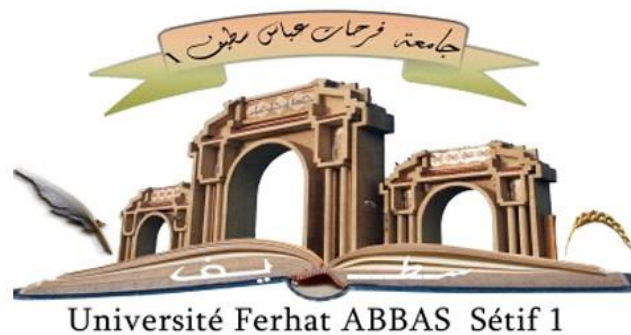


République Algérienne Démocratique et Populaire Ministère
de l'Enseignement Supérieur et de la Recherche Scientifique
Université Ferhat Abbès - Sétif



Faculté des Sciences

Domaine : Mathématique et Informatique
Spécialité : Réseaux et Systèmes Distribués

Présenté par :

DJIARA Abdelmalek

ZIGHEM Bouzid

Thème :

Réalisation d'une application de contrôle d'un tableau de
change en utilisant les sockets de la bibliothèque QT

Année Universitaire : 2024/2025

Supervisé par : Dr. GUELLATI Nabil

Table des matières

Introduction Générale	12
1 Raspberry Pi	13
1.1 INTRODUCTION	13
1.2 RASPBERRY PI MODELS	14
1.2.1 Raspberry Pi 1 (2012)	15
1.2.2 Raspberry Pi 2 (2015)	16
1.2.3 Raspberry Pi 3 (2016)	16
1.2.4 Raspberry Pi 4 (2019)	17
1.2.5 Raspberry Pi 400 (2020)	18
1.2.6 Raspberry Pi Pico (2021)	18
1.2.7 Raspberry Pi 5 (2024)	19
1.3 Systèmes d'exploitation supportés	20
1.3.1 Raspberry Pi OS (anciennement raspbian)	20
1.3.2 Ubuntu pour Raspberry Pi	20
1.3.3 Windows 10 IOT Core	20
1.3.4 RetroPie – Pour le Gaming Rétro	21
1.3.5 LibreELEC – Pour les Centres Multimédias	21
1.3.6 DietPi – Pour une Performance Optimisée	21
1.3.7 Kali Linux – Pour l'Éthique Hacking	21
1.3.8 Lakka – Rétro Gaming Simplifié	21
1.4 Cinq meilleurs langages de programmation pour Raspberry Pi	22
1.4.1 Python	22
1.4.2 Scratch	23
1.4.3 C/C++	23
1.4.4 Bash/Shell	23
1.4.5 Node.js	24
1.5 Raspberry Pi Home Projects	24
1.5.1 Télévision intelligent	25
1.5.2 Bouton intelligent	25
1.5.3 Miroir magique	26

	culatation	27
1.5.5	caméras de sécurité CCTV	28
1.5.6	Contrôle de la piscine	29
1.6	Comparaison entre le Raspberry Pi et l'Arduino ^{[3][4][5]}	29
1.6.1	Qu'est-ce qu'un Arduino?	29
1.6.2	Différences clés entre Arduino et Raspberry Pi	30
1.6.3	Intégration avec d'autres technologies	32
1.7	Limitations du Raspberry Pi	33
	1.8 Conclusion	34
2	Bibliothèque QT	36
2.1	INTRODUCTION	36
2.2	Brève histoire de Qt	37
2.2.1	Débuts de Qt (1990 - 1994)	37
2.2.2	Succès et évolutions initiales (1995 - 1999)	38
2.2.3	Évolution vers les systèmes embarqués et Qt 3 (2000 - 2001) ..	39
2.2.4	Qt 4 et son expansion majeure (2005)	39
2.2.5	Vente à Nokia et développement mobile (2008 - 2011)	39
2.2.6	Qt 5 : modularisation et ouverture (2012)	40
2.2.7	Qt 6 : modernisation et nouvelles technologies (2020)	40
2.3	Évolutions de Qt	42
2.4	Principales fonctionnalités de Qt	42
2.5	Plateformes supportées par version de Qt	44
2.6	Qt modules	45
2.6.1	Les essentiels de Qt(Qt Essentials)	45
2.6.2	modules complémentaires (Qt Add-Ons)	45
2.7	Les outils Qt	47
2.7.1	Qt Creator	47
2.8	Qt avec Paython	50
2.8.1	PyQt	50
2.8.2	PySide	50
2.8.3	Résumé : Différences entre PyQt et PySide	51
	Conclusion	52
3	Réalisation	53
3.1	Introduction	53
3.2	Problématique	53
3.3	Objectif de l'application	55
3.4	Architecture du système	55
3.5	Technologies utilisées	56
3.6	Développement de l'application de controle (ordinateur)	56
3.7	Développement de l'application d'affichage (Raspberry Pi)	56
	TABLE DES MATIÈRES	

3.8	la Communication	57
3.9	Résultats attendus	57
	3.10 Interface de l'application Contrôleur	57
3.11	Fonctionnalités de l'application	60
	3.11.1 Test de la connectivité	60
	3.11.2 Envoi des Cours (UDP)	61
	3.11.3 Envoi des fiches (TCP)	63
	3.11.4 Redémarrage à distance	65
3.12	Interface graphique – application d'affichage	66
	3.12.1 Affichage alterné : publicités et taux de change	67
	3.13 Problèmes rencontrés	68
	3.14 Conclusion	69
Conclusion Générale		70

