

bus CAN + Arduino pour la surveillance des cellules solaires

Détecter et localiser les panneaux défectueux dans les grands réseaux photovoltaïques

image: Shutterstock

Somnath Bera (Inde)

Pourquoi restreindre l'excellent bus CAN aux seuls capteurs embarqués dans les véhicules ? Voici comment ce bus a été mis à contribution pour lire le rendement de chaque panneau photovoltaïque individuellement ainsi que son numéro afin de localiser et d'identifier la source de problèmes. Cette idée de détournement de bus est venue à l'auteur quand il a été confronté à des clients, des installateurs et des autorités en désaccord à propos des résultats décevants d'un réseau de cellules solaires de 25 kW en exploitation commerciale.

L'installation à surveiller consiste en un réseau de cellules photovoltaïques sur un toit : 55 panneaux (24 V) délivrent 25 kW pour alimenter un grand onduleur monophasé à 220 V qui à son tour charge un essaim de batteries. Le consommateur est un conglomérat de magasins ayant besoin d'électricité le soir pour les ventes et d'autres activités. Ces commerces étaient censés bénéficier de 6 h d'éclairage, de 17h30 à 23h30. L'installation a été inaugurée en grandes pompes.

Laver et essuyer

Le réseau photovoltaïque, l'onduleur et les batteries ont fonctionné pendant... un certain temps. Puis, sur deux mois environ, la performance s'est progressivement détériorée : même avec des périodes de charge de 10 à 12 h, l'alimentation du soir ne tenait pas plus de 3 à 4 h. Ce fut le début des chicanes. L'administration locale a incriminé le travail des installateurs, la médiocre qualité des panneaux, etc. Il a suffi de jeter un coup d'œil aux panneaux sur le toit pour comprendre. Les 55 panneaux d'une puissance de 500 W chacun répartis sur trois toits sont survolés du matin au soir par de nombreux volatiles attirés par un grand marché à ciel ouvert à proximité : la plupart étaient couverts de fiente.

En attendant de déplacer les panneaux vers un endroit moins fréquenté par les oiseaux, j'ai proposé qu'on lave chaque panneau tous les deux jours. Ce nettoyage régulier a permis

de rétablir les performances initiales. Peu après, il s'est avéré qu'un nettoyage sélectif suffisait, rendant inutile le lavage systématique. Le préposé au lavage devait néanmoins d'abord vérifier visuellement tous les panneaux, puis procéder au nettoyage sélectif. Cet examen prend beaucoup de temps, car le préposé doit faire le tour de tous les panneaux. Pour cette installation de seulement 55 panneaux, ce n'est pas un problème, mais pour une autre installation similaire, prévue avec une capacité de 100 kW, un cauchemar se préparait pour le préposé au nettoyage. C'est là qu'est née l'idée d'un système de surveillance individuelle des panneaux qui permettrait à la personne chargée du nettoyage de se rendre directement aux seuls panneaux dont le rendement était affecté par des déjections.

Détournement du bus CAN

La combinaison du réseau CAN (*Control Area Network*) avec l'Arduino est une excellente solution pour mesurer les performances des différents panneaux photovoltaïques. Inventé par Bosch pour l'automobile, le bus CAN permet de collecter les signaux de centaines de capteurs connectés dans un véhicule. Parmi les paramètres mesurés sur une voiture, il y a la température int/ext., la pression des pneus, le niveau du liquide de frein, la température du moteur, la vitesse, la température de la climatisation et bien d'autres. Tous les capteurs sont connectés par deux fils seulement dans une structure de bus et reliés

