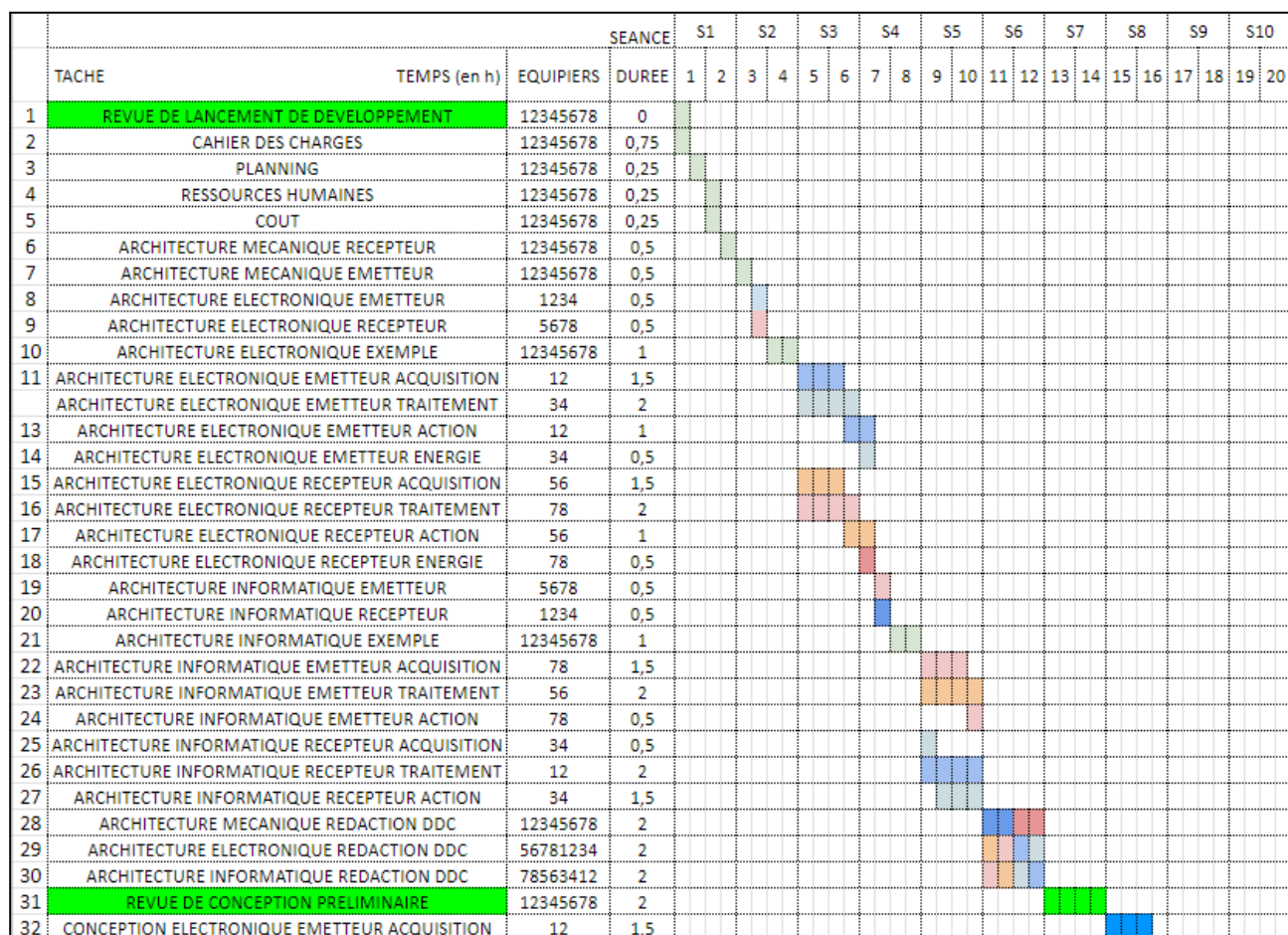


Figure 27 : Planning de la conception préliminaire

2.5 Conclusion de la conception préliminaire du produit

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : ROBOAM Damien et Lestage Margot

Cette phase de conception préliminaire nous a permis de nous familiariser avec le cahier des charges et d'élaborer une ébauche des schémas électriques de l'émetteur et du récepteur. Ainsi, nous avons choisi des solutions techniques pour chacune des exigences mentionnées. Nous certifions donc la faisabilité du projet et nous pouvons passer à la phase de conception détaillée afin de dimensionner les solutions retenues.

3. Conception détaillée du produit

Ce chapitre détaille la conception du produit développé. Il constitue une preuve de la conformité du produit. Chaque paragraphe de cette étude fait donc clairement référence aux exigences client issues du [CDC].

3.1 Électronique

3.1.1 Électronique - Émetteur

Référence du paragraphe : CDT_EM TT_IHM

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Mathieu Frieuh et Mathis L'Haridon

Exigences client vérifiées : EXIG_EM TT_IHM

Compétences GEII : C21/C22/C23/C25/C26

Bloc fonctionnel étudié : ACQUISITION

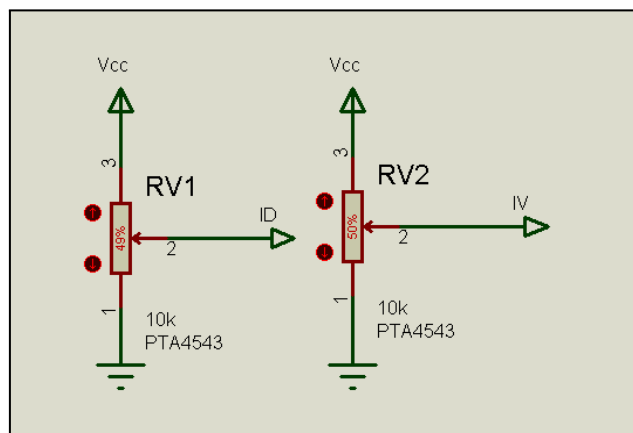


Figure 30 : Schéma électrique potentiomètres - émetteur acquisition

Préambule :

Un potentiomètre se comporte de la même manière qu'un pont diviseur de tension, nous pouvons donc utiliser la formule $V_{out} = V_{pont} * R_L / (R_H + R_L)$ comme vu sur la vidéo suivante : <https://eqrcode.co/a/ZPtBe1> => BUT 1ère Année/Semestre 1/Ressource n°5 - Comment transformer une tension continue entre une autre tension continue ?

Nous pouvons utiliser deux potentiomètres PTA4543-2015DP-B103 car la datasheet de l'ATEMEGA328P recommande une impédance d'entrée d'approximativement 10kΩ ou moins. Nos potentiomètres varient tous deux de 0Ω à 10kΩ. Ce qui fait qu'ils sont bien inférieurs à la résistance pull-up interne du MCU qui appartient à l'intervalle [30;50] kΩ.

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 32/69 |
|----------------------------------|---|-------|

Valeurs :

$$V_{out\ max} \leq 5V$$

$$V_{out\ min} \geq 0V$$

Nous avons ces inégalités puisque le potentiomètre a une incertitude de +/- 20%, mais nous ne pouvons avoir une tension supérieure à 5V et inférieure à 0. Cela va apporter une imprécision sur la position du potentiomètre, seulement cela n'aura pas d'impact sur l'utilisation du composant.

Pour l'exigence : EXIG_EMTT_IHM, ces potentiomètres sont suffisants, ils permettent donc bien la conformité de l'exigence.

Référence du paragraphe : CDT_EMTT_KLAXON

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Mathieu Friehe et Mathis L'Haridon

Exigences client vérifiées : EXIG_EMTT_KLAXON

Compétences GEII : C22/C24/C25/C26

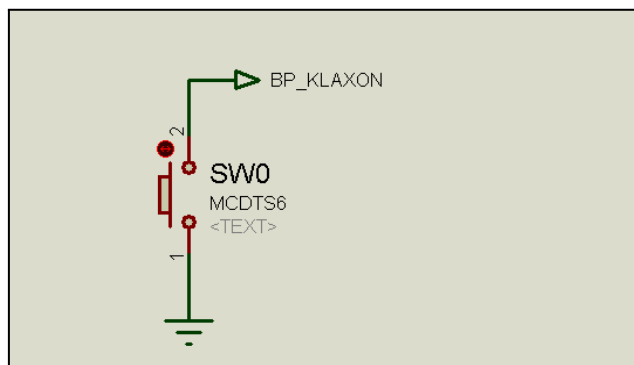
Bloc fonctionnel étudié : ACQUISITION

Figure 31 : Schéma électrique BP - émetteur acquisition

Préambule :

Nous prenons un petit boutons poussoir pour des raisons de dimensions, nous avons juste à le relier à la masse puisque l'ATMEGA328P a une résistance pull-up interne. Nous avons donc choisi le bouton MCDTS6-3K.

Cela permet au bouton qui actionnent le klaxon d'accomplir l'exigence **EXIG_EMTT_KLAXON**, l'exigence est donc bien vérifiée.

Calculés (Méthode : vidéo - <https://eqrcode.co/a/ZPtBe1> => BUT 1ère Année/Semestre 2/ Ressource n°23 - Comment dimensionner un étage binaire à transistor ?) :

- $V_{RDI} = V_{CC} - 2 \cdot V_{diode} - V_{CEsat} = 1.4V$
- $R3 = V_{RDI} / 200mA = 7\Omega \Rightarrow$ Normalisée, $R3 = 6.2\Omega$ (E24)
- $I_c = V_{RDI} / R3 = 0.225A$
- $I_B > I_c / h_{FE} \Rightarrow I_B > 3.75mA$
- $V_{RT} = V_{DI} - V_{BEsat} = 3.7V$
- $R2 < 3.7/0.00375 = 986.67\Omega \Rightarrow$ Normalisée, $R2 = 910\Omega$ (E24)

Alors $R2 = 910\Omega$ (E24) et $R3 = 6.2\Omega$ (E24).

Vérification des valeurs normalisées :

- $I_c = V_{RDI} / R3 = 225mA > 200mA$

Donc l'étage est conforme au cahier des charges.

Référence du paragraphe : CDT_EMTT_INDICATEUR

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Mathieu Friehe et Mathis L'Haridon

Exigences client vérifiées : EXIG_EMTT_INDICATEUR

Compétences GEII : C22/C23/C24/C25/C26

Nous sommes dans la bloc fonctionnel émetteur ACTION : schéma ci-dessus figure 32.

Le courant à l'entrée de la LED verte $I_v = I_{L5} \cdot I_{ref} / I_{L15} = 6.67E-3$

Avec les valeurs trouvée en datasheet : $I_{ref}=20mA$, $I_{L5}=50$ mcd, $I_{L15}=150$ mcd

Il faut dimensionner la résistance d'entrée : $R_v = (V_{CC} - V_{mcu\ bas}) / I_v = (5-2.1) / 6.67E-3 = 434.8\Omega$.

Une fois normalisée en série E12 (+/- 10%) la résistance est $R=470\Omega$.

(Méthode : vidéo - <https://eqrcode.co/a/ZPtBe1> => BUT 1ère Année/Semestre 2/ [Ressource n°7 - Comment faire briller une LED ?](#))

Avec cette résistance la DEL verte va s'allumer à 50 mcd à plus ou moins 20%. Cela répond à l'exigence : **EXIG_EMTT_INDICATEUR** et la rend conforme.

