ALGOFET Manual

Author: Winfried Ritsch, Atelier Algorythmics,

AlgoFET Version: 2.0, last updated 31.3.2007
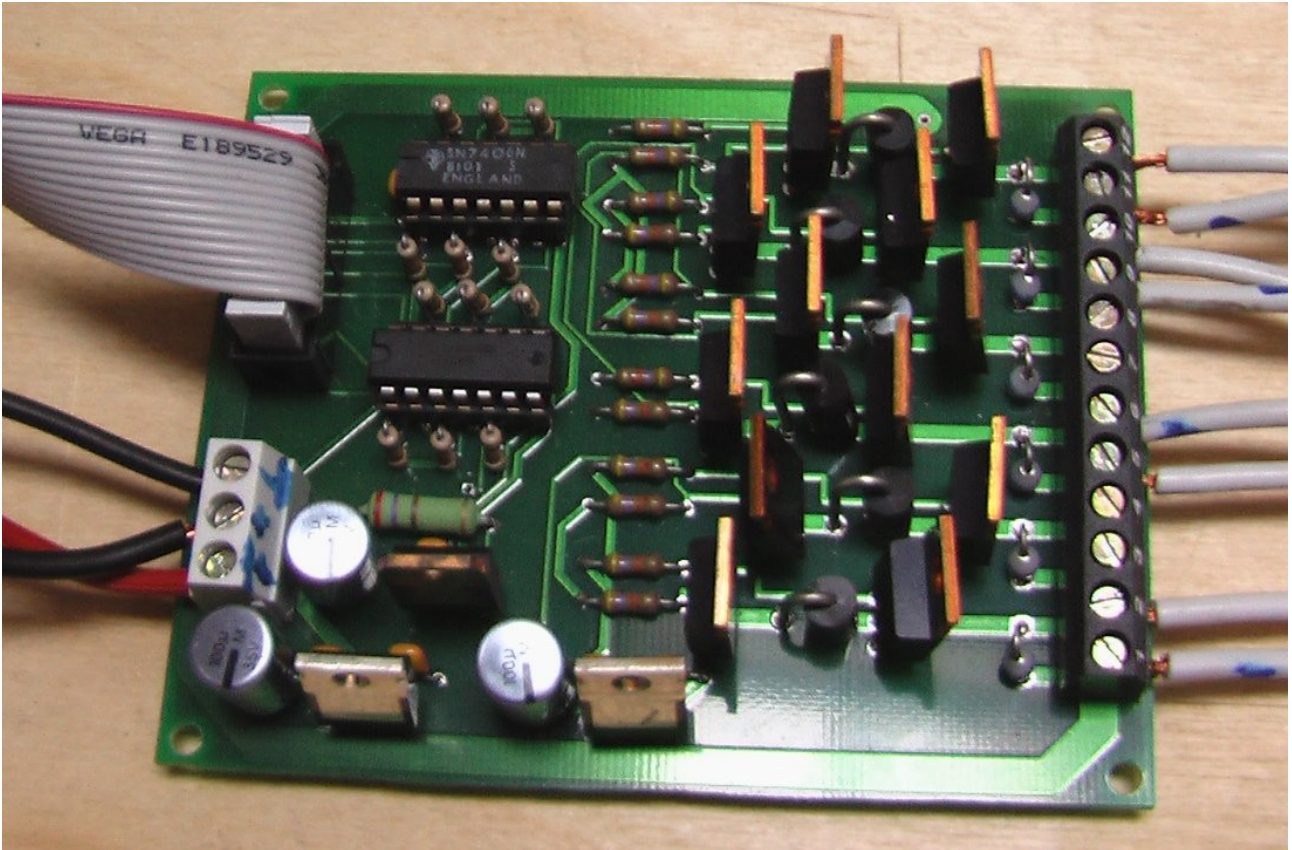

figure 1: ALGOFET cabled as an example with 4 of 6 Outs

This manual is a short description of the ALGOFET board v2.0 and is published under http://algo.mur.at/. As an example a typical use case is described, which can be adapted for special purposes. Therefore I will give some hints in this document. The used parts, especially the FETS and voltage regulators, have to be choosen for your special application. As a starting point the parts named in the circuit description can be used.