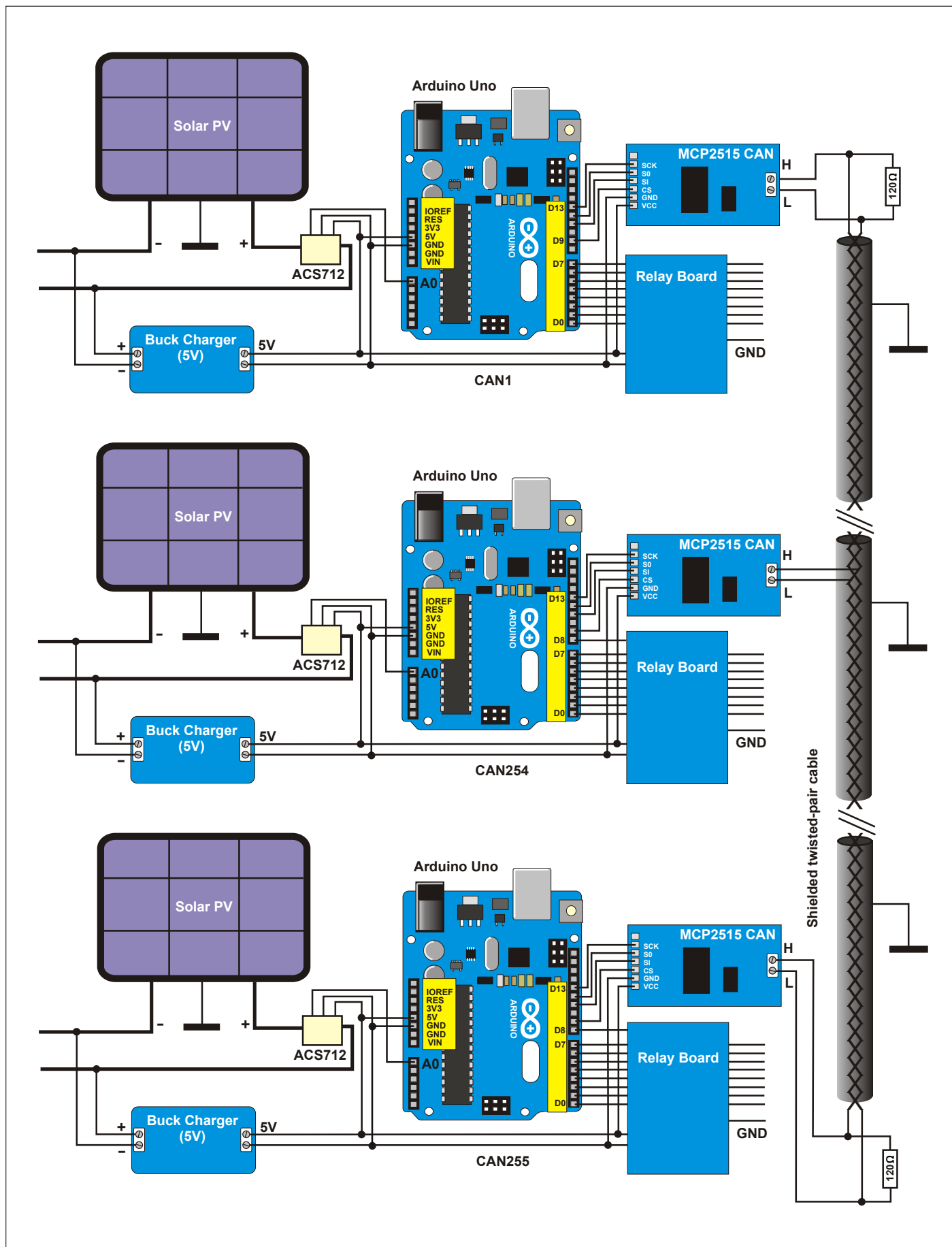


Figure 1. Schéma fonctionnel d'un système de surveillance pour réseau de jusqu'à 120 panneaux solaires (seulement 3 sont représentés ici) basé sur un bus CAN. Chaque panneau est surveillé individuellement pour détecter toute défaillance du courant de sortie, quelle qu'en soit la cause. Un Arduino UNO se voit attribuer le rôle de maître du bus CAN, les autres sont des esclaves.

à l'ordinateur central qui surveille les signaux de sortie des capteurs et, si nécessaire, répond en émettant des commandes aux capteurs concernés, là encore par le biais du bus à deux fils. Le bus CAN va de nœud en nœud avec un câble blindé à paires torsadées d'une impédance de 120 Ω. Pour une bonne adaptation et pour réduire la réflexion du signal, les deux nœuds d'extrémité du bus CAN sont équipés d'une résistance de terminaison de 120 Ω.