Référence du paragraphe : CDT_EMTT_TRAITEMENT

Rédacteur : Mathieu Friehe et Mathis L'Haridon

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées : EXIG_EMTT_TRAITEMENT

Compétences GEII : C22/C23/C24/C25

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 35/69 |
|----------------------------------|---|-------|

La réponse à cette exigence correspond au bloc fonctionnel traitement du synoptique l'architecture électronique de l'émetteur..

Nous avons choisi un microcontrôleur ATmega328-P parce qu'il possède plusieurs pins analogiques avec plusieurs ADC (Analog to Digital Converter), plusieurs pins PWM (Pulse Width Modulation) et une fréquence supérieure à 38 kHz pour recevoir le signal NEC. Ainsi, ce microcontrôleur répond parfaitement à l'exigence attendue.

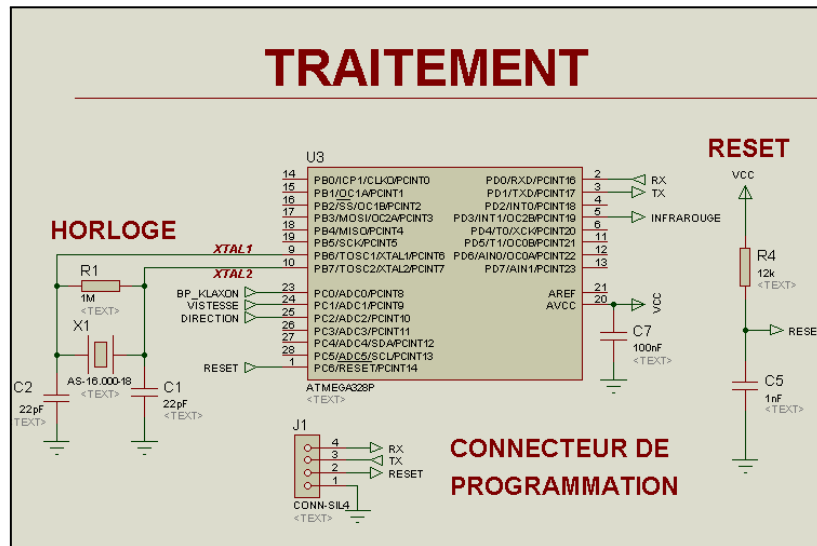


Figure 33 : Schéma de la partie traitement de l'émetteur

Nous avons dimensionné le microcontrôleur de notre émetteur (ATMEGA328P). Il a fallu décomposer le dimensionnement en 5 parties :

-Digital pins :

Selon le document constructeur, [Arduino-Uno-Pinout.png](#), les “digital pins” (les broches digitales) utilisables sont [PD0..,PD7], [PB0..,PB5] et [PC0..,PC5].

-Etage power supply pins :

Pour cet étage, nous avons simplement relié la tension Vcc aux broches 7 et 20 et les broches 8 et 22 à la masse (GND). Il faut noter que la tension de fonctionnement de l'ATMEGA328P est comprise entre 1.8 et 5.5 V.

-RESET :

L'étage reset est composé d'un condensateur et d'une résistance. Il faut dimensionner l'étage afin de le faire fonctionner afin des durées contrôlées.

D'après l'HTUT 22, la valeur standard de condensateur pour un étage reset est de 1nF.

Nous savons d'après la datasheet que la tension de seuil V_{IL} du microcontrôleur s'exprime par : $V_{IL}=0,2*V_{cc}$

Nous savons également d'après la datasheet du microcontrôleur que le temps de montée t_1 doit être supérieur à $2,5\mu s$.

De plus, la loi de l'évolution de la tension d'un condensateur en fonction du temps s'exprime par :

$$V_c(t) = V_{cc} - (V_{cc} - V_{ini})e^{-t/\tau}$$

D'après DS ATMEGA328P, $V(IL)[\text{seuil}] = 0,2 * V_{cc} = 1V$

avec t_1 le temps de commutation du reset

$V_{cc} = 5V$ (DS ATMEGA328P)

$V_{ini} = 0V$

$\tau = RC$ avec $C = 1nF$ (Condensateur normalisé standard)

$t_1 > 2,5\mu s$

$$V_c(t_1) > V_{cc} - (V_{cc} - V_{ini})e^{-t_1/\tau}$$

$$\Leftrightarrow -[(V_c(t_1) - V_{cc})/V_{cc}] > e^{-t_1/\tau}$$

$$\Leftrightarrow \ln[-(V_c(t_1)/5) + 1] > -t_1/\tau$$

$$\Leftrightarrow \tau > -(t_1 / \ln[-(V_c(t_1)/5) + 1]) > 11,2\mu s$$

$$\text{donc } R > \tau / C = 11,2 * 10^{-6} / 1 * 10^{-9} > 11200\Omega$$

$R_{\text{normalisée}} = 12000\Omega$ série E24(+/-5%)

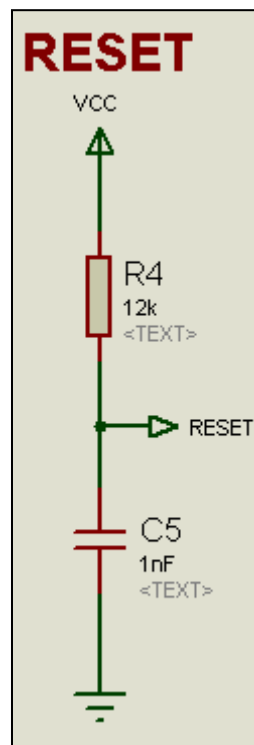


Figure 34 : Schéma de l'étage reset du MCU

Référence du paragraphe : CDT_EMTT_ENERGIE**Rédacteur :** Mathieu Friehe et Mathis L'Haridon**Relecteur :** Jerez Elouan et Roboam Damien**Exigences client vérifiées :** EXIG_EMTT_ENERGIE**Compétences GEII :** C22/C23/C24/C25

En accord avec la conception détaillée, nous avons gardé la batterie Lipo 2S de 7.4 V. Pour la dimensionner, nous avons fait la somme des principaux courants de notre carte et admis qu'il reste constant pendant une heure.

| Composants | Consommation en mA |
|--------------------|--------------------|
| MCU ATMEGA328-P | 9 |
| LED Vert | 6.67 |
| Potentiomètre (*2) | 1 |
| LED Infrarouge | $200 / 9 = 22.2$ |

Figure 37 : Tableau de consommation des composants de l'émetteur

Précision pour les deux leds infrarouges :

Tout d'abord, les deux leds infrarouges permettant de transmettre la trame NEC sont en série donc le même courant traverse les deux LED infrarouges. Ensuite, pour leur bon fonctionnement, elles ont besoin d'un courant de 200 mA. Enfin, lors d'un envoi de la trame NEC, l'état haut (le moment où le courant doit être à 200 mA) représente un neuvième du temps. Ainsi, nous pouvons réduire le courant à $200 / 9 = 22.2$ mA (ce qui fait un rapport cyclique d'environ 11% ($22,2/200$)).

En effectuant la somme des courants supposés constants pendant une heure, nous trouvons une consommation de 39.87 mA.

$$\text{Somme des courants} = 22.2 + 9 + 2 + 6.67 = 39.87 \text{ mA}$$

$$\text{Consommation 1 heure} = 39.87 \text{ mAh}$$

Aussi, nous devons prévoir de ne pas consommer en dessous des 20% de la capacité du Lipo 2S. Avec le Lipo 2S de 350 mAh, nous disposons de 350×0.8 mAh soit 280mAh. Or, $280 \text{ mAh} > 39,87 \text{ mAh}$.

Nous pouvons calculer l'autonomie de l'accumulateur par la formule $t=E/I$ avec E la capacité de l'accumulateur en milliampères heure (mAh) et I la consommation en ampères(A). Ainsi, $t=280/39,87=7\text{h}$. Donc l'émetteur aura une autonomie supérieure à 1h et le dimensionnement est conforme à l'exigence "Énergie".

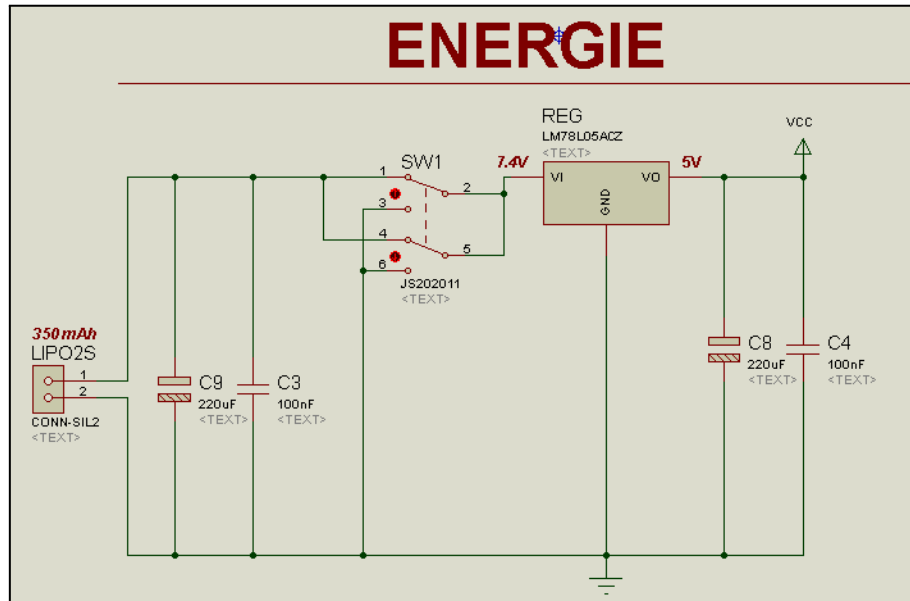


Figure 38 : Schéma électrique du bloc énergie de l'émetteur

Nous avons besoin d'abaisser la tension délivrée par l'accumulateur pour alimenter le reste du circuit (passage de 7,4V à 5V). Ainsi, nous avons choisi le régulateur linéaire LM78L05ACZ, modèle TO-92.

Le régulateur linéaire a un courant de sortie de 100mA et nos LEDs infrarouges ont besoin de 200mA pour fonctionner durant 1 neuvième du temps. Or, nous pouvons nous contenter de choisir le régulateur simplement en fonction de la valeur moyenne des courants car les condensateurs de découplage vont permettre d'apporter le courant nécessaire au bon fonctionnement des LEDs. Nous l'avons choisi car il est le moins cher.

Vout: 5V

Vin: 7,4V

Imprécision : +/-5%

Rthj-a: 150°C/W

Pregu = (Vin-Vout)*Iout (C.f HTUT 5)

Pregu = (7,4-5)*0,1=0,24W Tregu = Tambiant+Rthj-a*Pregu=61°C

Et 0<61<150 Le régulateur est adapté à notre utilisation.

Aussi, nous avons ajouté des condensateurs de découplage afin de limiter les fluctuations de courants et ainsi éviter d'éventuels dysfonctionnements. Ils stabilisent aussi la tension aux bornes des composants et sont placés au plus près des composants qu'ils découplent.