Introduction

ALGOFET is a board for controlling solenoids with a two voltages, using two FETs and two digital (TTL) input signals per channel. The board was designed with the free version of the layout-application "eagle".[1] The idea is to use a higher voltage V++ for a power-full attack, that must be only attached for a short time (eg. max 25% if time) to do not overheat the solenoid. Afterwards applying a hold-voltage V+ to hold the status of the magnetic field over a long period up to 100%. (Hint: Each solenoid has a specified on-time in percent for a nominal Voltage so that the solenoid does not overheat. So choosing the right Voltage

---

1 eagle is a layout editor from cadsoft, see http://www.cadsoftusa.com/info.htm

and on-time is very important.)

Like shown in figure 2 a typically timing is starting on time *t0* and giving an attack with a *V++* Voltage for a short time and then holding the a lower Voltage *V+* for an longer amount of time.
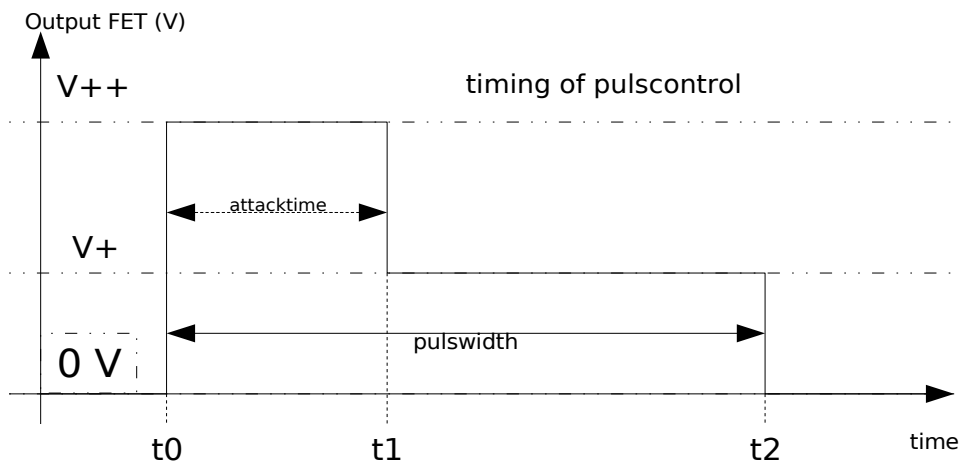

figure 2: Example timing of playing an solenoid

For this purpose a board was designed, which can handle these voltages and can be attached to the board with two wires. I uses two TTL-inputs from the ALGOPIC board for each output-channel.

Circuit

The whole detailed circuit can be viewed in the eagle files.

FET circuit

The circuit is simple two FETs which pull down one solenoid input either to V+ or to ground. Diode D1 is for suppressing the back-current from the solenoid and should be fast and diode D2 to block current flowing back to FET Q1 if Q2 is activated. To control the FETs the Gate voltage has to be pulled up to +12V over their source potential (or whatever you use in the power circuit) to get the lowest resistance the FET can do and the output of the TTL-Inverters are open collectors. The pull-up resistors R13, R14 should secure the circuit so if the board is unconnected the FETs should close(high resistance).

To choose a proper FET you should look for maximum current it needs and highest voltage it needs. Also respect the Gate voltage needed to fully open the FET. Here we can drive a BUZ11A without heat sink at 5A and maximum of 50V break through voltage.
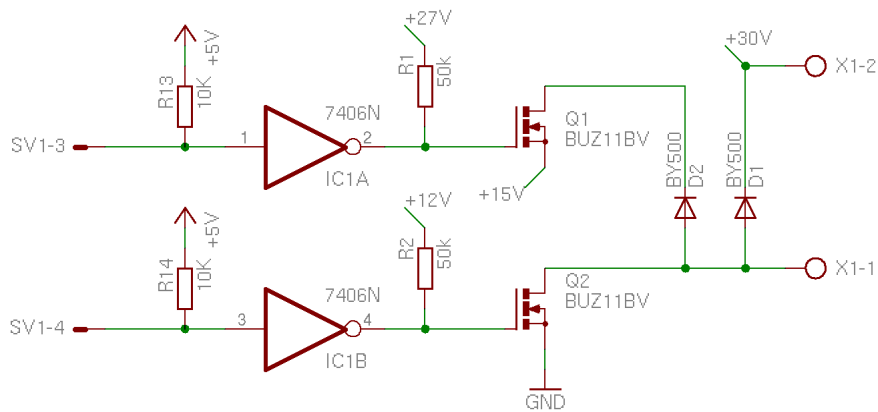
figure 3: FET circuit details

Power Supply

The voltages used depends on the application. (Hint: I used to have a variable power supply to adjust the holding and attack voltage for the solenoids). For example if the nominal Voltage is 12V (1A) I use a holding voltage between 7-15V and an attack Voltage from 15-48V, to control the maximum of attack.

