Interface standardisée communication entre PLCs et carte embarquée

Implémentation avec un panel boutons et MQTT

# Préambule

Il existe un grand nombre de bus de terrain actuellement en utilisation dans le milieu de l’industrie. Ces bus de terrain suivent des standards bien définis et reposent sur des technologies diverses et variées. Malheureusement, le choix du protocole utilisé est aujourd’hui presque complètement lié au fabricant du PLC que nous souhaitons utiliser. En effet, chaque marque a son bus de terrain de prédilection :

* Beckhoff : EtherCAT
* B&R : PowerLink
* Siemens : profiNET / profiBUS
* Allen&Bradley / Rockwell: Ethernet/IP (DeviceNet™ and ControlNet™)
* Schneider Electric : canOPEN

Ce qui rend l’interopérabilité de ces systèmes difficile.

Cependant, tous ces automates un en règle générale un connecteur réseau et sont utilisable pour l’utilisation du protocole TCP/IP. Il est donc judicieux d’utiliser cette interface commune.

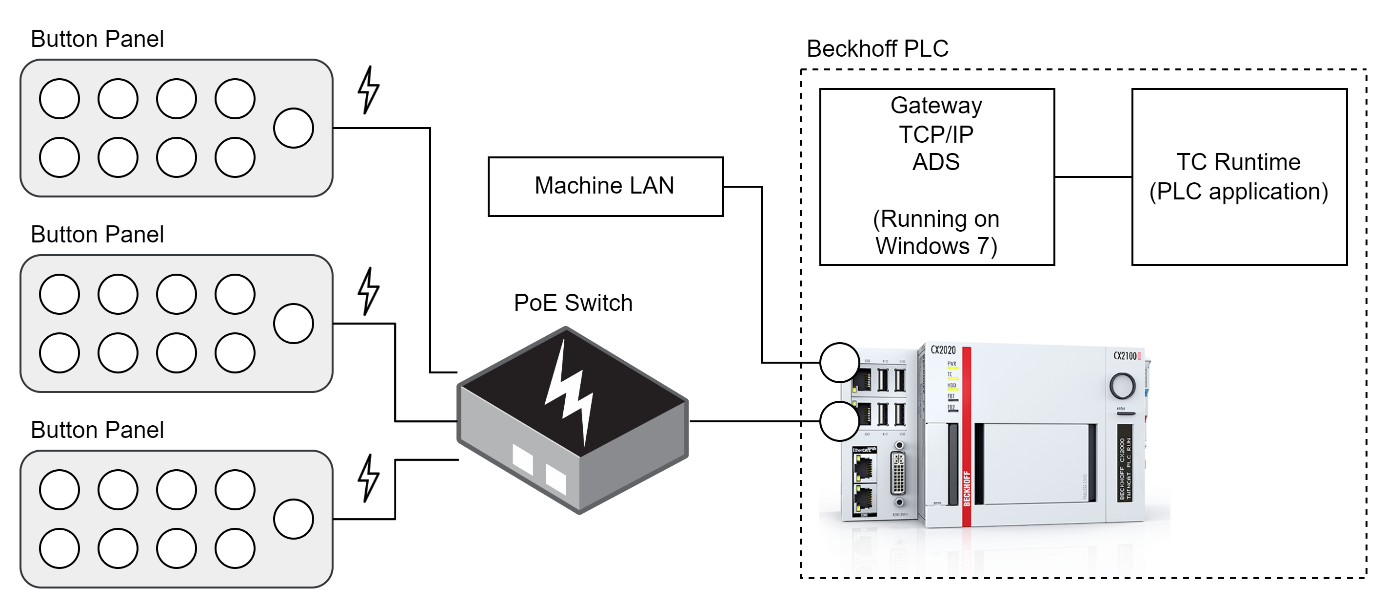
# Besoin

L’objectif est de trouver un protocole de communication basé TCP/IP (ou UDP) répandu et qui sera facilement configurable depuis les différents IDE fournis par les différents fabricants de PLC. Ce protocole doit ensuite pouvoir être intégré dans une carte embarquée produite en petite quantité (< 1000 pces) en restant simple, flexible et peu onéreux.