Nous les avons dimensionnés selon l'inégalité (donnée dans l'HTUT 25):

$C > I \frac{\Delta t}{\Delta V}$ avec Δt la période des pics de courant, ΔV la fluctuation autorisée de la tension (valeur standard 0,01 Vcc) et I la consommation du composant.

$$C_6 > 22,2 \times 10^{-3} \times \frac{26,3 \times 10^{-6}}{0,05} = 11,7 \mu F$$

Ainsi, $C_6 = 22 \mu F$ or la fréquence d'allumage de la LED infrarouge est incompatible avec la gamme de fréquence de fonctionnement du condensateur. Nous sommes dans l'obligation de rajouter en parallèle un condensateur de technologie différente et supportant la fréquence de 38kHz.

Ainsi, on doit prendre un condensateur 100 fois plus petit pour assurer le découplage. On ajoutera donc un condensateur $C_{10} = 220 nF$ car les condensateurs en polyester supportent des fréquences de 100kHz. (Voir figure 39)

$$C_7 > 9 \times 10^{-3} \times \frac{1}{\frac{16 \times 10^6}{0,05}} = 11,25 nF$$

Donc $C_7 = 100 nF$ Ici les fréquences correspondent, C_7 suffit à découpler la broche AVCC du MCU. (voir figure 40)

Aussi, nous devons découpler l'accumulateur et le régulateur linéaire. Pour cela, nous réutilisons les valeurs calculées précédemment. (voir figure 38(schéma energie complet)).

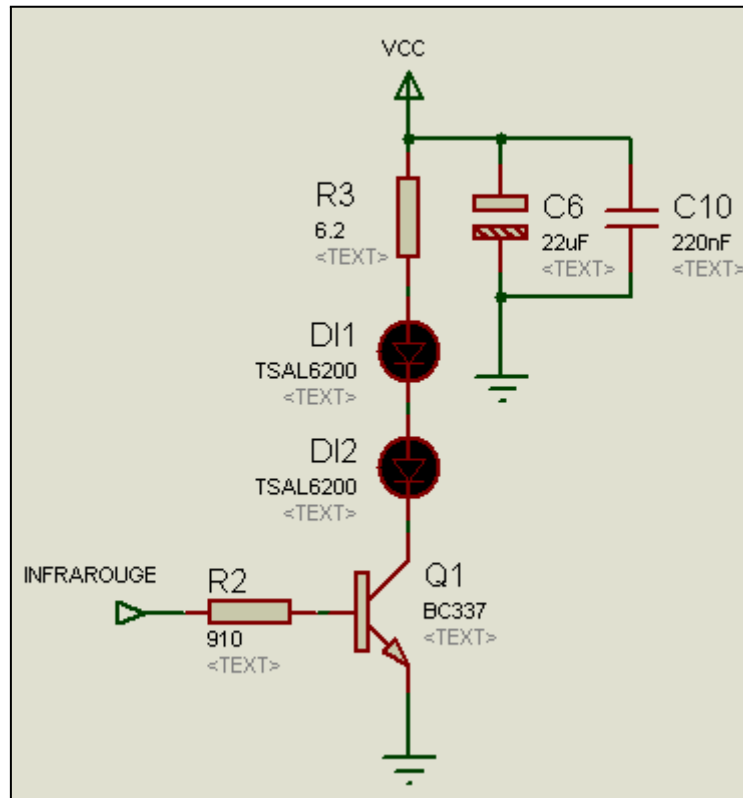


Figure 39 : Schéma de l'étage à LEDs infrarouges

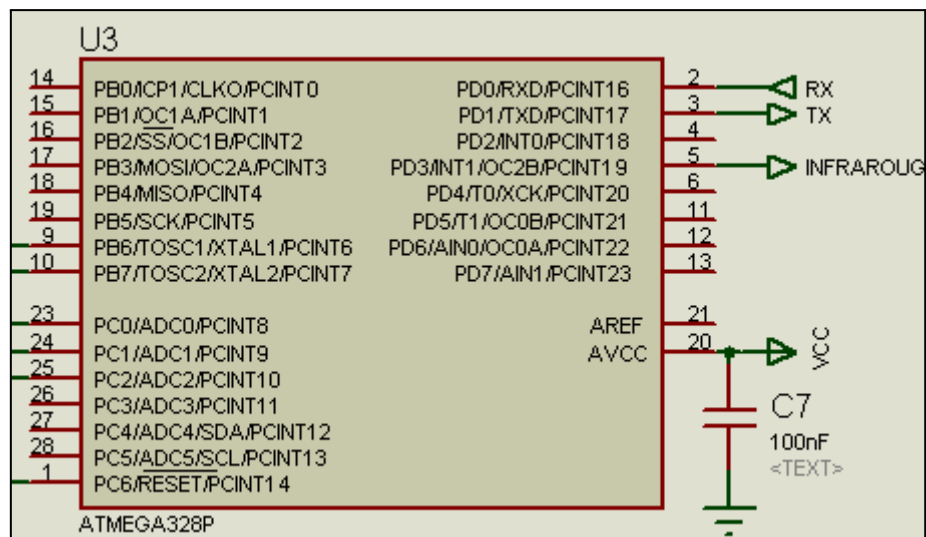


Figure 40 : Schéma du MCU

Référence du paragraphe : CDT_EMTT_INTERRUPTEUR**Rédacteur :** Mathieu Friehe et Mathis L'Haridon**Relecteur :** Jerez Elouan et Roboam Damien**Exigences client vérifiées :** EXIG_EMTT_INTERRUPTEUR**Compétences GEII :** C1b-22, C1b-26

Nous avons choisi de garder l'interrupteur Switch JS202011CQN comme prévu en conception préliminaire. Il assure la mise sous/hors tension de l'émetteur. Nous avons branché les broches 1 et 4 sur le Vcc de la LIPO 2s, les broches 3 et 6 sur la masse GND et les broches 2 et 5 sur l'entrée du régulateur linéaire.

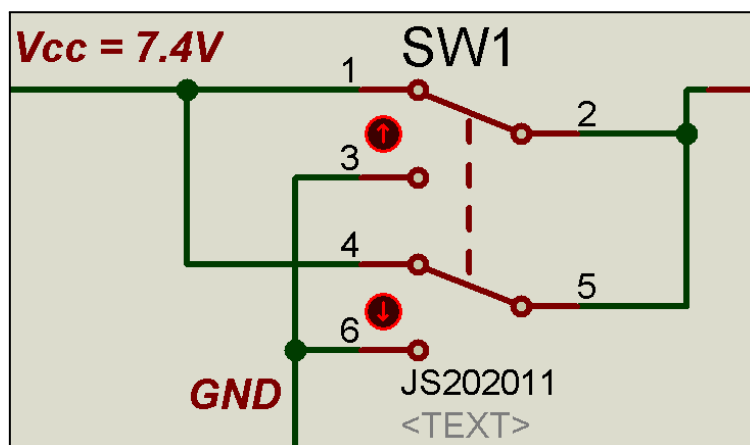


Figure 41 : Schéma électrique de l'interrupteur de l'émetteur

3.1.2 Électronique - Récepteur

Référence du paragraphe : CDT_RCPT_ARCHI_ELEC**Rédacteur :** Lestage Margot et Cassany Marie**Relecteur :** Bakas Laila et Larjuzan Noah**Compétences GEII :** C1-25

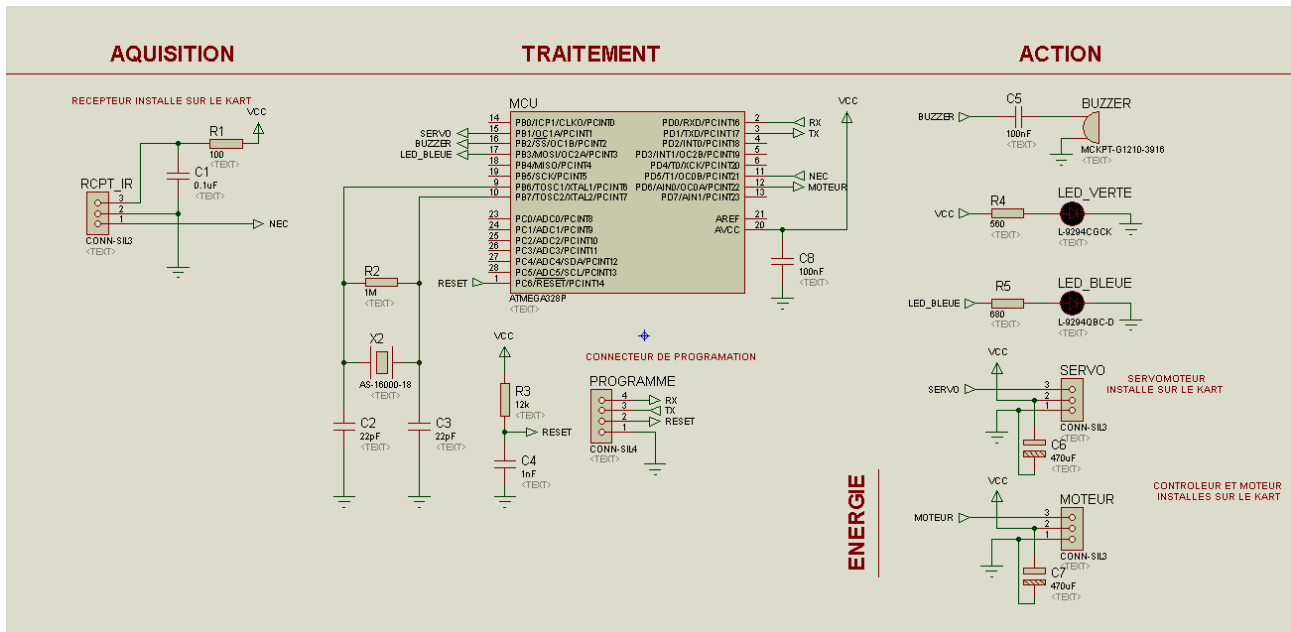


Figure 42 : Schéma électronique complet du récepteur

Référence du paragraphe : CDT_RCPT_ENERGIE

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées : EXIG_RCPT_ENERGIE

Compétences GEII : C22/C23/C24/C25

Pour dimensionner le LIPO 2s nous avons regardé chaque datasheet de tous les composants actifs pour vérifier chaque consommation de courant. Le moteur consomme 5A au maximum de sa puissance et consomme beaucoup plus que tous les autres composants (de l'ordre de mA) donc nous pouvons les négliger.

L'accumulateur doit assurer une autonomie d'au moins 15 minutes à mi-puissance donc nous prendrons 2.5A de consommation pour le moteur.

Calcul:

$$E = I \times t = 2.5 \times 0.25 = 0.625 \text{ Ah}$$

Nous ajoutons 20% pour ne pas abîmer l'accumulateur :

$$E_{tot} = 0.625 + 0.625 \times 0.20 = 0.750 \text{ Ah}$$

Donc nous aurons besoin d'un accumulateur LIPO 2s de 1000 mAh pour alimenter le contrôleur brushless et donc notre carte récepteur.