The voltage regulators ICs 7812 are used to limit the maximum gate voltage, but can also be omitted an a wire can be used instead. (Hint: Some FETs needs a different gate voltage to switch in low resistance mode). The voltage regulator IC 7805 is used to supply the TTL inverting buffer driver 7406. It is important that this voltage is taken from the highest voltage V++, because if there is a voltage breakdown of the +5V, the solenoid can be burned. On the V+ lower voltage the solenoid should take no harm (since it should operate within the 100% Duty limit).

ALGOPIC connector

The inputs can be connected with a ribbon cable, so two ALGOFETs can be driven with one ALGOPIC.

Attention: On the ribbon cable cut the +5V supply, since this is done by the ALGOFET itself and has been connected to ground accidentally !!!

figure 4: cut +5V of ribbon cable

Two inputs always drive one FET-channel, so 12 signals are needed for 6 channels.

For the whole pin assignment see circuit figures in board directory. Pin 1 is ground, and Pin 2 unconnected (could be +5V). Pin 3,4 is for first FET, Pin5,6 second and so on. Pin 1 is always right to the key of the connector.

The power-supply are connected via clamps. You can drive each FET channel up to 7A which is 42A for all. (Hint: Use thick high quality cables if using high currents). The board has successfully used with 15A.
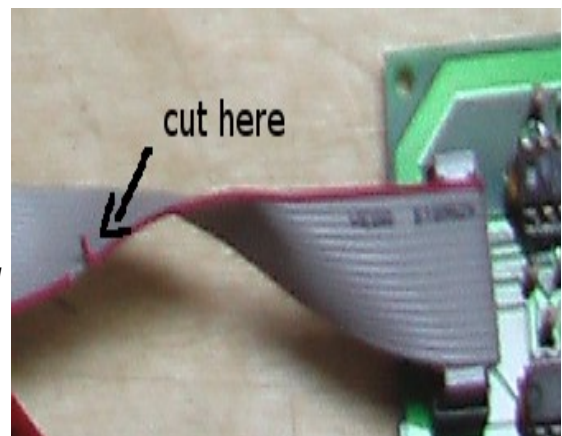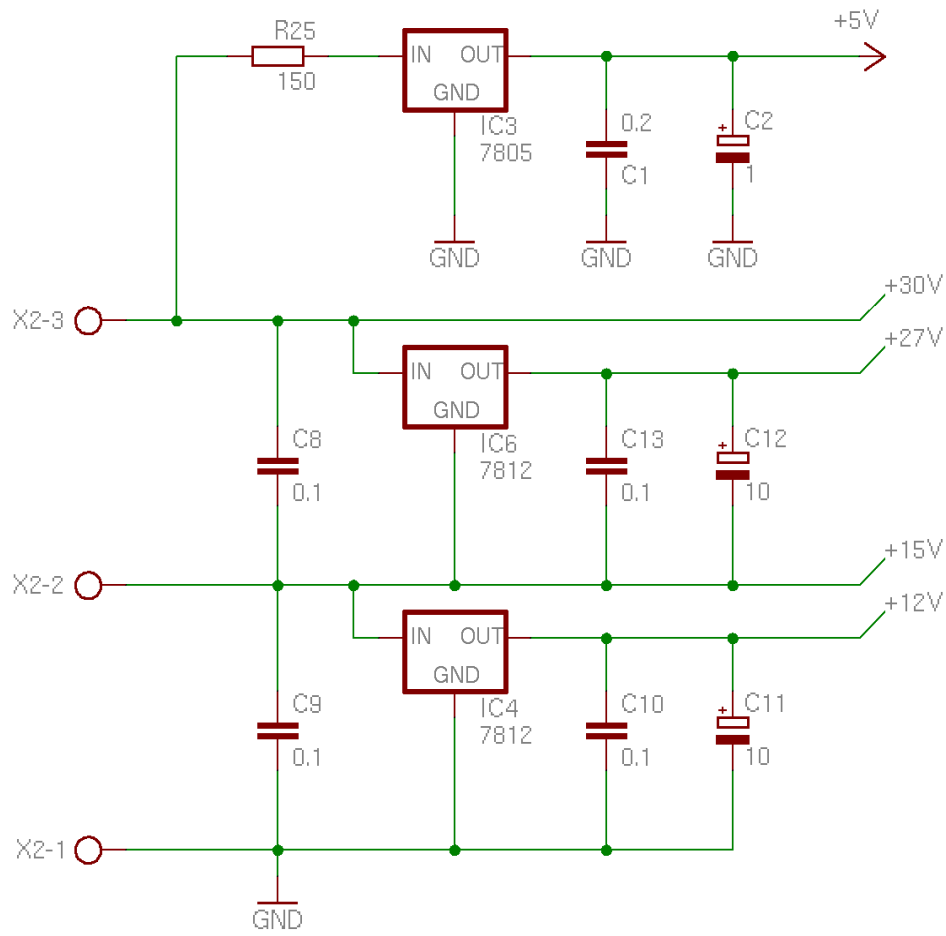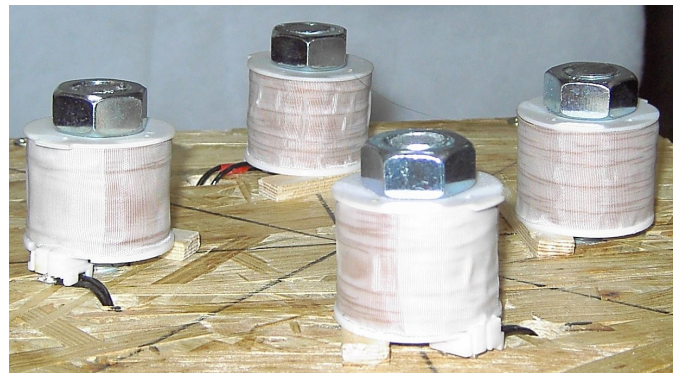
figure 6: power circuit of ALGOFET