Comme émetteur-récepteur CAN bon marché, nous utilisons la puce MCP2515, qui vient habituellement de Chine à moins de 1,50 \$. Du côté du microcontrôleur, elle est prête pour le SPI et du côté CAN, il y a un séparateur entre les fils de bus CAN H (haut) et CAN L (bas). On utilise un câble de cuivre à paires torsadées blindées de bonne qualité et durable. Pour le µC, nous proposons l'ATMEGA328 sur un Arduino UNO, qui reçoit les signaux du capteur sur ses broches d'E/S numériques et analogiques. Pour commander les actionneurs, vous pouvez connecter des relais, des servos et d'autres dispositifs de sortie sur les autres broches d'E/S.

Par souci de simplicité, nous nous contentons du CAN 2.0 qui accepte des données de 0 à 8 octets à la fois d'un nœud à l'autre. Tous les nœuds, y compris le maître, reçoivent un identifiant (0 à 255) pour l'émission et la réception. Plus cet identificateur est bas, plus la priorité pour l'envoi et la réception par cet appareil est élevée. Habituellement, c'est le maître de bus qui reçoit l'identifiant le plus bas. La vérification et la correction des erreurs par le protocole CAN sont excellentes. Le nombre de nœuds peut varier de 0 à 130 sur le même bus CAN dont la longueur varie de quelques mètres à quelques kilomètres. Vitesse de transmission et distance entre les nœuds finaux sont inversement proportionnelles. Au-dessus de quelques centaines de mètres, pour assurer l'immunité au bruit et l'intégrité du signal, il convient d'utiliser un répéteur de bus CAN. Le nombre maximum de nœuds autorisé sur un bus CAN 2.0 typique est d'environ 120, ce qui reste bien supérieur aux 32 que propose une alternative comme le MODBUS RS485.

débit	bus	tronçon de câble
1000 kbit/sec	40 m	0,3 m
500 kbit/sec	110 m	0,3 m
250 Kbit/sec	240 m	0,3 m
125 Kbit/sec	500 m	0,3 m
100 Kbit/sec	500 m	0,3 m
50 Kbit/sec	1300 m	0,3 m
20 Kbit/sec	3300 m	0,3 m
10 Kbit/sec	6600 m	0,3 m
5 Kbit/sec	30 km	0,3 m

Surveillance des panneaux photovoltaïques (PV)

Le bus CAN permet de surveiller jusqu'à 120 panneaux et lire leurs performances individuelles, c'est-à-dire la tension et le courant contribués par chacun d'eux au courant continu total fourni à l'onduleur. Le schéma de principe (**fig. 1**) du système de surveillance des performances des cellules photovoltaïques connectées en parallèle : elles produisent la même tension, mais leurs courants de sortie divergent. Un simple capteur de courant est connecté à la borne de charge de chaque

panneau photovoltaïque. L'ACS712 est une sonde de courant à effet Hall. La version -30 peut mesurer jusqu'à 30 A. La bande de barrage permet de connecter l'appareil en série avec le circuit d'alimentation PV. En Chine, l'ACS712-30 coûte environ 2 \$/pièce. Pour lire l'intensité du courant continu d'un capteur de courant ACS712-30 dont la sortie est connectée à la broche A0 de l'Arduino UNO, il suffit de programmer :

```
load_current = 0.075714*(analogRead(A0) - 512)
```

L'alimentation sous 5 V de l'UNO est assurée par un petit régulateur abaisseur (2 \$/pièce) connecté au bus des panneaux (24 V). Son principal avantage est sa très faible dissipation de puissance.

Outre les saletés qui leur font de l'ombre, les cellules photovoltaïques souffrent des petites bulles d'humidité qui se forment à leur surface, de la décoloration de la surface, des microfissures et des points chauds (dits *hot spots*). La perte de puissance de la cellule résultant de ces facteurs peut atteindre 35 %, ce qui se traduit par des pertes d'intensité du courant. À défaut d'approfondir la mesure de la contribution *absolue* du panneau au courant, on peut se contenter de ce calcul :

```
load_current = analogRead(A0) - 512
```

Cette approximation est comparée à une référence définie ainsi qu'aux valeurs d'autres panneaux PV en vue de décider de déclencher ou non une alarme. Après avoir vu le signal d'alarme et le lieu (numéro), le préposé au nettoyage se rendra au panneau en question pour en vérifier l'état. Soit un bon nettoyage suffit, soit il faut chercher d'autres causes (points chauds, humidité excessive, etc.). Dans le pire des cas, le panneau entier doit être remplacé.