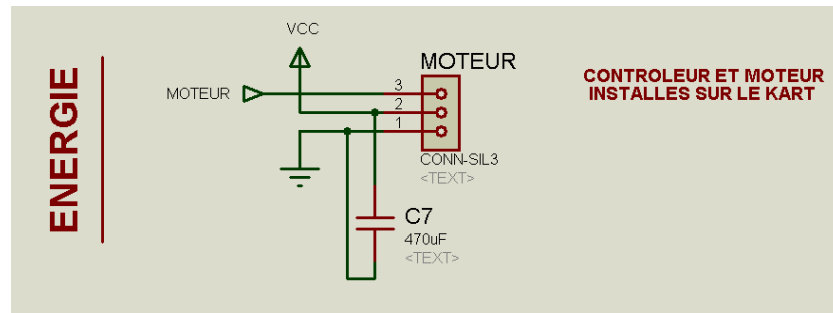


Figure 43 : Schéma électrique récepteur énergie

Nous avons calculé les condensateur de découplage pour les composants qui le nécessite avec la formule vu en cours : $C > I \times (\frac{\Delta t}{\Delta V})$. Pour ΔV la fluctuation autorisé, on prend 1% de l'alimentation, soit $0.01 \times 5V$ et Δt la période des pics. L'intensité se trouve dans les datasheets des constructeurs et la plupart des Δt est assumé par nous même en fonction de comment on les utilise.

Calcul :

Pour le servomoteur : $C > 700 \times 10^{-3} \times (\frac{0.1}{0.1 \times 5}) \Leftrightarrow C > 140 \text{ mF}$ avec $\Delta t = 0.1s$ car le condensateur va servir que pour compenser la réaction du BEC d'environ 0.1s, donc pour apporter ce qu'il manque.

Mais nous ne pouvons pas prendre une valeur aussi grande donc de manière empirique nous prenons la valeur la plus grande que nous avons donc $C_{\text{Servo}} = C_6 = 470\mu F$ pour sa valeur normalisée.

Pour l'alimentation : $C > 470\mu F$ car lors du calcul la valeur est elle aussi de l'ordre des millifarad donc nous prenons aussi une valeur qui est empirique et pas trop encombrante pour notre carte. Donc $C_{\text{alim}} = C_7 = 470\mu F$ en valeur normalisée.

Pour le MCU : $C > 9 \times 10^{-3} \times \frac{1}{\frac{16 \times 10^6}{0.05}} = 11,25 \text{ nF}$ mais d'après le HTUT 25 nous prenons au dessus donc $C_{\text{MCU}} = C_8 = 100 \text{ nF}$ pour sa valeur normalisée.

Référence du paragraphe : CDT_RCPT_CAPTEUR

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées : EXIG_RCPT_CAPTEUR

Compétences GEII : C22/C23/C24/C25

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 45/69 |
|----------------------------------|---|-------|

Bloc fonctionnel étudié : ACQUISITION

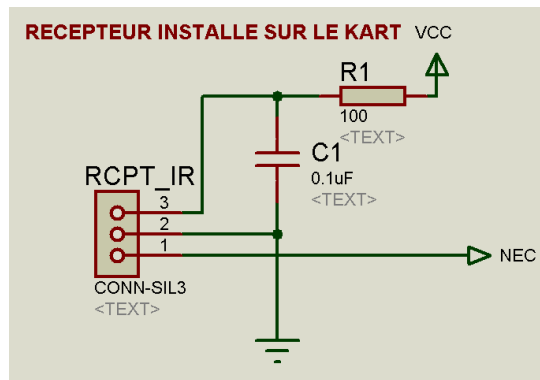


Figure 44 : Schéma électrique récepteur acquisition

Nous avons choisi le TSOP4438 pour son coût inférieur aux alternatives accessibles. la datasheet du composant [TSOP4438.pdf](#) nous recommande d'utiliser le montage suivant en fixant les valeurs de C1 à 0.1uF et R1 à 100 Ohms :

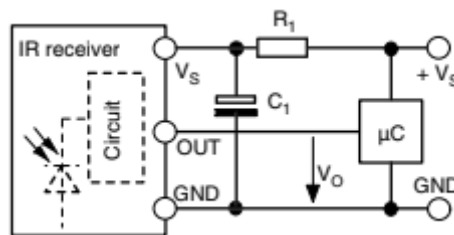


Figure 45 : montage de dérisquage de l'étage

Pour dérisquer notre choix nous avons assemblé notre montage sur une plaque à trous et avons visualisé la tension sortante de notre composant à l'aide d'un oscilloscope. Nous avons ensuite utilisé un émetteur pour envoyer une trame NEC connue qui est bien apparue sur notre oscilloscope. Notre montage est donc bien conforme au cahier des charges.

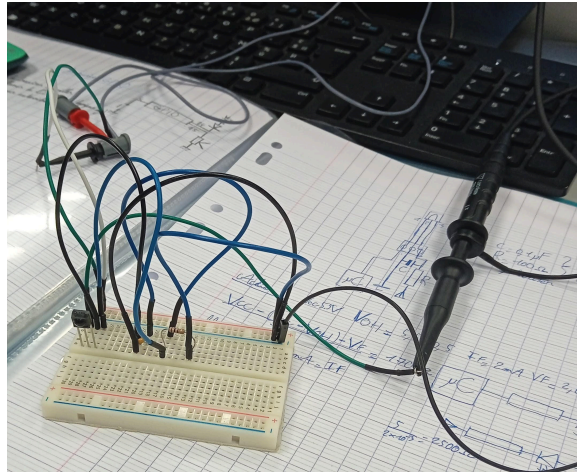


Figure 46 : montage de dérissage de l'étage

Référence du paragraphe : CDT_RCPT_TRAITEMENT

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Bakas Laila et Larjuzan Noah

Exigences client vérifiées : EXIG_RCPT_TRAITEMENT

Compétences GEII : C22/C23/C24/C25

Dans la partie traitement, nous devons dimensionner le microcontrôleur ATMEGA328P.

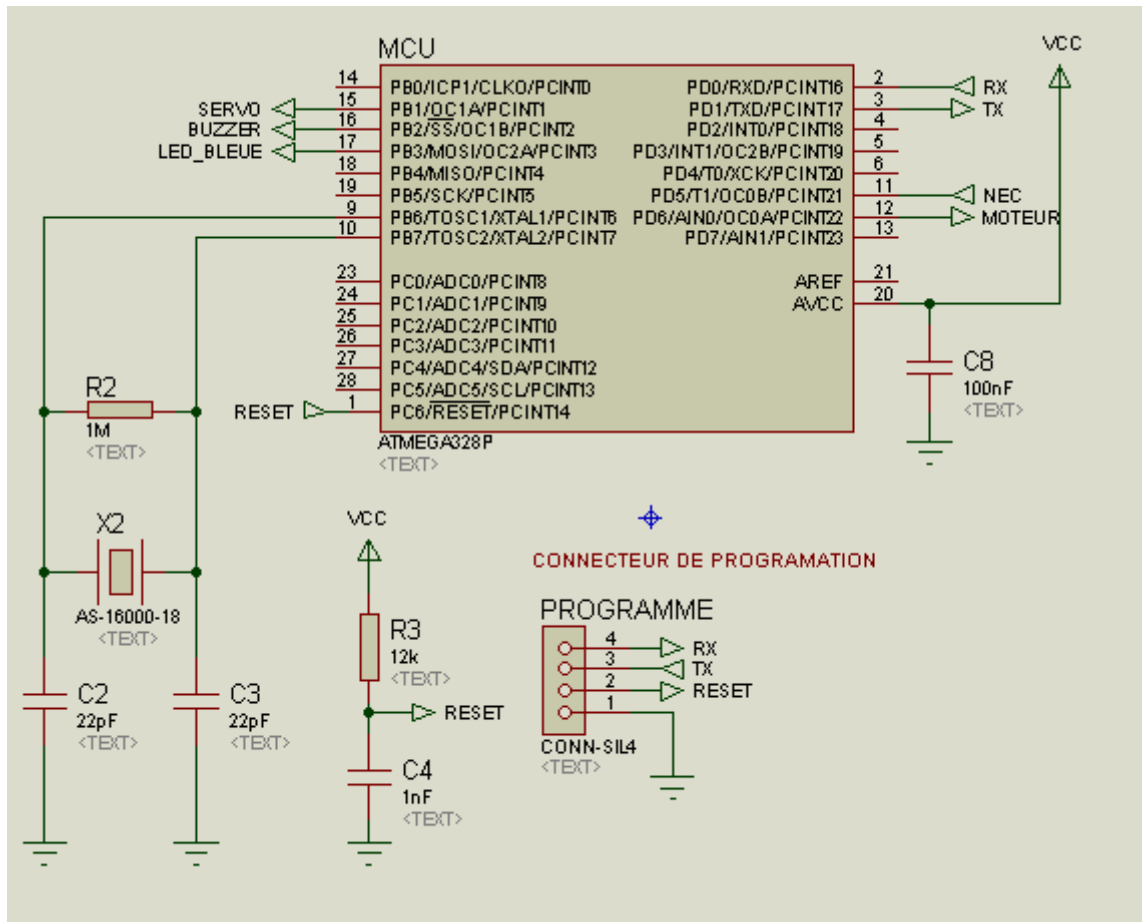


Figure 47 : Schéma électrique récepteur traitement

Etage power supply pins :

Pour cet étage, nous avons simplement relié la tension Vcc et la masse GND aux broches correspondantes sur le microcontrôleur ATMEGA328P.

Etage reset :

L'étage reset est composé d'un condensateur et d'une résistance. Il faut dimensionner l'étage afin de le faire fonctionner pour des durées contrôlées.

De plus, la loi de l'évolution de la tension d'un condensateur en fonction du temps s'exprime par :

$$V_c(t) = V_{cc} - (V_{cc} - V_{ini})e^{-t/\tau}$$

D'après la DataSheet de l'ATMEGA328P :

$$V(IL)[\text{seuil}] = 0,2 * V_{cc} = 1V$$

Avec $t_1 > 2,5 \mu s$ le temps de commutation du reset

$$V_{CC} = 5V$$

$$V_{ini} = 0V$$

$$\tau = RC \text{ et } C = 1nF \text{ (Standard)}$$

$$t1 > 2,5\mu s$$

Calcul:

$$V_c(t1) > V_{CC} - (V_{CC} - V_{ini}) * e^{-t1/\tau}$$

$$\Leftrightarrow -[(V_c(t1) - V_{CC}) / V_{CC}] > e^{-t1/\tau}$$

$$\Leftrightarrow \ln[-(V_c(t1)/5) + 1] > -t1/\tau$$

$$\Leftrightarrow \tau > -(t1 / \ln[-(V_c(t1)/5) + 1]) > 11,2\mu s$$

$$\text{donc } R > \tau / C = 11,2 * 10^{-6} / 1 * 10^{-9} > 11\,200\Omega$$

$$R_N = 12\,000\Omega \text{ série E24}(+/-5\%)$$

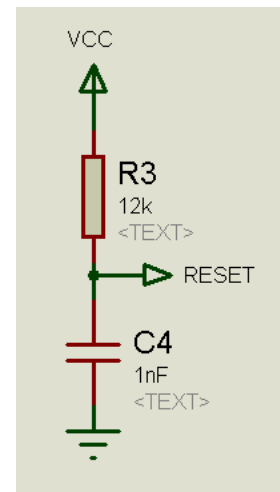


Figure 48 : Schéma de l'étage reset du récepteur

Clock pins :

Pour la partie horloge du microcontrôleur, nous avons choisi le quartz AS-16.000-18 relié aux broches XTAL1 et XTAL2.

