

1 Setup of the project

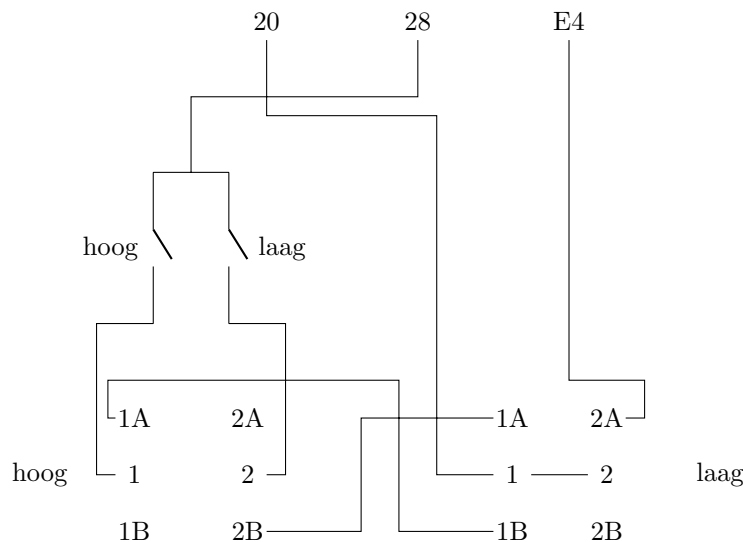
1.1 Hardware used

We used an Arduino Nano ATmega328, bought at [rs-components](#).

We used a [ENC28j60 Ethernet shield](#), the version specifically for the nano. This seemed easier than a wifi shield because of this reason, and with a wifi shield it seemed we needed extra components and a circuit, and we didn't really understand it.

1.1.1 Button circuit

We implemented the begin- and endstops in the circuit for the buttons that move the panels up and down. When a button is NOT pushed, 1 and 1B, and 2 and 2B of that button are connected. When a button is pushed, connections are made between 1 and 1A, and 2 and 2A of that button.



1.1.2 Potentiometer

The yellow-green wire from the potentiometer to the main control box is the signal wire, on connection 8 in that box. With the Arduino we measured 74 (analog in so scale 0-1024) in the highest position and 360 in the lowest position, with connection 9 to ground on the Arduino.

1.1.3 Motor control

We bought BC517 NPN transistors to control the 12V/20mA ([docs](#), page 4-11) or 14V/40mA (measured) current of the EVF8202-E frequency inverter, and also ...K ohm base resistors otherwise the Arduino needs to give too much

current to the transistor, and the transistor will be slow to turn off because of 'base charge storage'.

1.2 Software used

We decided on the [PlatformIO IDE](#), (which uses python 2.7 and Clang for autocompletion) because it is a lot better than the standard Arduino IDE, and also seemed better than the Stino plugin for Sublime Text 3. A plugin for CLion also looked good but we didn't get that to work. PlatformIO only worked when we imported an existing Arduino project, creating new files or new projects resulted in all kinds of errors.

2 Code

2.1 Internet/Ethernet connection

To connect the Arduino and the Ethernet shield to the internet, we used the [EtherCard](#) library. Because the ENC28j60 uses a different default CS pin (10 instead of 8), we had to add that in the code when making the connection. This is done by changing

```
if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
```

(with no pin specified, so the default pin is used) to

```
if (ether.begin(sizeof Ethernet::buffer, mymac, 10) ==  
    0)
```

Note the third argument 10 added after mymac.