Table des figures

1.1	Raspberry Pi 1.	15
1.2	Raspberry Pi 2.	16
1.3	Raspberry Pi 3.	17
1.4	Raspberry Pi 4.	17
1.5	Raspberry Pi 400.	18
1.6	Raspberry Pi pico.	19
1.7	Raspberry Pi 5.	19
1.8	Smart TV.	25
1.9	Smart button.	26
	1.10 Raspberry Pi 9.	27
	1.11 Garage opener with car plate recognition.	28
	1.12 CCTV security cameras.	28
	1.13 Pool control.	29
2.1	Le cycle de vie du développement d'application.	48
2.2	Interface d'accueil de Qt Creator.	49
3.1	Tableau de change à 7 segments.	54
3.2	schéma descriptif de fonctionnement 1.	55
3.3	schéma descriptif de fonctionnement 2.	56
3.4	Fenêtre principale du contrôleur.	57
3.5	Fenêtre de test de connexion avec retour visuel d'attente.	58
3.6	Interface de l'envoi des taux de change vers l'application d'affichage. .	58
3.7	Interface de l'envoi des taux de change vers l'application d'affichage. .	59
3.8	Code source – Fonction de test de connectivité (Ping).	60
3.9	Diagramme de séquence – Vérification de connectivité réseau.	61
3.10	Code source – Fonction d'envoi de cours.	62
	3.11 Code source – Méthode d'envoi depuis WindowCours.	62
	3.12 Diagramme de séquence – Envoi des cours via UDP.	63
	3.13 Code source – Fonction d'envoi de fiche (côté controle).	64
	3.14 Code source – Fonction d'envoi de fiche (côté affichage).	64
	3.15 Diagramme de séquence – Envoi des fiches via TCP.	65
3.16	Code source – Fonction de redémarrage (côté controle).	65
DES FIGURES		TABLE
3.17	Code source – Fonction de redémarrage (côté affichage).	66

3.18 Diagramme de séquence – Redémarrage à distance	66
3.19 Interface graphique – État initial (vide)	67
3.20 Interface graphique – Affichage des taux reçus	67
3.21 Exemple d’une image publicitaire affichée sur le raspberry	68

Résumé

Ce projet s’inscrit dans le domaine des systèmes distribués et a pour objectif principal la réalisation d’une application permettant de contrôler et d’afficher les taux de change en temps réel à l’aide de la bibliothèque Qt et d’un Raspberry Pi.

L’architecture adoptée repose sur un modèle réparti. L’application de contrôle, développée sur un ordinateur, récupère les données de change à partir d’une source en ligne, puis les transmet au Raspberry Pi via des sockets réseau. Deux protocoles sont utilisés : UDP pour l’envoi rapide des prix, et TCP pour les données plus sensibles comme les fiches.

Le Raspberry, quant à lui, reçoit les données et les affiche sur un écran connecté, en alternant entre les taux de change et des messages publicitaires. L’interface graphique est développée avec Qt, ce qui garantit une expérience utilisateur fluide et multiplateforme. Ce projet représente un exemple concret de mise en œuvre des concepts de communication réseau, de systèmes embarqués et de développement multiplateforme. Il ouvre également la voie à de futures améliorations telles que la sécurisation des échanges, le support multi-écrans ou encore l’intégration d’une base de données.

ملخص

يهدف هذا المشروع إلى إنجاز تطبيق موزع لعرض والتحكم في لوحة أسعار الصرف، بالاعتماد على مكتبة Qt ولوحة Raspberry Pi. يقوم الخادم بجلب بيانات الأسعار من مصدر على الإنترنت، ثم يرسلها إلى العميل (Raspberry Pi) باستخدام بروتوكولات الشبكة (TCP و UDP).

يعتمد النظام على هيكلية عميل-خادم، حيث يُستخدم بروتوكول UDP لإرسال الأسعار بسرعة، في حين يُستخدم TCP لنقل البيانات الأكثر حساسية. يقوم العميل بعرض المعلومات على شاشة، مع التبديل بين أسعار الصرف والإعلانات، من خلال واجهة رسومية صممت باستخدام Qt.

يمثل هذا المشروع تطبيقاً فعلياً لمفاهيم الشبكات، الأنظمة المدمجة، والبرمجة متعددة المنصات، كما يفتح المجال لعدة تحسينات مستقبلية كدعم قواعد البيانات، تأمين الاتصال، أو دعم شاشات متعددة.

Abstract

This project falls within the field of distributed systems and aims primarily to develop an application that allows controlling and displaying exchange rates in real-time using the Qt library and a Raspberry Pi.

The adopted architecture is based on a distributed model. The controlling application, developed on a computer, retrieves exchange rate data from an online source and then transmits it to the Raspberry Pi via network sockets. Two protocols are used : UDP for the fast sending of prices, and TCP for more sensitive data such as detailed records.

The Raspberry, for its part, receives the data and displays it on a connected screen, alternating between exchange rates and advertising messages. The graphical interface is developed with Qt, ensuring a smooth and cross-platform user experience.

This project represents a practical example of implementing concepts in network communication, embedded systems, and cross-platform development. It also paves the way for future enhancements such as securing exchanges, multi-screen support, or the integration of a database.