Document Technique

Automatisation poulailler

# Lycée Polyvalent Jean Zay

JARNY 54800

BTS Système Numérique option A Informatique et Réseaux

Epreuve E-62 Projet Technique

Nom du projet :

Automatisation d’un poulailler

Porteur du projet :

MARCHE. Frédéric

Information

| Nom du projet | Automatisation d’un poulailler |
| --- | --- |
| Type de document | Document technique |
| Date | 03/03/2022 |
| Version | 1.0 |
| Auteurs | * Ryan Dhoker * Dilan Da Fonseca * Julien Barthoulot |

Rédactions et Modification

| Version | Date | Auteur | Modification |
| --- | --- | --- | --- |
| 1.0 | 03/03/2022 | Ryan Dhoker | * Premier version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sommaire

# Résumé du document

# Présentation

* 1. Demande initiale
  2. Contraintes
  3. Réalisation
     1. Répartition des tâche
     2. Diagramme de structure
     3. Diagramme de déploiement Bloc de commande
     4. Diagramme de déploiement Central
     5. Composants
        1. Bloc de command
        2. Centrale

# Centrale

* 1. Présentation
  2. Réalisation du la raspberry pi

Résumé du document

Ce document est la documentation technique du projet Poulailler Automatisé pour les BTS Système Numérique option A Informatique et Réseaux , de 2022. il sera divisé en plusieur parties

1. Présentation du projet
2. Centrale
   1. Présentation
   2. Réalisation du la raspberry pi
   3. Code informatique
3. Bloc de commande
   1. Présentation
   2. Réalisation du la raspberrypi
   3. Code informatique
4. Résumé

Présentation

# D'amande initial

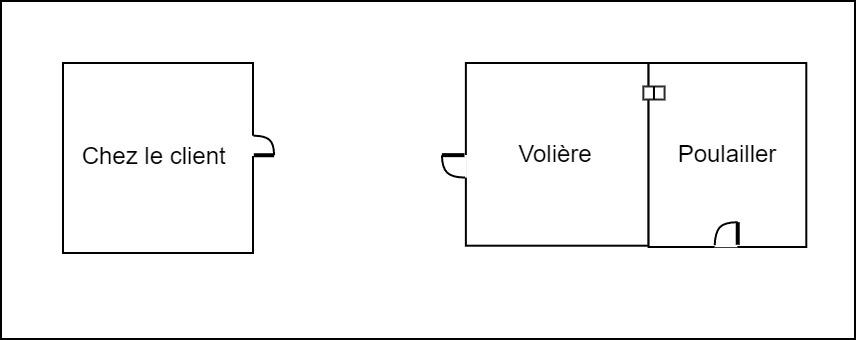
Nous avons été chargés de l'automatisation du poulailler déjà existant, dans le but de réaliser l'infrastructure Hardware et software du site, chez un client.

Le site est composé des 4 parties :

* Chez le client
* La Voliere
* Poulailler
* L'extérieur

Les spécificités du site :

* La volière est composée de mur grillagé et d’un toit en tôle.
* Le poulailler est un bâtiment en dure
* Porte non automatisé déjà présent entre le poulailler et la Voliere



Il nous a été demandé de réaliser un système de gestion dès l’Ouverture/ Fermeture des portes pour pouvoir relâcher les poules dans la volière et à l'extérieur, tout en contrôlant la présence des poules. l’ouverture et la fermeture seront contrôlées par la couche et le lever du soleil avec un horloge interne, avec possibilité des modifications par le client.

Les poules auront un baguer RFID pour détecter leur présence et un tunnel avec un porte automatique sera installé entre l'extérieur et la volière. Des lecteurs RFID seront installés à chaque entrée du tunnel. La porte entre la volière et le poulailler devra être automatisée. Un système manuelle d’Ouverture / Fermeture sera aussi implémenté en cas de mauvais fonctionnement des portes. Il aura un cher le client et dans le poulailler

Il a été aussi demandé de réaliser un système pour effrayer des potentiels prédateurs qui essayaient de pénétrer dans la volière pendant la nuit. Il a été opté d'utiliser un système lumineux pour éviter le plus possible les nuisances dues au système. Un capteur de présence sera là pour détecter si il aura un intrusion dans la volière. il sera aussi utilisé pour voir si tout les poule seront bien rentrer dans le poulailler

Pour contrôler le tout, il aura un central avec UI, chez le client qui sera chargé d'envoyer/Récupérer des informations d’un bloc de commande qui sera chargé d'automatiser le poulailler.