Bus Can 4*4

Le projet permet d'intégrer un clavier 4*4 sur le bus CAN, à des fins de commande et de saisie locale de chiffres. Les **figures 2** et **3** montrent le schéma et le dispositif expérimental sur l'établi de l'auteur. Vous verrez que le clavier est utilisé en option dans le logiciel du projet.

Le logiciel du projet

Le logiciel de ce projet prend la forme d'un ensemble de croquis Arduino hébergé sur le site d'Elektor Labs [1] avec des mises à jour et des notes de l'auteur. Le logiciel doit être considéré comme un point de départ, et non comme une application figée. Il entend stimuler votre propre ingéniosité et votre créativité maintenant que le premier pas est fait. On distingue maître et esclave. Un esclave lit l'intensité du courant continu sur le port A0 et la transmet au maître. Le maître a deux modes de fonctionnement : affichage et commande. Le maître CAN reçoit l'ID «1», les autres unités se voient attribuer «254» et «255» comme dans la figure 1, ou tout autre nombre entre 2 et 253. Le nombre maximum de nœuds est de 120.

Maître en mode d'affichage : Ceci est obtenu en utilisant la broche D10 des entrées/sorties numériques. Lorsque D10 est haute, le maître passe en mode d'affichage, dans lequel il collecte les données de tous les esclaves et pour les afficher en une structure de grille. Il détermine ensuite si un paramètre (p.ex. la puissance) est inférieur au niveau de consigne et émet une alarme pour le nettoyage de ce panneau particulier. Ce mode convient bien pour la surveillance des panneaux photovoltaïques.