Il possède une fréquence de 16 MHz qui est comprise dans l'intervalle de fréquence de la datasheet du microcontrôleur (10 MHz - 20 MHz).

Donc d'après la DataSheet de l'ATMEGA328P :

$$R2 = 1\,M\Omega \text{ (E24)}$$

$$C2 \text{ et } C3 = 22pF$$

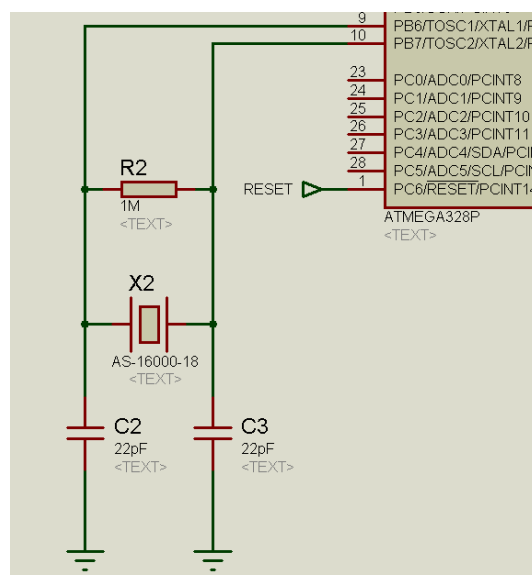


Figure 49 : Schéma de l'horloge de la partie Traitement du récepteur

Etage connecteur de programmation :

Nous avons besoin d'un connecteur mâle HE14 MH100 à 4 broches : 1 -> GND, 2 -> RST, 3 -> TX, 4 -> RX

L'ordre est important parce que nous avons un adaptateur qui transforme le signal USB d'un ordinateur en signal UART compréhensible par le microcontrôleur. Cet adaptateur pourra être réutilisé pour fabriquer d'autres cartes.

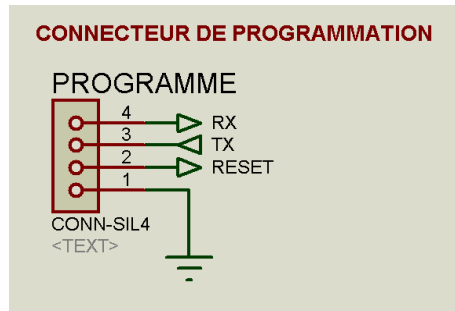


Figure 50 : Schéma électrique récepteur traitement

Référence du paragraphe : CDT_RCPT_INDICATEUR

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées : EXIG_RCPT_INDICATEUR

Compétences GEII : C22/C23/C24/C25/C26

Bloc fonctionnel étudié : ACTION

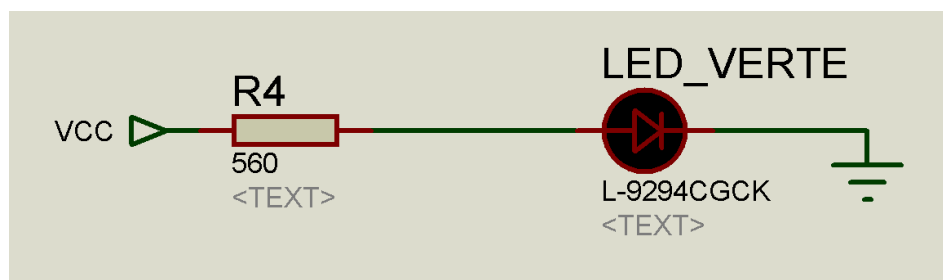


Figure 51 : Schéma électrique récepteur de la partie action (indicateur)

Préambule :

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 50/69 |
|----------------------------------|---|-------|

Après que nous avoir retenue la LEDL-9294CGCK de couleur bleu, nous avons aussi besoin d'une résistance de protection placée en série pour pouvoir réguler le courant qui la traverse. Pour cela on vas calculer la valeur de résistance convenable : à l'aide du CDC et la datasheet de la LED [L-9294CGCK.pdf](#) et selon la Méthode de vidéo - <https://eqrcode.co/a/ZPtBe1> => BUT 1ère Année/Semestre 2/[Ressource n°7 - Comment faire briller une LED ?](#)

1/ Intensité lumineuse :

L'intensité lumineuse du voyant est de 50 mCd (-/+20%) d'après le cahier des charges.

2/ Intensité électrique :

Grâce à des informations trouvées dans la datasheet.

$$I_F = (50 \times 20) / 150 = 6.66 \text{ mA}$$

3/ Tension électrique :

Grâce à ces valeurs de I_F nous avons pu trouver la tension V_F correspondante dans la datasheet.

$$V_F = (6.66 \text{ mA} \times 2V) / 10 \text{ mA} = 1.32V$$

4/ Résistance

$$R = (V_{CC} - V_F) / I_F$$

Avec $V_{CC} = 5V$

$$R = (5 - 1.23) / 6.66 \times 10^{-3}$$

$$R = 552 \Omega$$

| Nom | V_F (en V) | I_F (en mA) | Intensité lumineuse (en mCd) | R calculée (en Ω) | R normalisé (en Ω) série E12($\pm 10\%$) |
|-----------|--------------|---------------|------------------------------|---------------------------|--|
| LED verte | 1.32 | 6.66 | $50 \pm 20\%$ | 552 | 560 |

5/ montrer la conformité de la solution technique retenue par simulation à l'aide d'ISIS:

Nous avons calculé un courant I_F de 6.66 mA, d'autre part nous avons mesuré un courant I_F de 6.57 mA dans la simulation et ces deux valeurs ont un écart de -1,35% ce qui est inférieur au 20% alouer par le cahier des charges, donc on conclut que la résistance retenue **R=560 Ω -/+10% ; serie E12** est conforme à l'exigence.

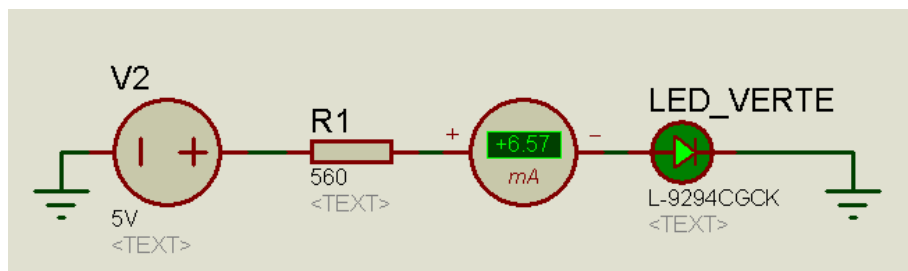


Figure 52 : Schéma électrique récepteur de la simulation de la partie action (indicateur)

Référence du paragraphe : CDT_RCPT_CONNEXION

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées : EXIG_RCPT_CONNEXION

Compétences GEII : C22/C23/C24/C25/C26

Bloc fonctionnel étudié : ACTION

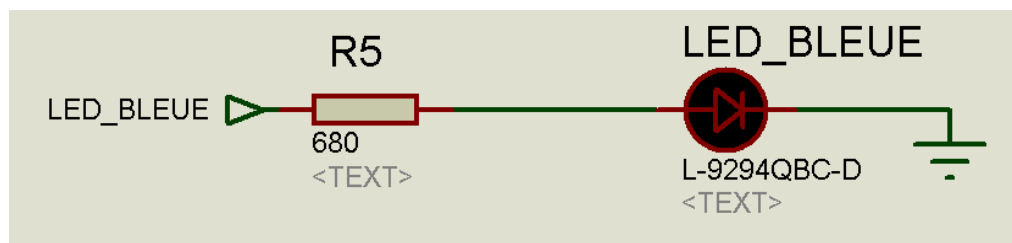


Figure 53 : Schéma électrique récepteur de la simulation de la partie action (connexion)

Préambule :

Après que nous avoir retenue la LED L-9294QBC-D de couleur bleu, nous avons aussi besoin d'une résistance de protection placée en série pour pouvoir réguler le courant qui la traverse. Pour cela on vas calculer la valeur de résistance convenable : à l'aide du CDC et la datasheet de la LED [L-9294QBC-D.pdf](#) et selon la Méthode de vidéo - <https://eqrcode.co/a/ZPtBe1> => BUT 1ère Année/Semestre 2/[Ressource n°7 - Comment faire briller une LED ?](#)

I/ intensité lumineuse :

L'intensité lumineuse du voyant est de 100 mCd (-/+20%), d'après le cahier des charges.

| | | |
|----------------------------------|---|-------|
| IUT Bordeaux Département GEii | Référence : TDB_DDC_EQ21 Révision : 3 – 20/02/2024 | 52/69 |
|----------------------------------|---|-------|

2/ intensité électrique :

Grâce à des informations trouvées dans la datasheet.

$$I_F = (100 \times 20) / 500 = 4 \text{ mA}$$

3/ tension électrique :

Grâce à ces valeurs de I_F nous avons pu trouver la tension V_F correspondante dans la datasheet

$$V_F = (4 \text{ mA} \times 2.8 \text{ V}) / 5 \text{ mA} = 2.24 \text{ V}$$

4/ résistance

$$R = (V_{cc} - V_f) / I_F$$

Avec $V_{cc} = 5 \text{ V}$

$$R = (5 - 2.24) / (4 \times 10^{-3})$$

$$R = 690 \Omega$$

| Nom | V_F (en V) | I_F (en mA) | Intensité lumineuse (en mCd) | R calculée (en Ω) | R normalisé (en Ω) série E12 ($\pm 10\%$) |
|-----------|--------------|---------------|------------------------------|---------------------------|---|
| LED Bleue | 2.24 | 4 | $100 \pm 20\%$ | 690 | 680 |

5/ montrer la conformité de la solution technique retenue par simulation à l'aide d'ISIS :

Nous avons calculé un courant I_F de 4 mA, d'autre part nous avons mesuré un courant I_F de 4.06 mA par la simulation et ces deux valeurs ont un écart de 1.5% ce qui est inférieur au 20% autorisé par le cahier des charges, donc on conclut que la résistance retenue $R = 680 \Omega \pm 10\%$; série E12 est conforme à l'exigence.

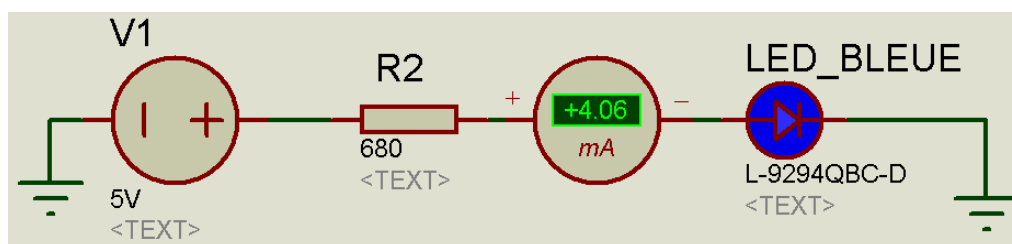


Figure 54 : Schéma électrique récepteur de la simulation de la partie action (connexion)

Référence du paragraphe : CDT_RCPT_KLAXON

Rédacteur : Bakas Laila et Larjuzan Noah

Relecteur : Lestage Margot et Cassany Marie

Exigences client vérifiées par pré-conception : EXIG_RCPT_KLAXON

Compétences GEII : C22/C24/C25/C26

Bloc fonctionnel étudié : ACTION

Pour préserver la durée de vie de notre buzzer nous voulons supprimer la composante continue du signal carré émis par le micro contrôleur.