Dans la central , il a été demandé de réaliser un interface utilisateur pour gérer le poulailler et un système configuration des bague RFID

# Contraintes

Les contrainte de réalisation :

* Contrainte de Financement

Le système n'étant pas destiné à une exploitation professionnelle et ne devant pas être répliqué de manière industrielle , le prix devra être raisonnable tout en assurant une qualité suffisante pour une exploitation durable.

* Contrainte de développement (Hardware et software)

Dans le bloc de commande,les différents programmes présents dans la raspberry seront écrits en python

Dans la centrale placée chez le client , comme environnement de développement pour construire les application, le logiciel Node-red est fortement conseillé

* Contrainte qualité

Les composent seront pérennes et pourront facilement être remplacé par des produit identique ou fonctionnellement équivalents

# Réalisation

# Répartition des tâche

# 

Ryan Dhoker / Développeur principale

Réalisation complète de la centrale, installation et configuration de la raspberry pi, configuration de base de données, programmation de l'interface utilisateur, réalisation des fonctionnalités, réalisation de la communication MQTT. Sur le bloc de commande, réalisation du programme principale, réalisation du la communication.

Julien Barthoulot / Électronicien et développeur secondaire

Réalisation des circuits électroniques permettant l’utilisation des différents composants. programmation des différentes fonctionnalités sur la raspberry pi du bloc de commande pour les composants tels que les vérins/motoréducteurs, les lecteur RFID, capteur infrarouge ainsi que les calculs des éphémérides (coucher/lever de soleil).

Dylan Da Fonseca / Informaticiens et développeur secondaire

Réalisation et configuration de la raspberry pi du bloc de commande, configuration du la base de donnée, programmation des différentes fonctionnalités sur la raspberry pi du bloc de commande.