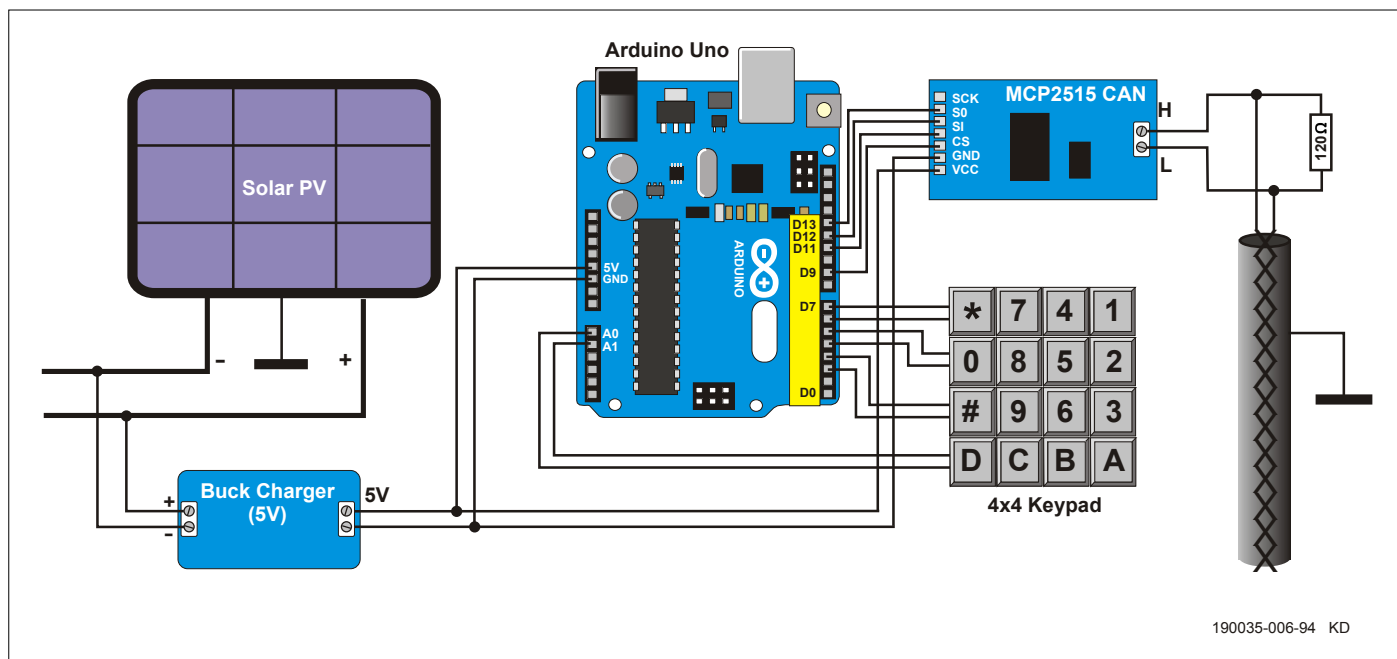


Figure 2. Schéma de principe du CAN Bus Master, avec un clavier 4x4 pour une utilisation en mode de commande.

Maître en mode de commande : Lorsque D10 est en l'air ou au niveau bas, le maître passe en mode de commande. Dans ce cas, les nœuds sont connectés à des relais commandés par les broches numériques D0 à 8. Un clavier CAN 4*4 est connecté sur l'appareil CAN maître (ID = 1). Le maître émet une chaîne de commande comme

25581#,25580#,13011#

etc. et les relais des nœuds respectifs sont activés ou désactivés pour commander, par exemple, une douche et/ou un essuie-glace connecté aux cellules PV. Exemples :

25580# : 255 = n° de nœud ; 8 = n° de broche numérique ;
0 = désactivé ; 1 = activé

25471# : 254^e nœud, relais connecté à la broche
numérique 7 et 1 = activé

25130# : 251^e nœud, relais connecté à la broche 3 et
0 = désactivé

Le caractère dièse (#) clot la commande.

Le maître du bus CAN peut avoir son propre capteur de courant PV, connecté à sa broche A0. Il peut tout aussi bien être dépourvu de tout panneau PV.

Votre application ?

Une autre utilisation de ce type de configuration consiste à déterminer, dans une petite ville ou un parc, combien de lampadaires sont effectivement allumés ou éteints. Un système tel que celui décrit ici permet également de contrôler de nombreuses

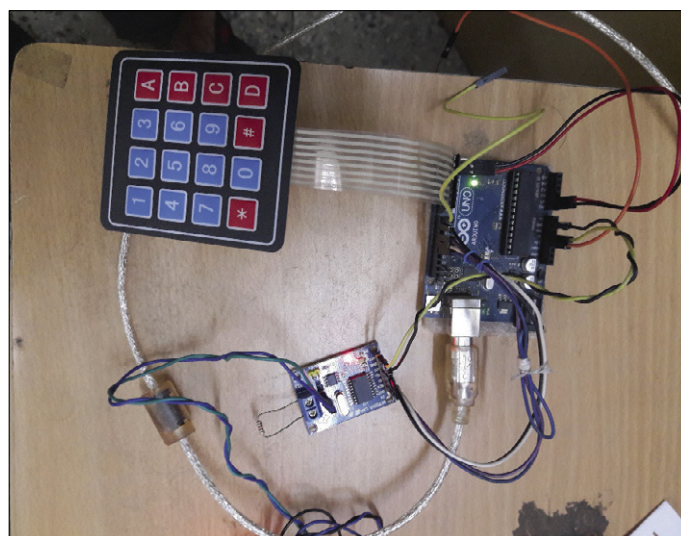


Figure 3. Version expérimentale du maître du bus CAN complétée par un clavier 4x4.

petites unités connectées sur le bus CAN à partir d'un bureau central. Parmi d'autres utilisations possibles, citons les boutons d'appel des chambres d'hôtel ou la commande de l'éclairage d'un hôtel. Au lieu de poser toutes sortes de câbles à l'intérieur du bâtiment, un seul bus CAN à paires torsadées suffit pour une commande et des mesures à la fois étendues et fiables. ◀

190035-02

Lien

[1] Projet sur Elektor Labs : www.elektormagazine.com/labs/can-bus-for-solar-pv-cell-monitoring