Pour cela nous devons créer un filtre passe haut qui éliminerait les basse fréquence à l'aide d'un condensateur en série avec notre Buzzer.

Calculons alors la valeur de le condensateur qu'on doit utiliser :

$$\text{On a } f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 3000 \cdot C}$$

$$\text{Avec } R = \frac{V_{eff}}{I_{eff}} = \frac{6}{2 \cdot 10^{-3}} = 3k\Omega$$

- V_{eff} : d'après la datasheet de notre buzzer piézoélectrique [MCKPT-G1210-3916.pdf](#) on a $V_{pp} = 12V$, d'où $V_{eff} = 6V$
- $I_{eff} = 2 \cdot 10^{-3}$ d'après la datasheet du buzzer
- f_c la fréquence de coupure doit être inférieure à 4 khz

$$\text{donc } C > \frac{1}{2\pi \cdot 3000 \cdot 4000}$$

$$C > 13.3 \text{ nF}$$

Nous avons retenu une valeur de 100 nF pour le condensateur puisqu'il se trouve dans notre stock et est bien supérieure à 13.3 nF.

$$\text{Pour } C = 100 \text{ nF on a } f_c = \frac{1}{2\pi \cdot 3000 \cdot 100 \cdot 10^{-9}} = 530.51 \text{ hz} < 4 \text{ khz}$$

donc on peut conclure que le condensateur retenu $C = 100 \text{ nF} \pm 5\%$ est conforme à l'exigence.

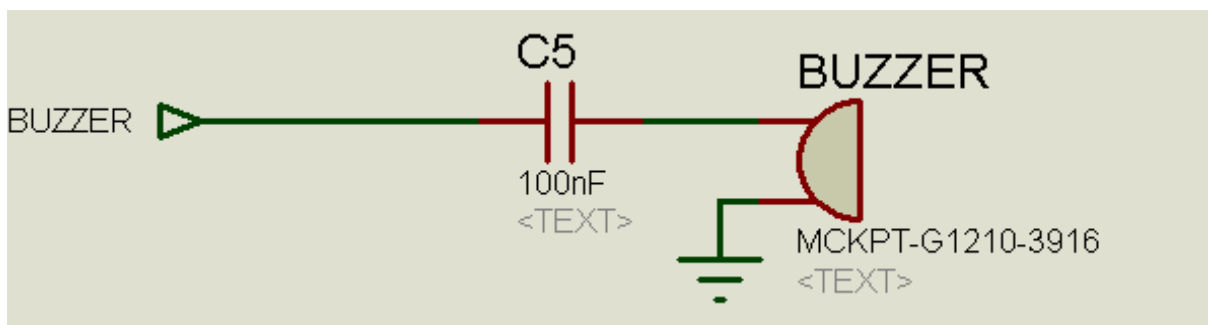


Figure 55 : Schéma électrique récepteur de la partie action (Klaxon)

3.2 Informatique

3.2.1 Informatique - Émetteur

Référence du paragraphe : CDT_EMTT_TRAITEMENT

Rédacteur : Lestage Margot , Cassany Marie , Noah Larjuzan et Bakas Laila

Relecteur : Lestage Margot , Cassany Marie , Noah Larjuzan et Bakas Laila

Exigences client vérifiées par pré-conception : EXIG_RCPT_TRAITEMENT

Compétences GEII : C22/C23/C24/C25/C26

Fichier : [EMTT](#)

Pour permettre l'acquisition des valeurs des potentiomètres et du bouton poussoir placés sur la carte trois fonctions ont été créées pour lire leurs valeurs.

La fonction AcquerirDirection() ne prend pas d'argument et retourne la valeur lue sur le potentiomètre de direction sous la forme d'un chiffre codé sur 16 bits qui aura une valeur comprise entre 0 et 1023.

```
// Lecture de la tension du potentiometre direction [0,1023]
uint16_t AcquerirDirection(void)
{
    int Delec = analogRead(PIN_POTAR_D);
    return Delec;
}
```

Figure 56 : Fonction AcquerirDirection()

La fonction AcquerirVitesse() ne prend pas d'argument et retourne la valeur lue sur le potentiomètre de vitesse sous la forme d'un chiffre codé sur 16 bits qui aura une valeur comprise entre 0 et 1023.

```
// Lecture de la tension du potentiometre vitesse[0,1023]
uint16_t AcquerirVitesse(void)
{
    int Velec;
    Velec = analogRead(PIN_POTAR_V);
    return Velec;
}
```

Figure 57 : Fonction AcquerirVitesse()

La fonction `AcquerirKlaxon()` ne prend pas d'argument et renvoie l'état inverse du bouton lié au klaxon car il est connecté en logique inverse. La valeur renvoyée est un 1 si le bouton est actif ou un 0 sinon. La valeur est codée sur 8 bits.

```
// Lecture de l'état logique du bouton du klaxon [0,1]
uint8_t AcquerirKlaxon(void)
{
    int klaxon = !digitalRead(PIN_KLAXON);
    return klaxon;
}
```

Figure 58 : Fonction AcquerirKlaxon()

La fonction `GenererTrameNEC()` prend pour paramètre la broche sur laquelle la trame va être émise, l'adresse NEC et la donnée à transmettre. La fonction `GenererTrameNEC()` ne renvoie rien et émet directement les impulsions conformément au protocole NEC pour qu'elles soient émises par un composant connecté à la broche spécifiée. Elle est comprise dans la bibliothèque NEC.

```
GenererTrameNEC(RECEPTEUR_INFRAROUGE_Pin, Adresse, Donnee);
```

Figure 59: Fonction GenererTrameNec()

La fonction `CalculerDonnee()` prend en paramètres les valeurs retournées par les fonctions `AcquerirVitesse()` et `AcquerirDirection()`.

Vitesse : Si l'on map de 0 à 1023 vers 0 à 15, la valeur 15 n'est atteinte que lorsque le potentiomètre atteint la valeur 1023. Pour régler ce problème, la valeur de vitesse est mappée de 0 à 1024 vers 0 à 16. Ainsi, la valeur 16 n'est atteinte que lorsque la valeur 1024 (valeur impossible) est atteinte par le potentiomètre.

Direction : Avec la même logique que pour la vitesse, on map de 0 à 1024 vers 0 à 15 afin que le nombre de valeurs soit impair pour que le kart aille tout droit.

Elle retourne un octet où les 4 bits de gauche contiennent la valeur de la vitesse et les 4 bits de droite contiennent la valeur de la direction.

```
// Retourne 8 bits qui comprennent la vitesse et la direction
uint8_t CalculerDonnee(uint16_t vitesse, uint16_t direct)
{
    // Vitesse sur les 4 bits de gauche, Direction sur les 4 bits de droite
    int DonneeVitesse = map(vitesse, 0, 1024, 0, 16);
    int DonneeDirection = map(direct, 0, 1024, 0, 15);
    int Donnee = DonneeVitesse << 4 | DonneeDirection;
    return Donnee;
}
```


Figure 60 : Fonction CalculerDonnee()

La fonction `Algorithme()` ne prend pas de paramètre et ne retourne rien. C'est elle qui est appelée en boucle dans `void loop()`.

Elle appelle les fonctions `AcquerirVitesse()`, `AcquerirDirection()`, `AcquerirKlaxon()`, `CalculerDonnee()` et `CalculerAdresse()`.

Elle vérifie ensuite si la tram NEC doit être généré à nouveau en appelant la fonction `OkToSend()`.

```
void Algorithme(void)
{
    Vit_potar = AcquerirVitesse();
    Dir_potar = AcquerirDirection();
    Klaxon = AcquerirKlaxon();
    Donnee = CalculerDonnee(Vit_potar, Dir_potar);
    Adresse = CalculerAdresse(Klaxon);
    if (OkToSend(Adresse, Donnee) == 1)
    {
        GenererTrameNEC(RECEPTEUR_INFRAROUGE_Pin, Adresse, Donnee);
    }
}
```

Figure 61 : Fonction Algorithme()

Référence du paragraphe : CDT_EMTT_REPETITIVITE

Rédacteur : Lestage Margot et Cassany Marie

Relecteur : Noah Larjuzan et Bakas Laila

Exigences client vérifiées par pré-conception : EXIG_RCPT_REPETITIVITE

Compétences GEII : C22/C23/C24/C25/C26

Pour permettre de répondre à l'exigence répétitivité du cahier des charges nous avons créé une fonction et des variables mémoires.

La fonction `OkToSend()`, prend en paramètre les valeurs de l'adresse et de la donnée et si un des paramètres a changé ou si le temps définie pour la période (333 ms) a été dépassé, la fonction retourne 1 et génère la nouvelle trame dans l'algorithme. Sinon la fonction retourne 0.

```
// Retourne 1 si le signal doit être envoyé, 0 sinon
bool OkToSend(uint8_t adresse, uint8_t donnee)
{
    // Test si les données, l'adresse(klaxon) ou le temps à dépasser 333ms
    if ((donnee != donneePrecedente) || (adresse != adressePrecedente) || (millis() >= (tempsPrecedent + 333)))
    {
        donneePrecedente = donnee;
        adressePrecedente = adresse;
        tempsPrecedent = millis();
        return 1;
    }
    else
    {
        return 0;
    }
}
```

Figure 62 : Fonction OkToSend()

```
if (OkToSend(Adresse, Donnee) == 1)
{
    GenererTrameNEC(RECEPTEUR_INFRAROUGE_Pin, Adresse, Donnee);
}
```

Figure 63 : Utilisation de la fonction OkToSend()

Référence du paragraphe : CDT_EMTT_RETENTISSEMENT**Rédacteur :** Lestage Margot et Cassany Marie**Relecteur :** Noah Larjuzan et Bakas Laila**Exigences client vérifiées par pré-conception :** EXIG_RCPT_RETENTISSEMENT**Compétences GEII :** C22/C23/C24/C25/C26

La fonction CalculerDonnee() prend en paramètres l'état du klaxon retourné par AcquerirKlaxon(). Elle retourne un octet où les 7 bits de droite contiennent le numéro d'équipe et les bit de gauche contiennent l'état du klaxon.

```
// Retourne 8 bits qui comprennent l'état du klaxon et le numéro de l'équipe
uint8_t CalculerAdresse(bool klaxon)
{
    // Klaxon sur le bit dominant, numéro de l'équipe sur les 7 autres
    Adresse = klaxon << 7 | NumeroEquipe;
    return Adresse;
}
```

Figure 64 : Fonction CalculerAdresse()

3.2.2 Informatique - Récepteur

Référence du paragraphe : CDT_RCPT_TRAITEMENT

Rédacteur : Frieu Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG_RCPT_TRAITEMENT

Compétences GEII : C22/C23/C24/C25/C26

Fichier : [RCPT_DETAILLE](#)

Pour répondre à l'exigence EXIG_RCPT_TRAITEMENT, nous avons écrit un programme informatique. Nous avons choisi de suivre les six points de l'exigence :

- * détection de l'absence de signal infrarouge en réception,
- * décodage des trames protocolaires NEC reçues,
- * vérification de la validité de la trame,
- * identification si l'adresse NEC est bien celle attendue,
- * extraction de l'information de puissance moteur transmise sur 4 bits et l'information de direction de roue sur 4 bits,
- * calcul de la puissance à appliquer sur le moteur et l'angle de direction à appliquer aux roues avant.