# 

# 

# Diagramme de structure

# 

# 

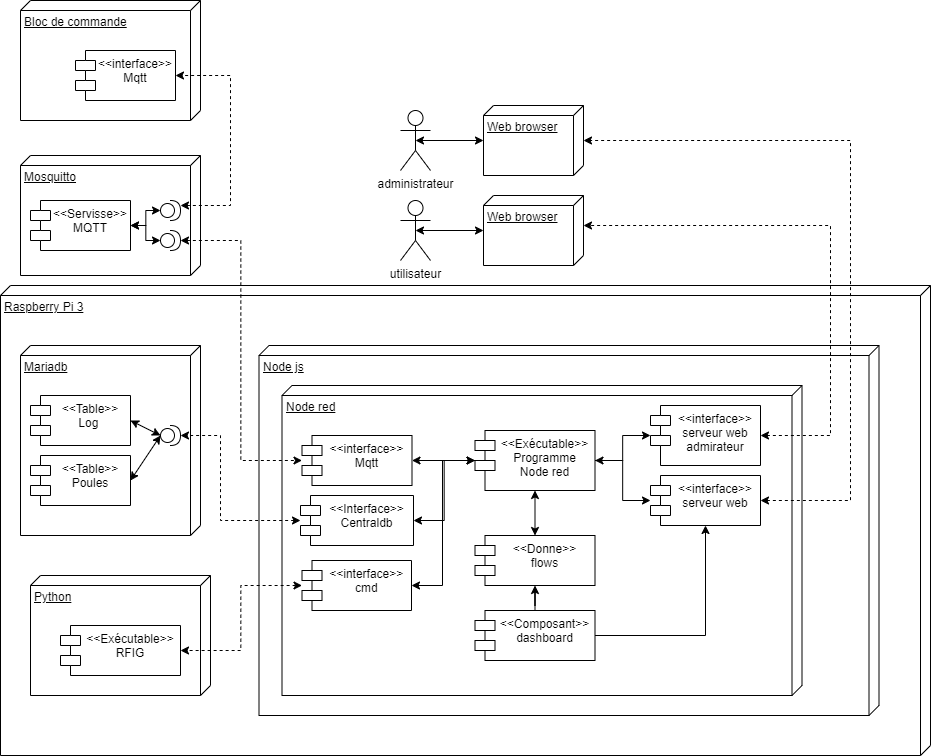
# Diagramme de cas d’utilisation

# 

# 

# Diagramme de déploiement Bloc de commande :

# Diagramme de déploiement Centrale :



# Composants

| Composant | Caractéristique |
| --- | --- |
|  |  |
| Bloc de commande | |
| Raspberry PI 3 | Elle dispose de :   * 4 port USB 2 * 1 port RJ-45 * 1 port HDMI |
| Module horloge temps réel | Bus I2C |
| Contrôleur de moteur Regénératif Sabertooth | 2 \* 5A , 6V-24V  Elle permet de piloter les moteur vérin |
| Vérin électrique | Courant continu |
| Module convertisseur analogique numérique | CAN 16 bit  Bus I2C |
| Capteur infrarouge | 3cm - 80cm  Elle permet de détecter la présence d'une poule dans la porte |
| Capteur de fin de course |  |
| Capteur de mouvement | Elle permet de détecter la présence une poule au autre dans la volière |
| Bloc de lumière LED | Groupe de LED 12V |
| Bouton poussoir | [None] |
| Module GSM | [None] |
| Servomoteur | il sera associé à un ensemble de pignon cremaillere pour la fermeture |
| Lecteur de badge RFID | [None] |
|  |  |
| Centrale | |
| Raspberry PI 3 | Elle dispose de :   * 4 port USB 2 * 1 port RJ-45 * 1 port HDMI |
| Ecran 7 pouces WaveShare | Ecran tactile HDMI |
| Bouton poussoir | [None] |
| Lecteur RFID | Permet de lire la bague |

Central

# Présentation

Pour pouvoir commander le poulailler à distance, chez le client. Il a été adopté d'utiliser un raspberry pi 3 avec Node-red comme environnement de développement, pour réaliser les différentes fonctionnalités et l'interface utilisateur. Python sera utilisé pour le lecteur RFID. Pour la base de données, ce sera MariaDB et MQTT pour la communication avec le bloc de commande.

L'ensemble de la centrale sera composée d’une raspberry pi avec un écran 7 pouces de WaveShare, de 2 boutons poussoir pour l’ouverture/Fermeture forcée des portes, d'un coffret et d’un clavier AZERTY. il sera relié à une connexion internet.

Les chapitres suivants vont représenter l'installation à neuf de la centrale, pas à pas.

# Réalisation du la raspberry pi

Installation de la raspberry pi

Pour commencer ils vous faudra les materielle suivant.

* Une rasberry Pi 3 avec alimentation
* Sourie et Clavier
* Un cable RJ45 avec un connection internet
* Un écran tactile avec ces cable
* un carte SD avec un adaptateur USB

Tout d'abord il faudra télécharger l’OS et un logisielle sur les site <https://www.raspberrypi.com/software/>. télécharger le logicielle “ Raspberry Pi Imager ” et ensuite en dessou, cliquez sur le bouton “see all download option “ qui vous donnera les possibilité de télécharger différente Pi OS

Dans la catégorie **RASPBERRY PI OS**, téléchargez **RASPBERRY PI OS WITH DESKTOP.**

Quant “Raspberry Pi Imager” est téléchargé ,installé le, puis attendez que l’OS se télécharge.

Insérez l’adaptateur avec la carte SD et ouvrez l’explorateur de fichier pour voir quelle lettre définit la carte SD. Lanche “Raspberry Pi Imager” , il vous demandera 2 choses, la première est un image ISO ( le fichier qui contient l’OS ) , la deuxième est l'emplacement de la carte SD. Quand vous avez donné ce 2 paramètre vous pouvez lancer l'écriture sur la carte SD , ça peut prendre un peu de temps .

Pendant ce temps vous pouvait branche les different cable sur la raspberry pi, **SAUFE L'ALIMENTATION**

Quand l'écriture et la vérification finir, retirer la carte SD et insérez la dans la raspberry pi et mettez la l’alimentation. Après un certain temps l’a raspberry pi va se lâche et vous pourrez voir sur l'écran le bureaux de l’utilisateur Pi.

Mise à jour de la raspberry pi

Fait les commandes suivent et attend que ça s'installe, il permet de mettre la raspberry pi à jour. **Ca peut prendre un petit heur**

**sudo apt update**

**sudo apt upgrade**

Puis pour un raisson de sécurité nous alon change le mot de passe de l’utilisateur Pi

**sudo raspi-config**

Dans “System Options” et dans “S3 Password”, fait entre.

