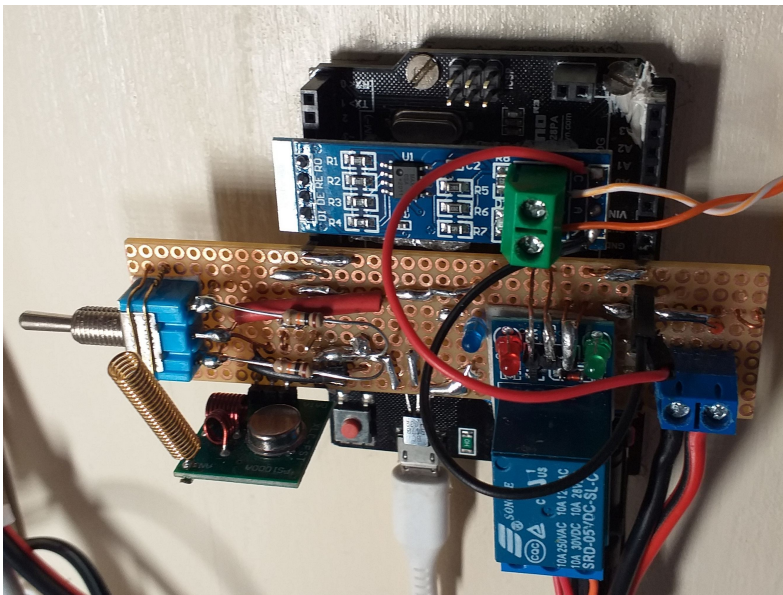
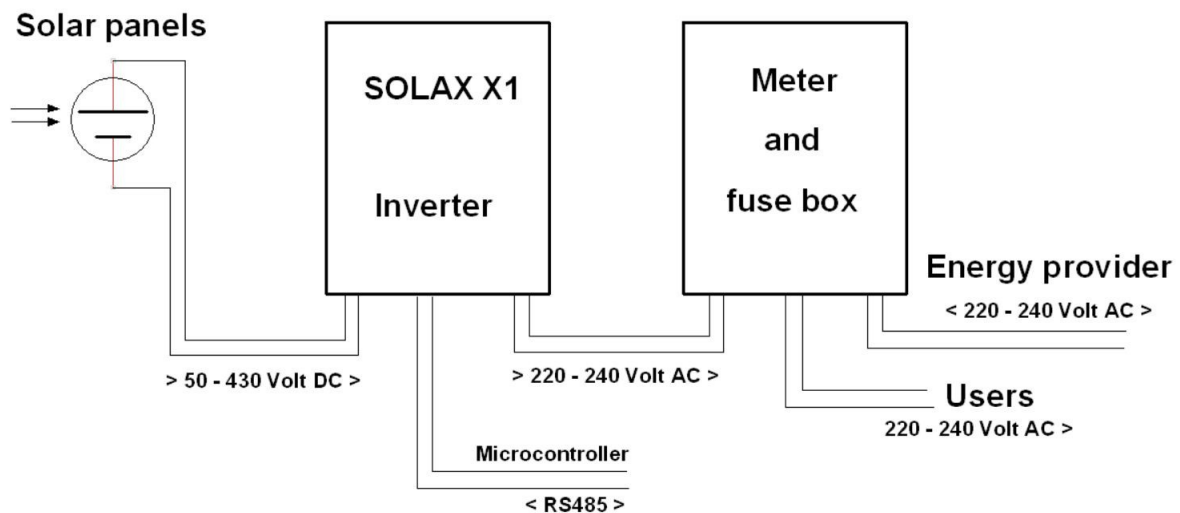


Building an Arduino shield to control sockets and cooling fan

Purpose:

1. **The direct use of solar power** while it is provided by the solar-panels.
Switchable plug-in sockets are used for power users like electric boiler, battery chargers, heaters or other appliances. The boiler socket can be hard wired or Remote Controlled with a small 433MHz transmitter.
2. **To switch a cooling fan "on"** when the temperature of the inverter rises above a certain level. (to prevent the inverter from powering down)



The internal data of the inverter is accessed via a MAX485 module at the top. (with the green connection block)

It is "plugged in" on digital data ports 3, 4, 5 and 6 at the left side.

The orange and white (twisted) RS485 A and B wires, coming from the green connection block, are connected to RJ45 jack pin 4 and 5 of the inverter.