* Détection de l'absence de signal infrarouge en réception.

Dans notre diagramme d'architecture, nous avons dit que l'absence de signal infrarouge (ici trame NEC) renvoyait la valeur -2 dans la variable Erreur.

Ainsi, notre choix a été de faire prendre à Erreur la valeur de AcquerirTrameNEC afin de vérifier sa valeur. De cette manière, via un test "si", nous vérifions si la trame est présente ou non. Si la trame est absente, les variables associées à la commande de la LED bleue et du moteur reçoivent un 0 logique. La LED bleue et le moteur sont éteints.

* décodage des trames protocolaires NEC reçues

Pour cette partie, la fonction AcquerirTrameNEC() de la librairie "NEC.h" décode les variations de tensions en nombre binaire de 8 bits chacun pour les variables Adresse et Donnee.

* vérification de la validité de la trame

Pour cette partie, nous utilisons la fonction `AcquerirTrameNEC()` de la librairie “NEC.h”. Lorsque la trame NEC reçue n’est pas valide, cette fonction nous retourne la valeur -1 que nous mettons dans la variable `Erreur`. Puis, nous effectuons un test Si pour valider ou non la trame NEC. Si le test (`Erreur == 0`) n’est pas valide, nous éteignons la LED bleue et le moteur brushless avec deux fonctions spécifiques.

```
//__Boucle_Principale_Du_Programme__

void loop() {
    uint8_t Adresse;
    uint8_t Donnee;
    int8_t Erreur;
    int8_t Klaxon;
    uint8_t numEQ;

    Erreur = AcquerirTrameNEC(RECEPTEUR_INFRAROUGE_PIN, &Adresse, &Donnee);
    // Acquisition de la trame NEC à partir du récepteur infrarouge

    if (Erreur == 0) {
        // (V1) Vérification de la bonne réception de la trame NEC
        numEQ = ExtraireNoEQ(Adresse);
        // Extraction du numéro d'équipe dans l'adresse

        if (numEQ == EQUIPE_NUMBER) {
            // (V2) Vérification du numéro d'équipe
            PiloterLedBleue(ON);
            // Allumage de la LED bleue
            PiloterMoteur(ExtraireVitesse(Donnee));
            // Pilotage du moteur en fonction de la vitesse extraite de la donnée
            PiloterServoMoteur(ExtraireDirection(Donnee));
            // Pilotage du servo-moteur en fonction de la direction extraite de la donnée
            PiloterKlaxon(ExtraireKlaxon(Adresse));
            // Pilotage du klaxon en fonction de l'état extrait de l'adresse
        }
        else {
            // (V2) Si numéro d'équipe ne correspond pas dans l'adresse
            PiloterLedBleue(OFF);
            // Extinction de la LED bleue
            PiloterMoteur(OFF);
            // Arrêt du moteur
        }
    }
    else {
        // (V1) Si erreur est détectée dans l'acquisition de la trame
        PiloterLedBleue(OFF);
        // Extinction de la LED bleue
        PiloterMoteur(OFF);
        // Arrêt du moteur
    }
}
```

Figure 65 : Lignes de code correspondantes à la vérification de la validité de la trame NEC

* identification si l'adresse NEC est bien celle attendue

Dans le cas de l’étape d’identification, nous avons validé l’acquisition de la trame NEC (`Erreur=0`). Nous avons par la suite pris la valeur retournée par la fonction `ExtraireNoEQ(Adresse)` via la variable `numEQ`. Si cette valeur correspond à notre numéro d’équipe (0x21), nous allumons la LED bleue et nous appelons les fonctions suivantes `PiloterMoteur()`, `PiloterServoMoteur()` et `PiloterKlaxon()` afin de prendre les valeurs respectives suivantes `ExtraireVitesse(Donnee)`,

ExtraireDirection(Donnee) et ExtraireKlaxon(Adresse). Et si le test n'est pas bon, nous éteignons la LED bleue et le moteur brushless.

*extraction de l'information de puissance moteur transmise sur 4 bits et l'information de direction de roue sur 4 bits

Pour cette partie, nous avons suivi le diagramme d'architecture informatique. Nous avons choisi de créer quatre fonctions : ExtraireVitesse(), ExtraireDirection(), ExtraireKlaxon() et ExtraireNoEQ(). Elles permettent d'extraire des variables Adresse et Donnee les informations suivantes : la vitesse (variable = vitesse), la direction (variable = Direction), l'état du klaxon (variable = klaxon) et le numéro d'équipe (variable = numEQ). Ces informations sont utilisées plus loin dans le programme.

```
// Fonction qui extrait la vitesse à partir de la donnée
uint8_t ExtraireVitesse(uint8_t Donnee) {
    uint8_t vitesse = Donnee >> 4;
    // Décalage de 4 bits vers la droite pour obtenir les 4 bits de poids fort représentant la vitesse
    return vitesse;
}

// Fonction qui extrait la direction à partir de la donnée
uint8_t ExtraireDirection(uint8_t Donnee) {
    uint8_t Direction = Donnee & 0b00001111;
    // Masquage pour extraire les 4 bits de poids faible représentant la direction
    return Direction;
}

// Fonction qui extrait l'état du klaxon à partir de l'adresse
bool ExtraireKlaxon(uint8_t Adresse) {
    bool klaxon = Adresse >> 7;
    // Décalage de 7 bits vers la droite pour obtenir le bit de l'état du klaxon seule
    return klaxon;
}

// Fonction qui extrait le numéro d'équipement à partir de l'adresse
uint8_t ExtraireNoEQ(uint8_t Adresse) {
    uint8_t numEQ = Adresse & 0b01111111;
    // Masquage pour extraire les 7 bits de poids faible représentant le numéro d'équipement.
    return numEQ;
}
```

Figure 66 : Lignes de code correspondantes au décodage des trames NEC

* calcul de la puissance à appliquer sur le moteur et l'angle de direction à appliquer aux roues avants

Pour cette partie, nous avons choisi de créer deux fonctions qui calculent la puissance et l'angle et les appliquent sur le moteur brushless et le servo moteur. Ces fonctions ont en paramètre les variables Direction et Vitesse. Elles réalisent un map (Rapport entre de plage de donnée). Pour la

direction, ce rapport se fait de 0 à 14 pour avoir un milieu pile à 7 et avoir les roues droites. Puis, ces rapports sont directement envoyés à leurs moteurs respectifs via la fonction write().

```
// Fonction qui pilote le servo-moteur en fonction de la direction
void PiloterServoMoteur(uint8_t Direction) {
    Servomotor.write(map(Direction, 0, 14, 0, 180));
    // Mapping de la direction (entre 0 et 14) sur l'intervalle de 0 à 180 degrés pour le servo-moteur
}

// Fonction qui pilote le moteur en fonction de la vitesse
void PiloterMoteur(uint8_t Vitesse) {
    Motorbrushless.write(map(Vitesse, 0, 15, 0, 180));
    // Mapping de la vitesse (entre 0 et 15) sur l'intervalle de 0 à 180 degrés pour le moteur brushless
}
```

Figure 67 : Lignes de code correspondantes aux calculs de puissance et d'angle

Référence du paragraphe : CDT_RCPT_SECURITE

Rédacteur : Friei Mathieu et L'Haridon Mathis

Relecteur : Jerez Elouan et Roboam Damien

Exigences client vérifiées par pré-conception : EXIG_RCPT_SECURITE

Compétences GEII : C22/C23/C24/C25/C26

Pour répondre à cette exigence, nous avons mis en place deux tests “si” (figure 68) qui vérifient la validité du signal reçu. Dans le cas où il n’y a pas de signal, la fonction PiloterMoteur() renvoie 0.

Dans le cas où la réception du signal est invalide, là encore la fonction PiloterMoteur() renvoie 0.

Notre solution est conforme à l'exigence.

Référence du paragraphe : CDT_RCPT_RETENTISSEMENT

```

Erreur = AcquerirTrameNEC(RECEPTEUR_INFRAROUGE_PIN, &Adresse, &Donnee);
// Acquisition de la trame NEC à partir du récepteur infrarouge

if (Erreur == 0) {
    // (V1) Vérification de la bonne réception de la trame NEC
    numEQ = ExtraireNoEQ(Adresse);
    // Extraction du numéro d'équipe dans l'adresse

    if (numEQ == EQUIPE_NUMBER) {
        // (V2) Vérification du numéro d'équipe
        PiloterLedBleue(ON);
        // Allumage de la LED bleue
        PiloterMoteur(ExtraireVitesse(Donnee));
        // Pilotage du moteur en fonction de la vitesse extraite de la donnée
        PiloterServoMoteur(ExtraireDirection(Donnee));
        // Pilotage du servo-moteur en fonction de la direction extraite de la donnée
        PiloterKlaxon(ExtraireKlaxon(Adresse));
        // Pilotage du klaxon en fonction de l'état extrait de l'adresse
    }
    else {
        // (V2) Si numéro d'équipe ne correspond pas dans l'adresse
        PiloterLedBleue(OFF);
        // Extinction de la LED bleue
        PiloterMoteur(OFF);
        // Arrêt du moteur
    }
}
else {
    // (V1) Si erreur est détectée dans l'acquisition de la trame
    PiloterLedBleue(OFF);
    // Extinction de la LED bleue
    PiloterMoteur(OFF);
    // Arrêt du moteur
}
}

```

Figure 68 : Lignes de code correspondantes à la sécurité du programme

Référence du paragraphe : CDT_RCPT_RETENTISSEMENT

Rédacteur : Jerez Elouan et Roboam Damien

Relecteur : Friehe Mathieu et L'Haridon Mathis

Exigences client vérifiées par pré-conception : EXIG_RCPT_RETENTISSEMENT

Compétences GEII : C1-10, C1-11

```

// Fonction qui extrait l'état du klaxon à partir de l'adresse
bool ExtraireKlaxon(uint8_t Adresse) {
    bool klaxon = Adresse >> 7; // Décalage de 7 bits vers la droite pour obtenir le bit de l'état du klaxon seule
    return klaxon;
}

```

Figure 69 : Lignes de code correspondantes à la fonction ExtraireKlaxon()

```
// Fonction qui pilote le klaxon en fonction de l'état
void PiloterKlaxon(uint8_t etat) {
    if (etat == ON) {
        tone(KLAXON_PIN, FREQUENCE); // Activation du klaxon à la fréquence spécifiée si l'état reçu est ON = 1
    }
    else {
        noTone(KLAXON_PIN);          // Désactivation du klaxon
    }
}
}
```

Figure 70 : Lignes de code correspondantes à la fonction PiloterKlaxon

```
if (numEQ == EQUIPE_NUMBER) { // (V2) Vérification du numéro d'équipe
    PiloterLedBleue(ON);        // Allumage de la LED bleue
    PiloterMoteur(ExtraireVitesse(Donnee)); // Pilotage du moteur en fonction de la vitesse extraite de la donnée
    PiloterServoMoteur(ExtraireDirection(Donnee)); // Pilotage du servo-moteur en fonction de la direction extraite de la donnée
    PiloterKlaxon(ExtraireKlaxon(Adresse)); // Pilotage du klaxon en fonction de l'état extrait de l'adresse
}
```

Figure 71 : Lignes de code correspondantes au pilotage du klaxon

Étape 1 : figure 69

Identifier si le klaxon doit ou non retentir, pour cela on utilise le code : “ExtraireKlaxon”.