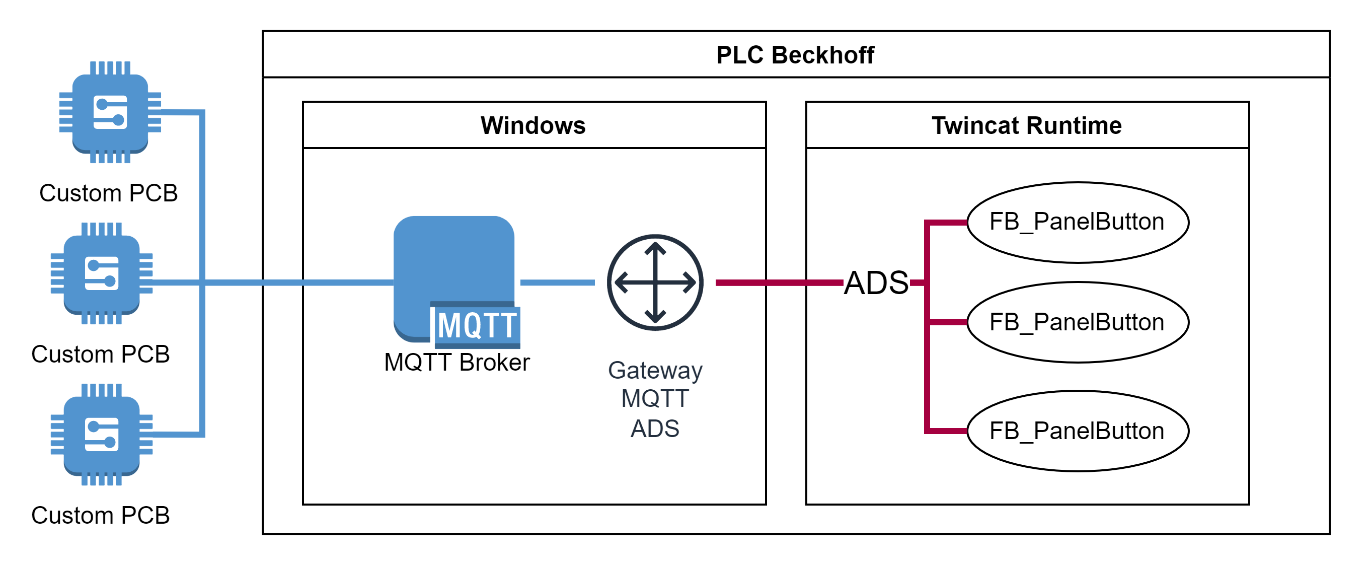
De plus, afin de limiter la configuration et le temps de câblages de ces différents éléments standard, Help-tec souhaite, si possible, que le protocole sélectionné utilise le même câble que l’alimentation des différents nœuds du réseau. Une des variantes de la technologie PoE (Power over Internet) est à privilégier. Un protocole de communication sans fil n’est pas souhaité actuellement, car les machines sont composées majoritairement d’aluminium relié à la terre. Certains nœuds pourraient alors être inaccessibles.

# Schéma bloc

Montage :



Communication/Software :