Attention à bien noter le mots de passe**.**

installation et configuration de Node-red

Cettes commande permet d'installer node-red avec ce depandance

**sudo apt install build-essential git curl**

**bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered )**

Pour accéder à l'environnement de développement node-red, faites dans votre navigateur

Localhost:1880 ou [IP]:1880. A savoir de node-red ce lanche lentement au démarrage de la raspberry pi

Après l’installa fin, fait la commande suivante pour que Node-red se lance automatique

**sudo systemctl enable nodered.service**

Par la suite, il faudra sécuriser node-red, faire la commande suivante. Il vous demandera le mot de passe qui sera attribué à l'admin. il vous enverra ce mot de passe en hash, récupère le et stocke le pour la suite .

**ATTENTION A BIEN NOTÉ LE MOT DE PASSE !**

**sudo node-red admin hash-pw**

Pour attribuer le mot de passe a l’admin, il faudra modifier un fichier. En même temps, nous allons réaliser d’autre manipe dans ce fichier. Pour accéder à ce fichier de configuration faite la commande suivante

**sudo cd ~**

**sudo cd .node-red**

**sudo nano settings.js**

Ensuite vous allez cherche le code suivant qui sera commenté.

**ATTENTION CE FICHIER EST TRÈS VOLUMINEUX EN CODE, ALOR PRENEZ VOTRE TEMPS POUR CHERCHER !**

**# adminAuth : [**

**# type : “ credentials”,**

**# users : [ {**

**# username ; “admin”,**

**# password : “$2a$08$......”**

**# permission “\*”**

**# ]}**

**#],**

Quand vous avez trouvé ce code décommenter le et dans la parti “password” mettait le mot de passe hash que vous avait mis de côté

En suit vous allez chercher ce code qui est pas commente

**editorTheme : {**

**codeEditor : {**

**lib :”ace”**

**}**

**}**

Modifié le paramètre a l'intérieur de “lib” par “monaco”. Ça permettra de modifier l'éditeur de code de Node-red. Ça vous aidera à réaliser des blocs fonctions à l'intérieur de Node-red

installation et configuration de mariadb

Pour installer mariadb :

**sudo apt install mariadb-server**

Ensuite nous alon définir le mot de passe pour l’utilisateur root

**ATTENTION A BIEN NOTÉ LE MOT DE PASSE !**

**sudo mysql\_secure\_installation**

Appuyez sur entrer pour continuer (vous n’avez pas besoin de mot de passe par défaut)

Tapez ensuite « Y » pour définir un nouveau mot de passe, et entrez le mot de passe de votre choix.

Maintenant, appuyez trois fois sur « Y » pour

Supprimer les utilisateurs anonymes

Interdire la connexion à distance à la racine

Supprimer la base de données des tests

Et enfin, appuyez à nouveau sur « Y » pour recharger les privilèges

C’est tout, cette fois MariaDB est prêt à être utilisé avec pour connexion root.

Puis faites cette commande pour se connecter , il vous demandera le mot de passe.

**mysql -u root -p**

Nous alon créer les base de donnée et un utilisateur pour celle-ci:

**CREATE DATABASE centraldb ;**

Puis les table à l'intérieur

**CREATE TABLE poules (**

**IDdata int PRIMARY KEY AUTO\_INCREMENT**

**ID int NOT NULL UNIQUE**

**nom VARCHAR(100)DEFAULT NULL**

**nombre int DEFAULT NULL**

**date DATETIME DEFAULT CURRENT\_DATETIME**

**info TEXT DEFAULT NULL**

**);**

**CREATE TABLE log (**

**ID int PRIMARY KEY AUTO\_INCREMENT**

**date DATETIME DEFAULT CURRENT\_DATETIME**

**info TEXT DEFAULT NULL**

**)**

Pour l’utilisateur :

**CREATE USER 'centraldb'@'localhost' IDENTIFIED BY '[password]';**

Après ça, nous alon donne les droit à ce nouveau utilisateur sur les base de donnée

**GRANT ALL PRIVILEGES ON [base de donnée].\* TO '[username]'@'localhost';**

Recharger les privilèges.

**FLUSH PRIVILEGES;**

Maintenant vous pouvons quitte mariadb

installation et configuration de python

Par défaut, Python 3 est installé dans la raspberry pi. python nous servira à récupérer une trame RFID est sera récupéré par Node red.