figure 5: four coils to be driven by ALGFET

ALGOPIC application for controlling ALGOFET

The application to control the ALGOFET with ALGOPIC is implemented in assembler and can be used as as a starting point for special purposes. To control the ports the serial interface with following parameters are used:

- baudrate 115200
- handshake none
- format: 8Bit, 1 Startbit, 1 Stopbit, no parity

As syntax two bytes for each command is used where first byte is a Status Byte and the second data byte. For further details see syntax table 1.

All timing is done in interrupt which has a tick time of 0.4ms. To play one fet with different attacktime, *<velo>* is used. The *attacktime* can be calculated with following formula:

attacktime = 0.4ms * <scaling> * <velo>

Example:

<scaling> is 5 and <velo> 10 *attacktime* => 20ms

<scaling> is 100 and <velo> is 100 => *attacktime* is 4sec.

There is a example for PD[2] pd-application in the software folder.

```
<message> = <status> <data>

status byte has first bit set, data byte not:
      <status>    = B'1aaacccc' (128-255)
      <data>      = B'0ddddddd' (0-127)

Status:
      aaa = board adress 0-7 for identifying the board
      ccc = command Nummer (0-15)

Commands (ccc):
   0-11 ... play FET with <data> = velo
                        velo = (1-127) play fet
                        velo = 0 stop playing fet.
   12   ... set prescale <data> = prescale factor
   13   ... playmode: <data bit 0> = 0 ad hoc playing
                      <data bit 1> = 1 synchron playing
                      <data bit 1-7> = not used
   14 .. not used
   15 .. Synchron Puls
```
table 1: syntax for playing FETs

With command 12 the *scaling* can be set. There are two play modes, where playmode 'ad hoc' is default.

---

2 PD – Pure Data, real-time graphical programming environment see http://puredata.info/

Mode	Description	value
adhoc	play fets immidiatly after receiving a play-message	0
synchron	play fets synchron after receiving one or more play-messages	1

Disclaimer

I cannot be held responsible for the information contained within this docs and hardware documented here, even I tried hard not to make any failures. Any uses and/or misuse of the information that could result in bodily and/or mental harm to you or any other person that came in contact with the information cannot hold me or any company or anyone else responsible.