# Points à réaliser

* Création d’un PCB pour y mettre des boutons et un potentiomètre
  + Se baser sur la platine Help-tec pour les dimensions
  + MCU : ESP32
  + 1 interface Ethernet
  + Alimentation via PoE (standard à utiliser : TBD par l’élève)
  + Configuration IP statique (comment : Proposition par l’élève)
  + LED de statut (à définir par l’élève)
  + Au moins 8 boutons-poussoirs + Leds, dimension et montage à définir
  + 1 potentiomètre sur ADC de l’ESP32
  + Voir pour les protections galvaniques
* Software embarqué
  + Implémentation de MQTT Publisher/Subscriber,
  + Drivers nécessaires (ADC, stack TCP/IP, ...)
  + Mise à jour E/S de la carte via MQTT (Bouttons et LEDs)
  + Configuration MQTT via un fichier JSON
  + (SECONDAIRE) Configuration client MQTT
    - Ajouter un serveur web (requête REST) pour configurer MQTT ou autre solution OU
    - Utiliser l’antenne Bluetooth / WLAN de l’ESP32 pour le configurer
* Software MQTT Broker sur le PLC (windows)
  + Installation d’un Broker sur le PLC (Exemple : Mosquitto)
* Gateway MQTT <-> ADS
  + Programme en python
  + Lire écrire des variables dans le PLC
  + Lier les variables du PLC aux informations disponibles du Broker MQTT
* Mécanique
  + Montage des composants sur le PCB
  + Montage du PCB sur la platine Help-tec
  + Test avec une peau (Étiquettes autocollantes) A4 ou autre format à imprimer soi-même

# Contraintes

* Prix du PCB sans les boutons mécaniques : 20 - 30 CHF (PCB + composant)
* Latence entre pression du bouton et changement d’état dans la Runtime Twincat : 200 ms
* Utilisations de composants disponibles actuellement
* Design du PCB avec KiCAD
* MCU
  + ESP32-WROOM, révision au choix (S1/S2/S3)
    - Flash size : à définir
    - Choisir la version avec antenne en PCB
  + Code en C + FreeRTOS
  + Utilisation de Git, code héberger sur Github dans un repo public chez Help-tec
  + IDE : VScode ou Eclipse
  + toolchain de Espressif + ESP-IDF (pas la version pour Arduino)

# Dimension

Le PCB doit s’intégrer sur une platine en acier. Celle-ci sera découpée à la commande pour correspondre aux dimensions du PCB. Le placement des boutons doit être le mieux possible pour une utilisation ergonomique. Une peau autocollante sera ajoutée au-dessus de la platine et des boutons pour afficher un symbole de la fonction du bouton et cacher le PCB.

* Les dimensions exactes du PCB sont à définir selon la disposition sur la platine de montage
  + Largeur : 40 – 60 mm
  + Longueur : 60 – 90 mm
  + Hauteur : Selon bouton et platine de montage
* Les boutons ou le capuchon du bouton doivent avoir un diamètre d’au moins 10mm. La disposition des boutons sur le PCB devra être validée par Help-tec.
* Fixation du PCB à la platine : TBD
  + Avec des goujons intégrés à la platine et filetage M3
  + Perçage puis vis + écrou
  + Autre

Exemple du montage de la platine avec la solution actuelle, avec dimension :



# Matériel

Pour la réalisation du projet, les installations de l’HE-ARC St-Imier sont à privilégier. Seul le matériel pour le montage du PCB ainsi qu’un câble Ethernet est requis pour effectuer les tests sur le PCB. Afin de gagner du temps, Help-tec met à disposition un kit de développement comprenant une carte avec un MCU ESP32 alimenté via PoE et/ou usb.

La partie concernant le PLC peut être intégralement réalisée sur le Laptop de l’élève.

Si besoin et pour la démonstration, le matériel suivant peut être mis à disposition de la part de Help-tec à l’élève :

* Alimentation 24VDC
* PLC Beckhoff CX9020
* Switch PoE
* Divers câbles RJ45 Cat5 / Cat6

# Fonctions

Roger Schluep est le chef de projet côté Help-tec

* Coordonne la réalisation de la partie software avec l’élève
* Mets à dispositions les ressources nécessaires en ce qui concerne Help-tec
* Valide les points importants

Sacha Wittmann est le directeur général de Help-tec

* Valide les points concernant les coûts et l’esthétique du produit (si nécessaire)

Yves Meyer est le professeur responsable du suivi de la thèse de Master

* Suivi hebdomadaire du projet