VCC of the MAX module (red wire) is plugged on a pin sticking through the shield into the 3,3 V outlet of the Arduino.

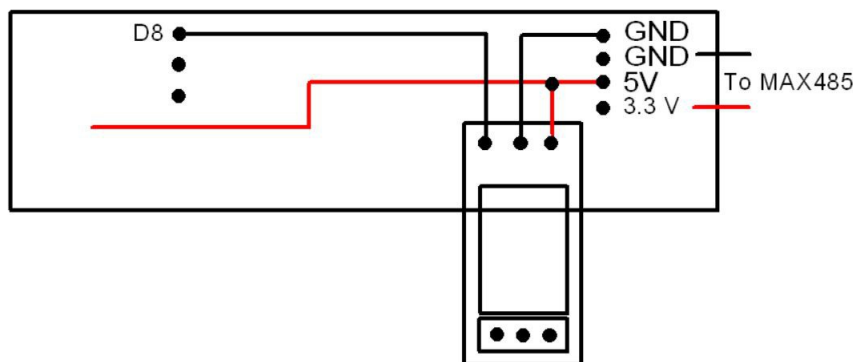
GND of the MAX module (black wire) is likewise connected to the Arduino's GND close to the 3.3V pin.

The shield contains 4 circuits which sometimes share connections:

1. Digital OUTPUT D8 controls the fan:

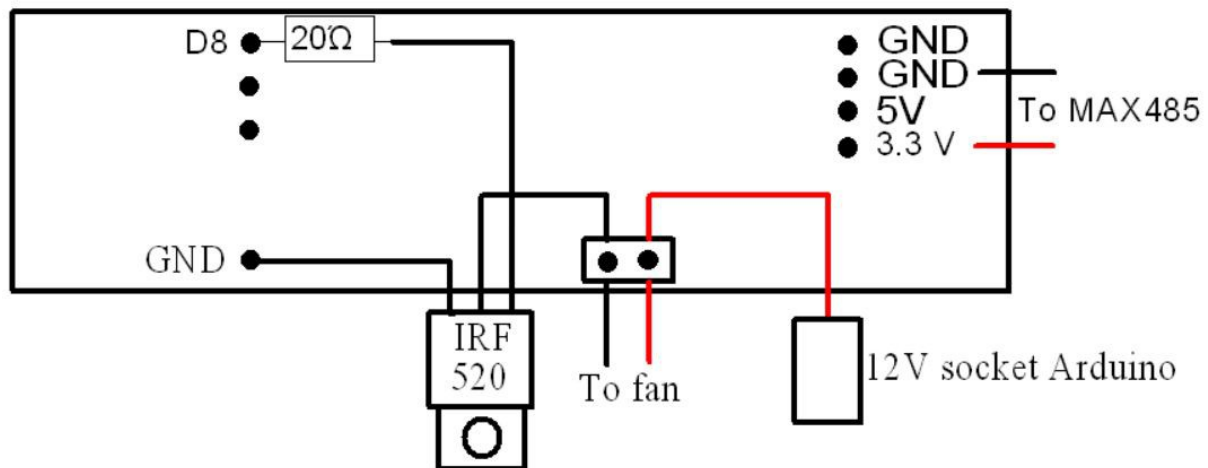
When the temperature of the inverter reaches a certain level, D8 is programmed to LOW. The relay module provides the fan with the same power feeding the Arduino. (A 230V > 12 V power adaptor) To switch the fan “off”, D8 is programmed to HIGH. The relay module uses the 5V and GND from the Arduino as illustrated:

D8 Output 5V to Cooling FAN relay



An alternative, which I used later, uses a MOSFET:

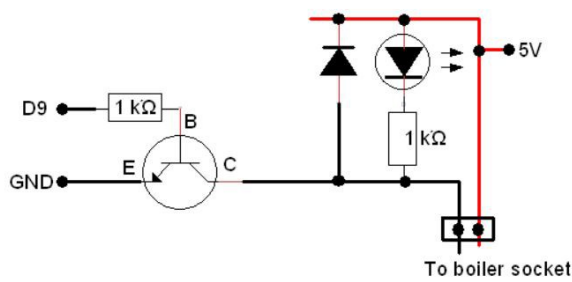
D8 Output 5V to Cooling FAN FET



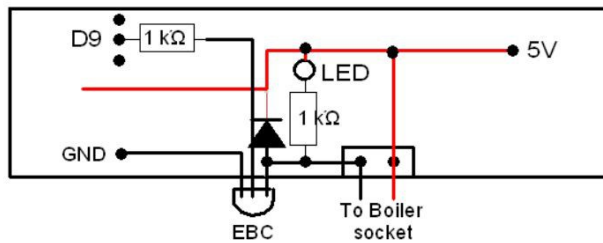
To switch the fan “on”, D8 is programmed to HIGH

