

DYODE

Do Your Own Dyode

Une diode réseau à bas coût pour les réseaux industriels

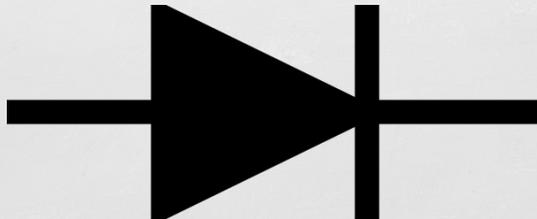
Arnaud Soullié, Solucom
Ary Kokos

1

solucom 

PRINCIPES DE LA DIODE RESEAU

- Passerelle unidirectionnelle
 - Utilisation d'une liaison optique pour la transmission
 - La jonction PN empêche les électrons de circuler dans le sens inverse
- ⇒ La propriété de sécurité est assurée physiquement. *Hack that.*



POURQUOI CE PROJET ?

**Retour d'expérience de missions SI industriel
⇒ Nombreux besoins d'échange de données**

Les diodes réseau existantes sont performantes, mais chères

⇒ Le rapport gain en sécurité vs prix n'est pas évident

Deux exemples :

- **Envoi de données d'usure des pièces : 100ko à envoyer toutes les 6h au prestataire en charge du MCO**
- **Systeme frigorifique : le prestataire exige une copie des données d'un automate en temps quasi-réel**

⇒ Ces deux besoins d'échange de données résultent en une connexion avec un SI tiers

LE PROJET DYODE

Basé sur des travaux existants : Lagadec, Austin Scott, Robert Gabriel

**Utilisation de matériel standard et développement d'un logiciel libre
Coût cible de 200€ par diode**

Objectifs

- **Prouver la faisabilité**
- **Créer une solution simple à mettre en place, transparente**
- **Partager nos résultats**



Note: le projet DYODE n'a aucune visée commerciale, mais une implémentation et une maintenance par un éditeur/constructeur est autorisée

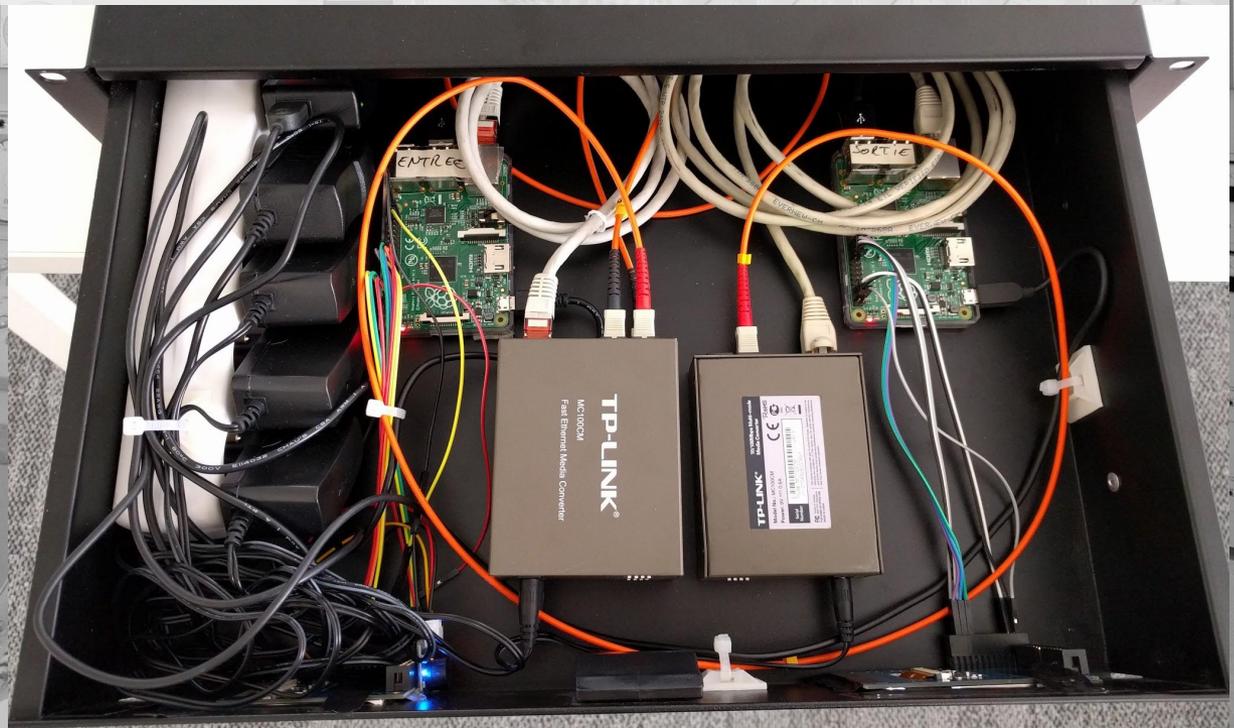
LE MATERIEL

- Utilisation de convertisseurs Cuivre-Optique pour assurer l'unidirectionnalité en ne branchant qu'un câble TX -> RX
- Utilisation d'un 3ème convertisseur pour "tromper" l'émetteur (*link failure protection*)
- Utilisation de Raspberry Pi pour les guichets d'entrée et de sortie



LE MATERIEL

- Utilisation de convertisseurs Cuivre-Optique pour assurer l'unidirectionnalité en ne branchant qu'un câble TX -> RX
- Utilisation d'un 3ème convertisseur pour "tromper" l'émetteur (*link failure protection*)
- Utilisation de Raspberry Pi pour les guichets d'entrée et de sortie



LE LOGICIEL

Volonté d'avoir une solution fonctionnelle rapidement

Utilisation du logiciel open-source "udpcast" pour le transfert de fichier plat via la diode

Construction de fonctions autour de cet outil via un développement en Python

3 fonctions supportées aujourd'hui

- Transfert de fichier plat
- Transfert de données Modbus
- Partage d'écran sans interaction (télémaintenance, etc)

Code source : ~500 lignes utiles

LIMITES ACTUELLES

- **Vitesse de transfert faible, quelques Mo/s**
- **Latence élevée du fait de l'utilisation de fichiers plats
⇒ Remplacé récemment par une implémentation directe via sockets UDP pour Modbus et partage d'écran**
- **Solution non-adaptée pour des usages critiques (composants standards non-durcis, pas de haute disponibilité)**

ROADMAP

Prochaines étapes

- Durcissement sécurité
- Fiabilisation (mise en place d'un heartbeat)

Améliorations possibles

- Ajout de connecteurs
 - S7 (automates Siemens)
 - SYSLOG
 - SMTP
 - ...
- Contrôle d'intégrité / d'inocuité des données transmises
- Signature des fichiers échangés
- Création d'une version *very low cost* basée sur Raspberry Pi Zero et liaison infrarouge

Des questions ?



@arnaudsoullie



arnaud.soullie@solucom.fr



<https://github.com/arnaudsoullie/dyode>