

## **MODEM PLCBUS OEM**

PLCBUS-4809

### **Présentation du module :**



Ce module est un modem permettant de communiquer avec les équipements PLCBUS.

Il peut remplacer à l'identique un module XM10 (modem X10).

Vous pourrez ainsi piloter les modules PLCBUS à partir de votre alarme.

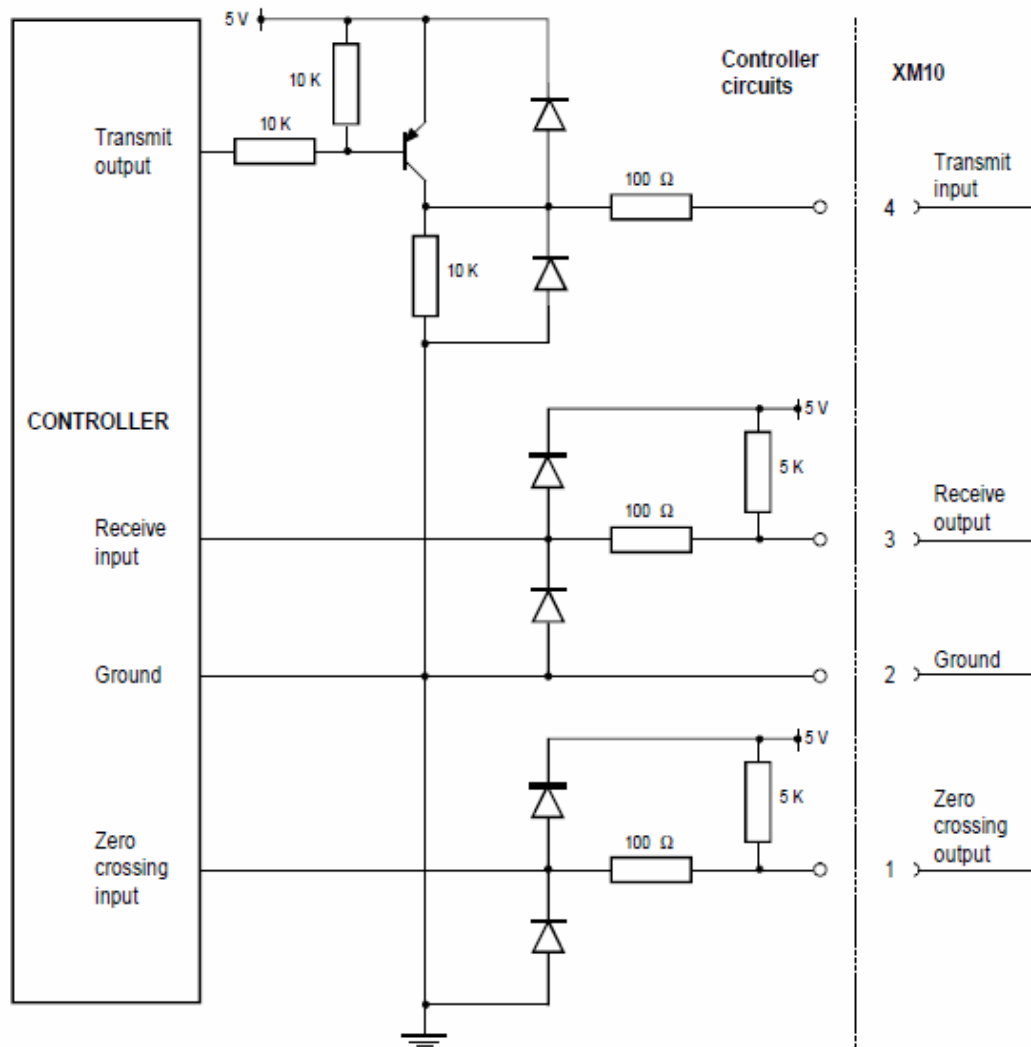
### **Configuration :**

Pour définir le code utilisateur, il faut appuyer sur le bouton SETUP pendant 5 à 10 secondes jusqu'à ce que le voyant s'allume et clignote, envoyer ensuite un ordre avec n'importe quel contrôleur PLCBUS.

Vous pouvez par exemple utiliser le PLCBUS 1141 et notre logiciel Configurateur PLCBUS disponible à l'adresse <http://www.planete-domotique.com/plcbus>

Une fois le code utilisateur défini, vous pouvez utiliser directement les codes A1 à P16 comme cela se ferait avec le XM10.

**Schéma de branchement :**



La connectique du RJ11 du module PLCBUS 4809 est le suivant :

- La borne 1 permet de détecter le *zéro crossing* afin d'encoder les trames
- La borne 2 est connectée à la masse
- La borne 3 permet de recevoir les données PLCBUS sur le réseau
- La borne 4 permet de transmettre les données PLCBUS sur le réseau

**Caractéristiques :**

- Tension d'alimentation : 230V +/- 10% 50Hz
- Température de fonctionnement : -10°C à +50°C



```

        delay_ms(1);
        output_low(X10_FROM_PIC);
    }
}

void x10_write(byte house_code, byte key_code) {
    byte i;

    i=0;
    while (X10_HOUSE_CODES[i]!=house_code)
        i++;
    house_code=i;
    if(key_code<16) {
        i=0;
        while (X10_KEY_CODES[i]!=key_code)
            i++;
        key_code=i;
    }
    x10_write_bits(7,4,1);
    x10_write_bits(house_code,4,0);
    x10_write_bits(key_code,5,0);
    x10_write_bits(0,6,1);
}

byte x10_data_ready() {
    port_b_pullups(TRUE);
    return(!input(X10_TO_PIC));
}

byte x10_read_bits(byte n) {
    byte data,i;

    for(i=1;i<=n;++i) {
        wait_for_zero_cross();
        delay_us(300);
        shift_right(&data,1,input(X10_TO_PIC));
        wait_for_zero_cross();
        delay_us(300);
    }
    data>>=8-n;
    return(data);
}

void x10_read(byte *house_code,byte *key_code) {

    port_b_pullups(TRUE);
    x10_read_bits(2);
    *house_code=x10_read_bits(4);
    *house_code=X10_HOUSE_CODES[*house_code];
    *key_code=x10_read_bits(5);
    if(*key_code<16)
        *key_code=X10_KEY_CODES[*key_code];
}

```