2. Digital OUTPUT 5 V to switch a hardwired boiler socket

The socket part in which the boiler is plugged, is controlled by a relay. This relay switches the 230V "on" when it is powered with 5V from the Arduino. In our case we let the digital output D9 switch the negative output to the relay of the boiler socket. We do this as illustrated with a transistor. The (blue) LED on the Arduino shield, as well a LED on the boiler socket, indicate "the power on" situation.



D9 Output to boiler socket relay



Note: When a (not hard wired) remote controlled boiler socket is used, only the blue led with a 1 k resistor are needed (Between D9 and GND) (If you want an indication) The remote controlled socket is triggered by the transmitter. See circuit 4

3. Digital INPUT D10 and D11 from mode switch

A 3-way switch lets u choose from different modes:

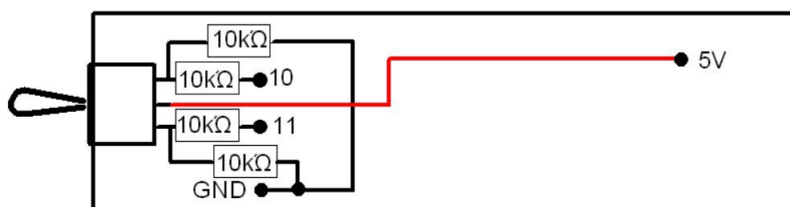
switch up - power always on the boiler socket

switch down - power boiler socket off

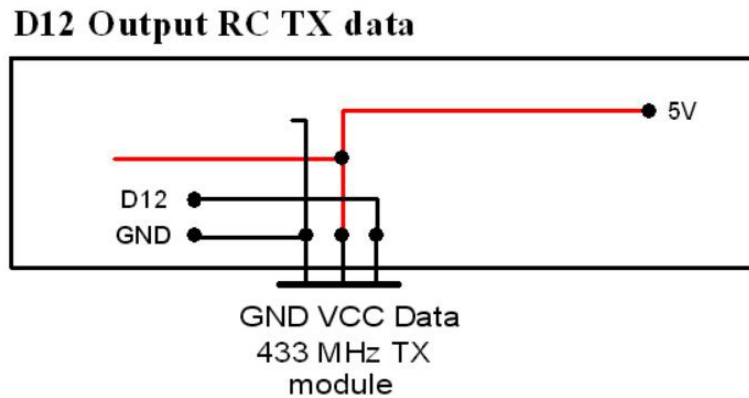
switch centre position - boiler socket regulated by the power value given by the inverter.

In the first case 5V is put on pin 11 and in the second case 5V is put on pin 10 both pins are set to INPUT mode, In the 3th case pin 10 and 11 are 0V. Pull down resistors are used to make 0 really 0.

D10 and D11 Input from switch to select mode



4. Digital OUTPUT 12 sends the codes for de remote controlled switches to the small 433MHz transmitter. GND and VCC of the transmitter are connected as illustrated.



The print is used "upside down" to facilitate the fixing and connecting of the header pins which stick into the Arduino. Except for the 3.3V pin and one GND pin (to power the MAX 485 Module) they are bend flat and soldered on. I used an old not working Arduino while assembling the parts. Some of the parts are on the not visible side.

For the programming, further explanations and last updated version, you can look on these pages:

<https://www.bootprojecten.nl/solar-energy/solar-power-regulated-socket>

<https://www.bootprojecten.nl/solar-energy/programming-rc-sockets-433mhz>

<https://www.bootprojecten.nl/solar-energy/getting-data-from-a-solar-panel-inverter>

<https://www.bootprojecten.nl/solar-energy/solar-controlled-socket-shield-for-arduino>

Feedback or questions? Mail to bootprojecten@gmail.com
(Dutch or English)

Jeroen Droogh