Ce code permet de se décaler sur l’octet de la lecture de la transmission NEC vers la droite pour isoler le bit d’information klaxon.

Étape 2 : figure 70

Actionner le klaxon. Pour cela on utilise la fonction “PiloterKlaxon” qui utilise le résultat de l’étape 1 pour appliquer ou non l’instruction TONE qui fait retentir le buzzer. On applique à tone une variable (fréquence) qui applique les 4 KHz demandés dans le cahier des charges.

Étape 3 : figure 71

Faire appel à ces deux fonctions dans le programme principal du loop qui va donc le répéter à l’infini. PiloterKlaxon doit utiliser ExtraireKlaxon pour faire l’action.

Fichier : [KAH PDP 2024.xlsx](#)

Sur le planning prévisionnel, les cases de couleurs pâles représentent les séances déjà passées. Nous sommes donc dans les temps.

[illegible]

Figure 73 : Planning de la conception détaillé

3.4 Conclusion de la conception détaillée du produit

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : Noah Larjuzan

La conception détaillée a permis d'établir deux schémas électriques et deux programmes complets comprenant toutes les exigences du cahier des charges. Nous n'avons rencontré aucun problème.

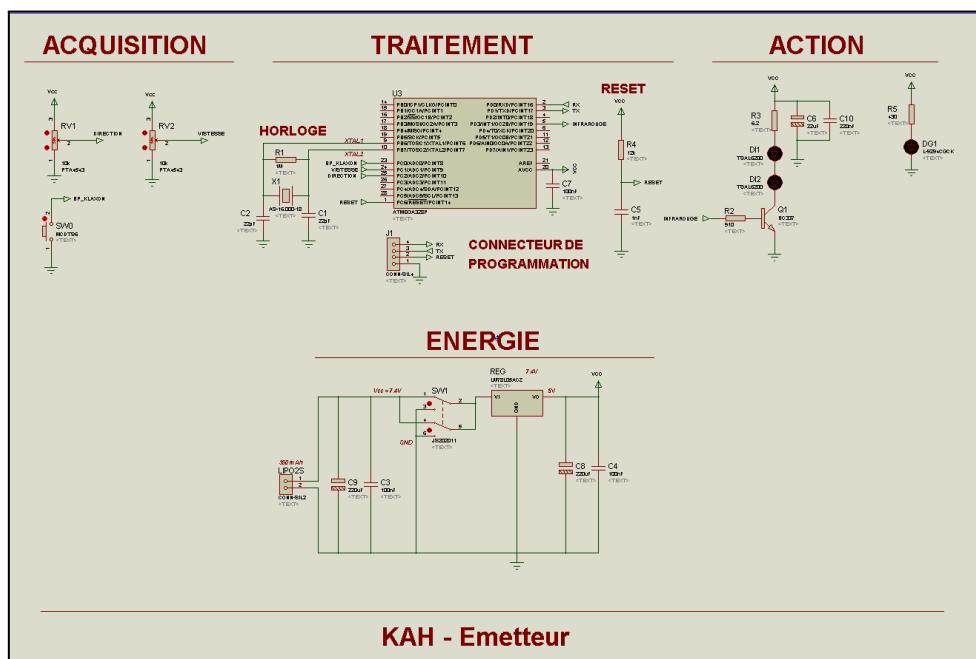


Figure 74 : Schéma de l'émetteur

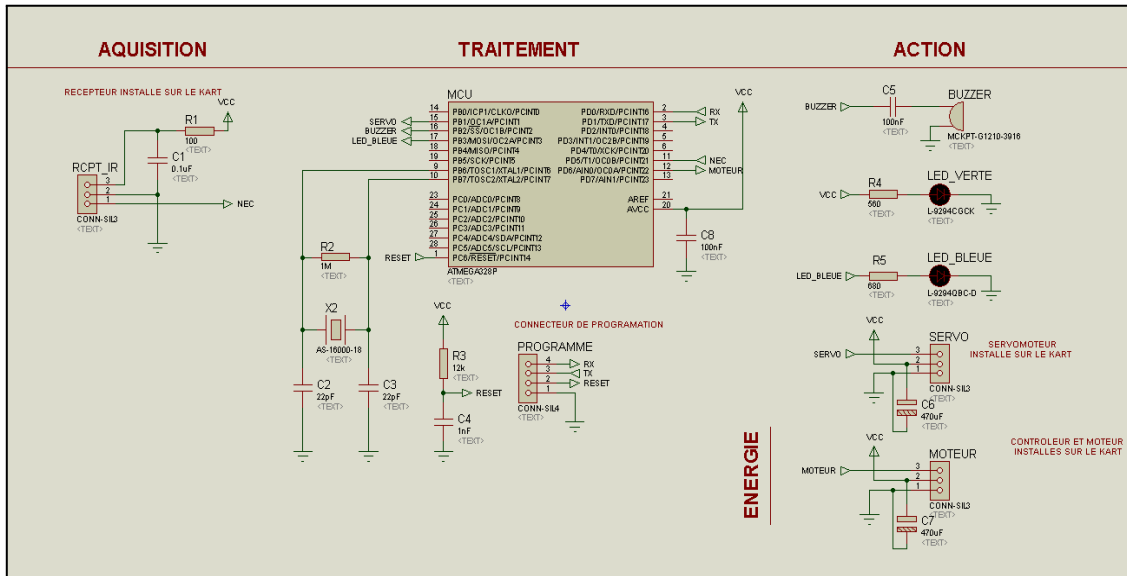


Figure 75 : Schéma du récepteur

4. Conclusion de la conception du produit

Rédacteur : Frieih Mathieu et L'Haridon Mathis

Relecteur : Lestage Margot et Cassany Marie

Nous n'avons relevé aucunes non-conformités. Cela signifie donc que toutes les exigences ont été vérifiées. Ainsi, nous avons pour but de passer à la fabrication de notre produit.

5. Matrice de conformité du produit

Ce chapitre synthétise par l'intermédiaire d'un tableau la conformité du produit développé par rapport aux exigences issues du Cahier des Charges.

| Exigence | Méthodes de développement | Paragraphes en lien avec l'exigence | Statut |
|------------------------|---|--|----------------------|
| EXIG_EMTT_DIMENSIONS | Conception préliminaire | CPR_EMTT_DIMENSIONS | Conforme |
| EXIG_EMTT_LOGO | Conception préliminaire | CPR_EMTT_LOGO | Conforme |
| EXIG_EMTT_ENERGIE | Conception préliminaire Conception détaillée | CPR_EMTT_ENERGIE CDT_EMTT_ENERGIE | Conforme Conforme |
| EXIG_EMTT_INTERRUPTEUR | Conception préliminaire Conception détaillée | CPR_EMTT_INTERRUPTEUR CDT_EMTT_INTERRUPTEUR | Conforme Conforme |

| Exigence | Méthodes de développement | Paragraphes en lien avec l'exigence | Statut |
|--------------------------|---|--|----------------------|
| EXIG_EMTT_IHM | Conception préliminaire Conception détaillée | CPR_EMTT_IHM CDT_EMTT_IHM | Conforme Conforme |
| EXIG_EMTT_KLAXON | Conception préliminaire Conception détaillée | CPR_EMTT_KLAXON CDT_EMTT_KLAXON | Conforme Conforme |
| EXIG_EMTT_TRAITEMENT | Conception préliminaire Conception détaillée | CPR_EMTT_TRAITEMENT CDT_EMTT_TRAITEMENT | Conforme Conforme |
| EXIG_EMTT_REPETITIVITE | Conception préliminaire Conception détaillée | CPR_EMTT_REPETITIVITE CDT_EMTT_REPETITIVITE | Conforme Conforme |
| EXIG_EMTT_RETENTISSEMENT | Conception préliminaire Conception détaillée | CPR_EMTT_RETENTISSEMENT CDT_EMTT_RETENTISSEMENT | Conforme Conforme |
| EXIG_EMTT_PUISSANCE | Conception préliminaire Conception détaillée | CPR_EMTT_PUISSANCE CDT_EMTT_PUISSANCE | Conforme Conforme |
| EXIG_EMTT_INDICATEUR | Conception préliminaire Conception détaillée | CPR_EMTT_INDICATEUR CDT_EMTT_INDICATEUR | Conforme Conforme |
| EXIG_RCPT_DIMENSIONS | Conception préliminaire | CPR_RCPT_DIMENSIONS | Conforme |
| EXIG_RCPT_LOGO | Conception préliminaire | CPR_RCPT_LOGO | Conforme |
| EXIG_RCPT_ENERGIE | Conception préliminaire Conception détaillée | CPR_RCPT_ENERGIE CDT_RCPT_ENERGIE | Conforme Conforme |
| EXIG_RCPT_INTERRUPTEUR | Conception préliminaire Conception détaillée | CPR_RCPT_INTERRUPTEUR CDT_RCPT_INTERRUPTEUR | Conforme Conforme |
| EXIG_RCPT_CAPTEUR | Conception préliminaire Conception détaillée | CPR_RCPT_CAPTEUR CDT_RCPT_CAPTEUR | Conforme Conforme |
| EXIG_RCPT_TRAITEMENT | Conception préliminaire Conception détaillée | CPR_RCPT_TRAITEMENT CDT_RCPT_TRAITEMENT | Conforme Conforme |
| EXIG_RCPT_SECURITE | Conception préliminaire Conception détaillée | CPR_RCPT_SECURITE CDT_RCPT_SECURITE | Conforme Conforme |
| EXIG_RCPT_RETENTISSEMENT | Conception préliminaire Conception détaillée | CPR_RCPT_RETENTISSEMENT CDT_RCPT_RETENTISSEMENT | Conforme Conforme |
| EXIG_RCPT_MOTEUR | Conception préliminaire Conception détaillée | CPR_RCPT_MOTEUR CDT_RCPT_MOTEUR | Conforme Conforme |

| Exigence | Méthodes de développement | Paragraphes en lien avec l'exigence | Statut |
|----------------------|---|--|----------------------|
| EXIG_RCPT_ROUE | Conception préliminaire Conception détaillée | CPR_RCPT_ROUE CDT_RCPT_ROUE | Conforme Conforme |
| EXIG_RCPT_INDICATEUR | Conception préliminaire Conception détaillée | CPR_RCPT_INDICATEUR CDT_RCPT_INDICATEUR | Conforme Conforme |
| EXIG_RCPT_CONNEXION | Conception préliminaire Conception détaillée | CPR_RCPT_CONNEXION CDT_RCPT_CONNEXION | Conforme Conforme |
| EXIG_RCPT_KLAXON | Conception préliminaire Conception détaillée | CPR_RCPT_KLAXON CDT_RCPT_KLAXON | Conforme Conforme |
| EXIG_DELAI | Conception préliminaire Conception détaillée | CPR_DELAI CDT_DELAI | Conforme Conforme |
| EXIG_COUT | Conception préliminaire Conception détaillée | CPR_COUT CDT_COUT | Conforme Conforme |