**sudo pip3 install rdm6300**

il nous faudra cette dépendance pour lancer le programme qui sera détaillé plus tard

Configuration du navigateur en mode kiosk

Dans node-red, il y aura un module qui permettra de mettre en place un interface entre le programme et l’utilisateur. Pour que l’utilisateur n’a pas besoin de réaliser certaines manipulations. On va lui ouvrir automatiquement cette interface qui sera en forme de site web. Vous pouvez y accéder en tapant dans votre navigateur la ligne suivante :

**http://[ IP ou localhost ]:1880/ui**

Pour réaliser l’ouverture automatique, tape la commande suivante

**sudo nano /etc/xdg/lxsession/LXDE-pi/autostrat**

Puis rajoute ce code :

**/usr/bin/chromium-browser --kiosk http://[ IP ou localhost ]:1880/ui**

Quitte et sauvegarde, puis faite un reboot. Laisser la Raspberry Pi redémarrer. Apres un court laps de temps vous aurez la fenêtre qui va s'ouvrir. Il y a une forte chance que le navigateur s'ouvre avant que node-red se lance et vous affiche une erreur. Attendez pendant une minute ou deux que node-red se lance et normalement le navigateur va se réactualiser et afficher l'interface.

Pour quitter du mode kiosk fait Ctrl + F4

Configuration du port série GPIO 14

Par défaut, la liaison serial0 est désactivée comme le montre cette commande

**sudo cd /etc**

**sudo ls -l /dev | grep serial**

Pour activer serial0, fait la commande suivante

**sudo cd /boot/config.txt**

et ajouter à la fin du fichier la ligne suivante :

**enable\_uart=1**

Redémarrer le raspberry et après vous pouvez faire la première partie pour voir la modification.

Par défaut, la raspberry utilise le port serial0 pour logger, un terminal de debug via un logiciel appelé « getty ». Si on veut utiliser le port pour entrer des données à partir d’un périphérique (GSM, radiometrix…) il faut désactiver le login console pour que le périphérique puisse contrôler le port.

Entrez la commande ci-dessous pour arrêter le service et le désactiver

**sudo systemctl stop serial-getty@ttyS0.service**

**sudo systemctl disable serial-getty@ttyS0.service**

Ensuite, dans le fichier /boot/cmdline.txt, supprimer la chaine console=serial0,115200.

**ATTENTION , LE FICHIER QUE VOUS MODIFIEZ SERT AU DÉMARRAGE DU SYSTÈME. IL NE DOIT CONTENIR QU’UNE SEULE LIGNE, SANS TABULATION. CONTENTEZ VOUS DE SUPPRIMER LA PARTIE ENTOURÉE EN ROUGE CI-DESSOUS.**

**sudo nano /boot/cmdline.txt**

Supprimer la chaîne “ console=serial0,115200 ” sans laisser d'espace au début de la ligne, redémarrer la Raspberry

Pour finir, on assigne la communication Bluetooth à serial1

**sudo nano /boot/config.txt.**

A la fin du fichier ajoutez la ligne suivante

**dtoverlay=pi3-miniuart-bt**

Sauvegarder le fichier et redémarrer !

Désactivation du mode veille

Pour désactiver le mode veille après 10 minutes , Il faudra installer un module

# Code informatique

1. Programme LecteurRFID.py

Ce code a pour but de gérer le lecteur RFID, quand il détecte une bague RFID, il lira et return un code. Il sera utilisé par un node "Exécution" dans le programme principale

| **import random**  **import time**  **import serial**  **import rdm6300**  **reader = rdm6300.Reader('/dev/ttyAMA0')**  **while 1:**  **card = reader.read()**  **if card:**  **print(f"[{card.value}]")**  **return card.value** |
| --- |

il faudra aussi installer les hardware pour que le programme marche correctement. Sur la pin 10 de la raspberry pi il faudra mettre le RX du lecteur RFID

1. Programme Principale
   1. Fonctionnalités Récupération RFID

La fonctionnalité récupération RFID nécessaire le programme LecteurRFID.py pour son fonctionnement. Il sera aussi couble avec la fonctionnalités “Forme” et “ Tableaux et sélection par le tableaux ”.

Les dépendances Node-red :

* Node-red-contrib-led
* Node-red-dashboard
* Node-red-node-mysql
  1. Fonctionnalités Forme
  2. Fonctionnalités Tableaux et sélection par le tableaux
  3. Fonctionnalités HTML Head
  4. Fonctionnalités Log
  5. Fonctionnalités Ajoutez une poule
  6. Fonctionnalités Enlever une poule
  7. Fonctionnalités Modifie une poule
  8. Fonctionnalités Test de connexion auto et manuelle
  9. Fonctionnalités Batterie
  10. Fonctionnalités On/Off Notification
  11. Fonctionnalités Get IP